Smart Greenhouse using Machine Learning

Saumya Singh[1], Charen Vishwa[2]

[12]*Student, Electronics and Computer Engineering, Vellore Institue of Technology, Chennai, Tamil Nadu, India*

---***---

Abstract - *A greenhouse is a structure with transparent walls and roofs, typically made of glass, used for cultivating plants under controlled environmental conditions. The challenge lies in maintaining these conditions manually. Our project employs the Internet of Things (IoT) and machine learning to automate and optimize greenhouse operations.*

We utilize sensors to collect data on parameters like soil moisture, temperature, and humidity. This data is stored for future reference. IoT technology facilitates remote monitoring and analysis, connecting physical objects to the Internet.

Machine learning algorithms are then used to automatically adjust the greenhouse's climate to create an ideal growing environment. Additionally, the system allows for manual control by users.

Key Words: IoT, Machine Learning, Greenhouse, Sensors, Automation.

1. INTRODUCTION

A greenhouse is a place where vegetables and small plants grow under a transparent roof made of materials like glass or translucent plastic. Greenhouses are particularly important in colder regions where it's challenging to nurture vegetation. They create a greenhouse effect by trapping sunlight, increasing carbon dioxide levels, and ultimately promoting plant growth and yield.

The Internet of Things (IoT) is a system that combines sensors, controllers, actuators, physical objects, and the internet. It consolidates data from these devices and connects them to the internet, enhancing accuracy, efficiency, and economic benefits.

Machine learning, a subset of Artificial Intelligence (AI), empowers machines to learn from historical data and make decisions without explicit programming. In our project, data from sensors informs machine learning algorithms, which, in turn, dictate the actions to be taken.

Farmers often lack precise information, relying on past experiences and intuition, leading to unpredictable results. Our system aims to provide more accurate guidance for farmers in different environmental conditions.

The smart greenhouse comprises Arduino UNO microcontrollers, various sensors (e.g., temperature, soil moisture, humidity, and LDR sensors), and a Windows application for greenhouse control. When sensors detect specific conditions (thresholds or critical values), the system takes automated actions. For instance, a fan will start working when the greenhouse temperature crosses a critical value, say 30 degrees Celsius.

2. LITERATURE REVIEW

2.1 IOT Based Smart Greenhouse Automation Using Arduino

The proposed system consists of a microcontroller, specifically Arduino, along with various sensors like temperature, humidity, and light sensors. Additionally, there is a Windows application for managing the greenhouse and its internal conditions. The microcontroller serves as the system's core, monitoring sensor data once it's digitized and comparing it with predetermined threshold values. When an unfavorable situation is detected, it initiates the necessary corrective actions. The greenhouse also incorporates numerous sensors and actuators to regulate internal conditions. These sensors communicate with the microcontroller, which triggers appropriate responses when sensor data reaches predefined thresholds.

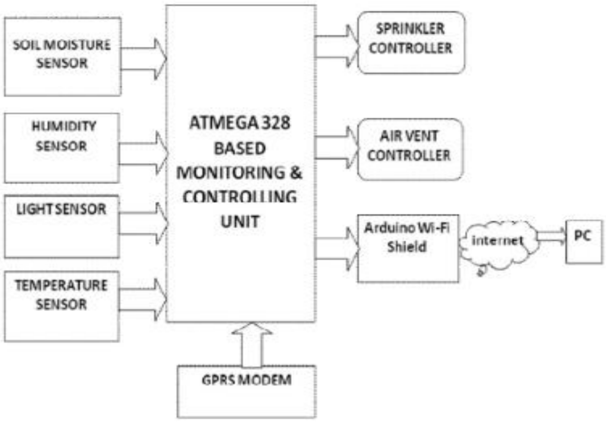

Fig -1: Architecture used in 2.1

2.2 Internet of Things Based Smart Greenhouse: Remote Monitoring and Automatic Control

In the beginning, the microcontroller initializes the GSM/GPRS module and HC05 Bluetooth module to connect the system to a network. The microcontroller then measures the condition of soil moisture and waters the plants if the level is lower than usual. Next, it takes the value of both temp and humidity, and takes air in or out and starts/stops the sprayers accordingly depending on the condition. Then it determines Day or Night condition using the LDR sensor. And at the end, all the data is sent to the user by sending an SMS using GSM, to the mobile application by HC05, and to the server using GPRS. There's also an LCD display along with the android app to show real-time data.

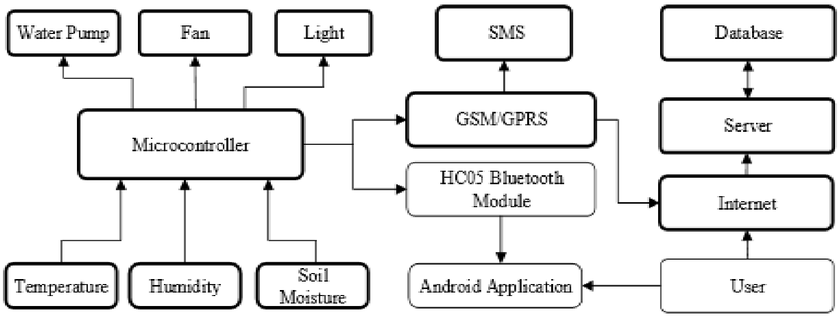

Fig - 2: Architecture used in 2.2

Comparing to other papers with similar perspective, this system is more precise and stable. As seen, they have used GSM-GPRS module to send data over internet, it is more reliable, fast and efficient comparing to ZigBee technology and RF modules.

2.3 IoT based smart greenhouse

This work primarily focuses on enhancing traditional agricultural methods through the adoption of modern technology to achieve higher crop yields. It introduces a smart temperature model that allows farmers to automate farming tasks, reducing the need for manual monitoring. Greenhouses, enclosed structures, shield plants from extreme weather conditions, including wind, hail, ultraviolet radiation, and pests.

Automated irrigation, based on precise soil moisture levels, ensures the optimal amount of water is used for crop growth. Soil health cards guide the application of nitrogen, phosphorus, potassium, and other essential minerals using composting techniques.

Despite India's ample rainfall and extensive river systems, only one-third of agricultural land is connected to canal systems. The remaining areas rely on rain or groundwater sources, resulting in water mismanagement. To address this issue, drip irrigation is employed to ensure efficient water usage.

2.4 IoT Based Intelligent Greenhouse Monitoring and Control System

Recently, Internet of Things (IoT) technology has revolutionized various systems, and this paper focuses on the design and implementation of an IoT-integrated greenhouse system. This system allows for the adjustment of environmental parameters and remote monitoring from anywhere in the world.

The system incorporates three intelligent controls aimed at regulating temperature, soil moisture levels, and light intensity within a thermal structure, such as a greenhouse. These controls consist of two essential components: hardware and software.

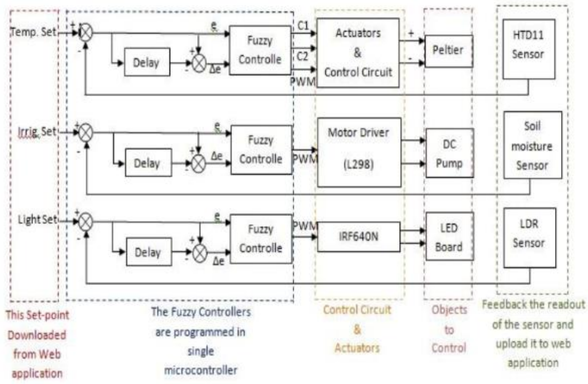

Fig - 3: Architecture used in 2.4

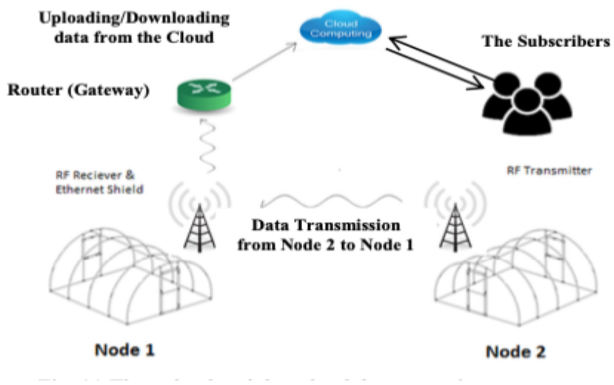

Fig - 4: Overview in 2.4

2.5 Iot Based Smart Greenhouse

The sensors linked to the Raspberry Pi are responsible for gathering environmental data, including temperature, humidity, soil moisture, and light intensity. This data is systematically collected and transmitted to the Ubidots cloud service through client API technology. An API, or Application Programming Interface, is a set of functions and processes that enable the creation of applications accessing features, data, or other services. The communication protocol utilized throughout the program is HTTP, and the cloud relies on a RESTful API service.

The REST API serves as the cornerstone of web design, allowing for complex interactions between clients (browsers) and servers without the client needing prior knowledge of the server's resources. The key challenge lies in ensuring both the server and client agree on the media format used, which, in the case of the web, typically involves HTML.

3. MATERIALS AND METHODS

- Temperature Sensor and Humidity Sensor: DHT11

  This sensor is used to measure temperature and humidity in the environment

- Light sensor: LDR (Light Dependent Resistor)

  This sensor is used to measure light intensity in the area. For example, if the light is too bright, we can put a shed on that area.

- Soil Moisture Sensor: VH400

  This sensor measures the moisture present in the soil. So, if the moisture is less, then we can start the motor which is connected to water supply

- Arduino UNO

  This is the core of our project. It collects data from all the sensors and sends it for analysis. When we use this system for different types of crops, machine learning becomes invaluable because it efficiently guides us on which parameters to follow for optimal results.

We built this circuit using Tinkercad, a user-friendly 3D CAD design tool. In this setup, we've connected light-detecting sensors, temperature sensors, and a potentiometer sensor (in place of a humidity sensor) to the Arduino. The Arduino is further linked to a Wi-Fi module, which enables us to transmit these data values to the Thingspeak server via the internet. Consequently, users with access to this server can take the necessary actions based on the information received.

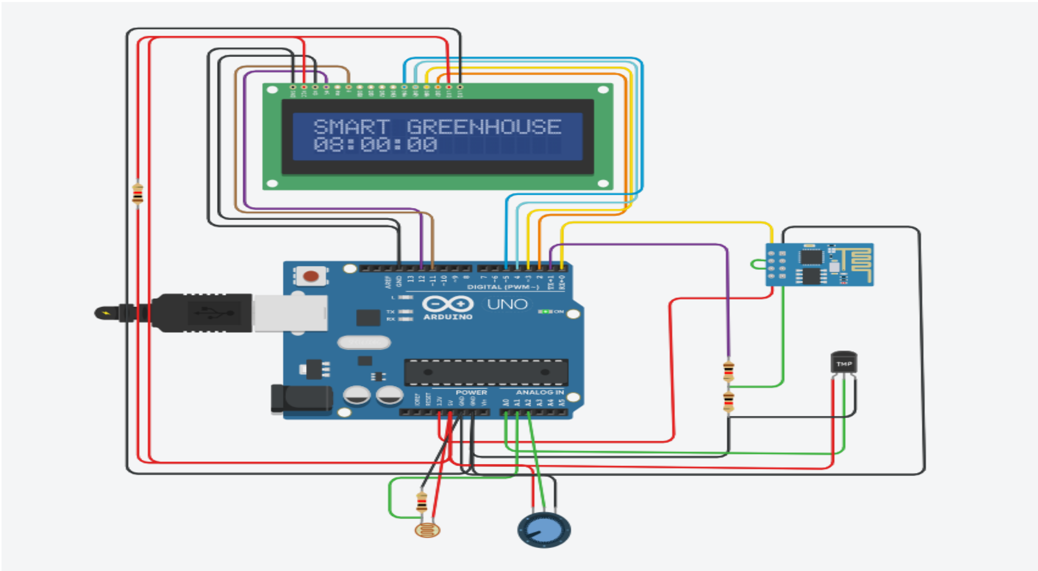

Fig – 5: Circuit simulation in Tinkercad

OUTPUT SCREENSHOTS OF THINGSPEAK SERVER:

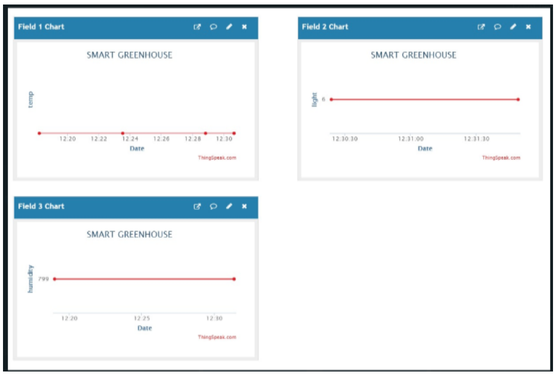

Fig – 6: Output 1

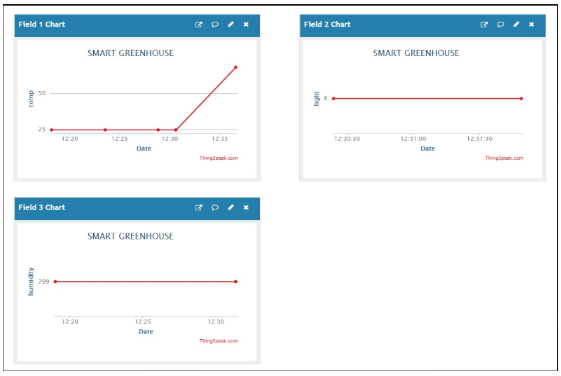

Fig – 7: Output 2

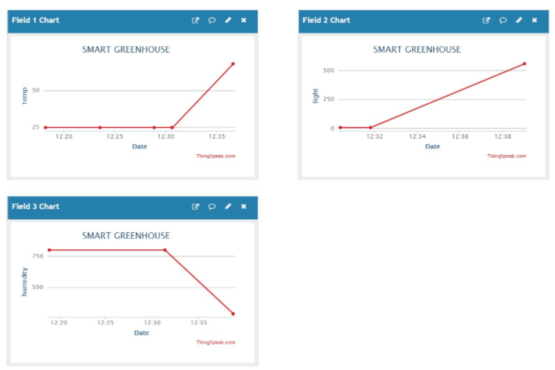

Fig – 8: Output 3

In our project, we've incorporated two Machine Learning (ML) algorithms to forecast various parameters. While we can predict each parameter individually, our current focus is on predicting Humidity. Specifically, we've implemented both a straightforward Linear Regression and a Random Forest Regressor. The Linear Regression model provides us with an accuracy rate of 97.99%, while the Random Forest Regressor boasts an accuracy rate of 99.02%. The Random Forest model excels in identifying more intricate relationships among variables, albeit at the expense of slightly longer computation time. However, if your variable of interest exhibits a linear dependency on the predictors, you'd likely achieve similar results with both algorithms. Given the significant computational complexity of the Random Forest model, especially with incorrect regularization parameters, it may not be worth the longer wait time for a marginal increase in performance. Nonetheless, when the underlying dependency is non-linear, the Random Forest Regressor tends to deliver slightly better accuracy.

Here's a step-wise breakdown of our algorithm:

Algorithm:

1. **Data Retrieval**: We start by extracting the predefined dataset from a '.csv' file.
2. **Data Split**: Using the train-test split functionality from scikit-learn, we divide the dataset into separate test and train sets.
3. **Model Selection**: Next, we apply the chosen ML models for classification, namely Logistic Regression and Random Forest Classification. The model training process occurs using the train set.
4. **Prediction**: After training, we predict the output of the test set, specifically the humidity value.
5. **Accuracy Assessment**: Finally, we calculate the accuracy of the predicted output from the test set

Dataset link:

https://www.kaggle.com/muthuj7/weather-dataset

	Formatted Date	Summary	Precip Type	Temperature (C)	Apparent Temperature (C)	Humidity	Wind Speed (km/h)	Wind Bearing (degrees)	Visibility (km)	Loud Cover	Pressure (millibars)	Daily Summary
0	2006-04-01 00:00:00.000 +0200	Partly Cloudy	rain	9.472222	7.388889	0.89	14.1197	251.0	15.8263	0.0	1015.13	Partly cloudy throughout the day.
1	2006-04-01 01:00:00.000 +0200	Partly Cloudy	rain	9.355556	7.227778	0.86	14.2646	259.0	15.8263	0.0	1015.63	Partly cloudy throughout the day.
2	2006-04-01 02:00:00.000 +0200	Mostly Cloudy	rain	9.377778	9.377778	0.89	3.9284	204.0	14.9569	0.0	1015.94	Partly cloudy throughout the day.
3	2006-04-01 03:00:00.000 +0200	Partly Cloudy	rain	8.288889	5.944444	0.83	14.1036	269.0	15.8263	0.0	1016.41	Partly cloudy throughout the day.
4	2006-04-01 04:00:00.000 +0200	Mostly Cloudy	rain	8.755556	6.977778	0.83	11.0446	259.0	15.8263	0.0	1016.51	Partly cloudy throughout the day.

Fig – 9: Dataset view

The Meteorological Dataset Consists of:

- Total Number of Data Point: 96453
- Total Number of Attribute/Columns in data set is 11.

Attribute Information:

1. Formatted Date
2. Summary
3. Precipitation Type
4. Temperature (C)

5. Apparent Temperature (C)
6. Humidity
7. Wind Speed (km/h)
8. Wind Bearing (degrees)
9. Visibility (km)
10. Pressure (millibars)
11. Daily Summary

The code can be found in the link below:

https://github.com/rookncook/Prediciting-Humdity-

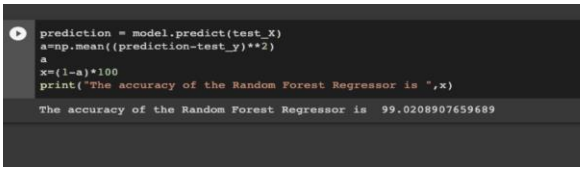

Fig – 10: Code Snippet 1

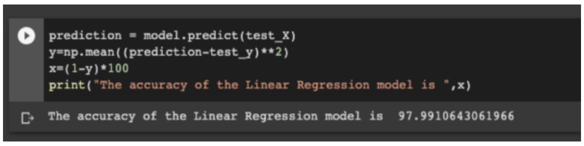

Fig – 11: Code Snippet 2

Linear Regression:

Simple linear regression stands as one of the most commonly employed statistical techniques. It offers a means to model the relationship between two sets of variables, yielding a linear regression equation. This equation serves as a valuable tool for making predictions based on the data.

Random Forest Regressor:

Random Forests emerge as a highly effective ensemble learning model employed for both classification and regression tasks. Ensemble learning models combine multiple machine learning models to achieve improved overall performance. The underlying principle is that each individual model used is relatively weak when applied independently but becomes potent when assembled into an ensemble. In the context of Random Forests, a substantial number of Decision Trees, considered the 'weaker' components, are employed, and their outputs are aggregated to form a 'strong' ensemble model.

4. RESULTS AND DISCUSSION

Features and Benefits of the System:

1. Temperature Control: Detects and maintains the optimal temperature for plant growth.
2. Humidity Management: Detects and regulates humidity levels to ensure ideal growing conditions.
3. Soil Parameter Monitoring: Monitors and manages essential soil parameters for sustainable plant growth.
4. Cloud Updates: Provides real-time updates to the cloud for remote monitoring and data access.
5. Automated Control: Intuitively and automatically controls various parameters, enabling year-round cultivation of different plant varieties.

6. Error Reduction: Eliminates human errors in the cultivation process.
7. Reduced Labor Costs: Drastically reduces labor costs associated with manual plant care.
8. User-Friendly: Offers a "plug-and-play" product, ensuring ease of use and setup.

5. CONCLUSION

This paper presents the creation and deployment of a regulated system accessible to the public. As data continuously streams to the cloud, it enables precise parameter adjustments. Storing data in the cloud ensures reliable future analysis, providing valuable assistance to farmers. Leveraging IoT technology, this system can be controlled from anywhere in the world, making it adaptable for various plant requirements. By making minor parameter adjustments, this concept can be applied in different scenarios as needed.

REFERENCES

[1] Shirsath, D.O. & Kamble, Punam & Mane, Rohini & Kolap, Ashwini & More, Rajendra. (2017). IOT Based Smart Greenhouse Automation Using Arduino. International Journal of Innovative Research in Computer Science & Technology. 5. 234-238. 10.21276/ijircst.2017.5.2.4.

[2] Woli Ullah, Mohammad & Mortuza, Mohammad & Kabir, Md & Ahmed, Zia & Supta, Sovan & Das, Partho & Hossain, Syed. (2019). Internet of Things Based Smart Greenhouse: Remote Monitoring and Automatic Control. DEStech Transactions on Environment, Energy and Earth Sciences. 10.12783/dteees/iceee2018/27803.

[3] R. K. Kodali, V. Jain and S. Karagwal, "IoT based smart greenhouse," 2016 IEEE Region 10 Humanitarian Technology Conference (R10-HTC), Agra, India, 2016, pp. 1-6, doi: 10.1109/R10-HTC.2016.7906846.

[4] A. Jasim, Abbas & Marhoon, Ali & Faisal, Zaidon. (2018). IoT Based Intelligent Greenhouse Monitoring and Control System.

[5] Shreyas Bhujbal, Yash Deshpande, Arpit Gupta and Ojas Bhelsekar "Iot Based Smart Greenhouse " in International Journal of Innovative Research in Science, Engineering and Technology, Vol. 7, Issue 1, January 2018

[6] D. Kornack and P. Rakic, “Cell Proliferation without Neurogenesis in Adult Primate Neocortex,” Science, vol. 294, Dec. 2001, pp. 2127-2130, doi:10.1126/science.1065467.

Pallavi Sethi, Smruti R. Sarangi, "Internet of Things: Architectures, Protocols, and Applications", *Journal of Electrical and Computer Engineering*, vol. 2017, Article ID 9324035, 25 pages, 2017. https://doi.org/10.1155/2017/9324035

[7] D. S. Kim, T. H. Shin, and J. S. Park, “A security framework in rfid multi-domain system,” in Availability, Reliability and Security, 2007. ARES 2007. The Second International Conference on, April 2007, pp. 1227–1234.

[8] Dataset link: https://www.kaggle.com/muthuj7/weather-dataset

[9] A. Bseiso, B. Abele, S. Ferguson, P. Lusch, and K. Mehta, “A decision support tool for greenhouse farmers in low-resource settings,” in Global Humanitarian Technology

[10] Z. Xiaoyan, Z. Xiangyang, D. Chen, C. Zhaohui, S. Shangming, and Z. Zhaohui, “The design and implementation of the greenhouse monitoring system based on gsm and rf technologies,” in Computational Problem-solving (ICCP), 2013 International Conference on, Oct 2013, pp. 32–35.

[11] M. Mahdavian, M. B. Poudeh, and N. Wattanapongsakorn, “Greenhouse lighting optimization for tomato cultivation considering real time pricing (rtp) of electricity in the smart grid,” in Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2013 10th International Conference on, May 2013, pp. 1–6

[12] D. S. Kim, T. H. Shin, and J. S. Park, “A security framework in rfid multi-domain system,” in Availability, Reliability and Security, 2007. ARES 2007. The Second International Conference on, April 2007, pp. 1227–1234.

[13] Conference (GHTC), 2015 IEEE, Oct 2015,)K. A. Czyzyk, S. T. Bement, W. F. Dawson, and K. Mehta, “Quantifying water savings with greenhouse farming,” in Global Humanitarian Technology Conference (GHTC), 2014 IEEE, Oct 2014, pp. 325–332