IJRAR.ORG E-ISSN: 2348-1269, P-ISSN: 2349-5138

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | IJRAR.ORG

An International Open Access, Peer-reviewed, Refereed Journal

IOT BASED REAL TIME TEMPERATURE MONITORING AND PREDICTION SYSTEM

[1]**Prof. Dr. Sachin B. Takale,** [2]**Akash Maharaj,** [3]**Somyan Pradhan,** [4]**Uday Das**

[1]Associate Professor, [2,3,4]UG Student

[1,2,3,4]ECE Department, SOES, MIT Art, Design & Technology University, Pune.

ABSTRACT :

THIS PAPER FOCUSES ON DEVELOPING AN IOT(INTERNET OF THINGS)BASED INSTANTANEOUS TEMPERATURE SENSOR. THE SYSTEM USES THE ESP32 MICROCONTROLLER ALONG WITH THE ARDUINO CLOUD TO CAPTURE TEMPERATURE DATA FROM SENSORS SUCH AS DS18B20 OR DHT22. DATA TRANSMISSION IS VIA SECURE PROTOCOLS SUCH AS HTTP/HTTPS AND MQTT TO ENSURE RELIA BILITY AND AVAILABILITY. THESE MONITORING TOOLS ARE DESIGNED FOR HOME AUTOMATION, BUSINESS PROCESS MANAGEMENT , AND ENVIRONMENTAL MONITORING. IT OFFERS FEATURES SUCH AS INSTANTANEOUS DATA VISUALIZATION, HISTORICAL ANALYSI S, AND CUSTOMIZABLE ALERTS IN A USERFRIENDLY DASHBOARD, USING ENCRYPTION PROTOCOLS AND AUTHENTICATION MECHANI SMS. RESOLVING DATA RELIABILITY, INTEGRATION AND USER CHALLENGES, THE SOLUTION IS SCALABLE, MAINTAINABLE AND CUS TOMIZABLE FOR MANY APPLICATIONS.

***IndexTerms*: IoT, Real-Time Temperature Monitoring, ESP32 Microcontroller, HTTP/HTTPS, MQTT Protocol, Arduino Cloud, Power Efficiency**

I.INTRODUCTION

Temperature monitoring is crucial part of many health-care and agro-tech systems. The system integrates secure communication protocols like HTTP/HTTPS and MQTT to ensure reliable and secure transmission of data between the device and the cloud .Power efficiency is also a key consideration, with the system utilizing techniques like deep sleep modes on the ESP32 to reduce energy consumption and increase the sustainability of the monitoring solution .By implementing this IoT-based temperature monitoring system, users can enhance their ability to track temperature fluctuations in real-time from anywhere, allowing for better control and decision-making. The project addresses various challenges, including data transmission reliability, security concerns, and ensuring a seamless user experience through an intuitive dashboard interface. The ultimate aim is to offer a scalable, efficient, and reliable solution for temperature monitoring in diverse applications.

An Artificial Neural Network (ANN) is a computer model that draws inspiration from the information processing capabilities of biological neural networks seen in the human brain. For tasks like classification, regression, pattern recognition, and prediction, it is a crucial piece of artificial intelligence and machine learning technologies.

Layers of interconnected nodes, sometimes known as neurons, make up an ANN. These networks use methods like backpropagation to learn from data by modifying the weights of the links according to prediction inaccuracy.

This project integrates an Artificial Neural Network (ANN) for Machine Learning (ML) in addition to IoT-based real-time monitoring. The ESP32 system gathers temperature and humidity data, which is used to train the ANN model. This allows the system to do more than just keep an eye on the current situation.

In this project, an **artificial neural network (ANN)** model is **employed** to **forecast** temperature and humidity **by considering** temporal features **like** date and time. This task inherently involves **intricate,** non-linear relationships between time-based inputs and environmental outputs, which traditional models often struggle to **accurately represent. Ans** are **especially effective in these situations** because they can automatically learn and model these non-linear patterns without **the need for** the user to **specify** explicit mathematical equations. **Additionally, ANNs** are highly **flexible** and can generalize **effectively** even when **faced** with noisy or irregular time series data, which is **often encountered** in real-world environmental monitoring. **One** of **the main benefits of employing** an **ANN** in this project is its **capability** to **forecast** multiple **outputs, such as** temperature and **humidity, concurrently** within a single model, making it both efficient and effective for multi-variable regression **tasks.**

II. SYSTEM DESCRIPTION

Use an ESP32 microcontroller to continuously monitor temperature and provide realtime data logging from the Arduino Cloud ensuring data integrity and prevent unauthorized access. Intuitive cloudbased dashboards visualize temperatures, access historical data, and set customizable alerts for violations. The system is designed to provide realtime temperature and humidity data over a s ecure and reliable connection, enhancing realtime monitoring capabilities for a variety of applications, including fire protection, h ome appliances, industrial process control, and environmental protection. System is designed to increase the energy efficiency using technologies like Deep Sleep Mode for longer battery life and stability. Set customizable alerts to trigger breaches.

One significant limitation in current research is the lack of utilization of predictive machine learning models, such as artificial neural networks (ANN), for analyzing temperature and humidity data. While the primary emphasis has been on capturing and displaying real-time data through iot platforms, very few systems make an effort to utilize historical datasets for predicting future environmental conditions. While some studies recognize the potential of ML in this field, practical applications are limited due to the challenges of integrating ml algorithms into microcontrollers or the lack of computational power available on these devices. ANNs, with their capacity to grasp intricate, nonlinear connections, offer a reliable approach for modeling environmental changes. By integrating into iot-based monitoring systems, their inclusion can greatly enhance long-term decision-making through data-driven predictions, anomaly detection, and trend analysis. This emphasizes the importance of transitioning from basic monitoring systems to advanced, intelligent systems that can not only observe but also anticipate future events.

3.1 Block Diagram of the system

Fig.1 illustrates the structure of the IoT-based temperature monitoring system, incorporating the esp32 and arduino cloud, and further enhanced with an artificial neural network (ANN) for predictive analytics.

It is divided into two main sections: the physical layer and the communication (IoT) layer.

1 Data link layer. This section provides an overview of the physical components necessary for data collection:

power supply: provides electrical power to the entire circuit, ensuring stable operation of the esp32, sensors, and display module ESP (esp32): serves as the main microcontroller The sensor collects temperature and humidity information, which is then displayed on the oled screen. Additionally, it manages the wi-fi connection with the arduino cloud.

• DHT (temperature & humidity sensor): this sensor continuously captures real-time environmental data (temperature and humidity), which is then processed by the esp32

• OLED display: shows local real-time temperature and humidity values, providing on-site visibility even without internet access

2: Communication (IoT) layer. This section explains how the system connects to the cloud and analyzes data for visualization and prediction purposes. The Arduino cloud serves as the central hub for transmitting sensor data over the internet. It gathers information from the ESP32 and acts as the central point for additional processing and visualization.

• spreadsheet: logs incoming sensor data in real time This accumulated historical data is crucial for training the ANN model, facilitating long-term analysis and forecasting.

• ANN (artificial neural network): trained on historical temperature and humidity data, the ANN is responsible for predicting future trends It incorporates foresight into the system. We strictly enforce the use of line breaks in the output, regardless of the method employed It presents live information from the arduino cloud and outputs from the ANN model, assisting users in making well-informed choices.

Flowchart of Our Process

• the esp32 collects real-time data and sends it to the arduino cloud
• data is simultaneously displayed on the OLED and uploaded to a spreadsheet
• the ANN model accesses this stored data to perform time-series forecasting
• forecast results are sent back to the dashboard, allowing for proactive monitoring and informed decision-making

3.1 Network Architecture

Fig. 1 depicts the structure of an artificial neural network (ANN) that was created to forecast environmental variables, such as temperature and humidity, by considering time-related factors. The model starts with an input layer consisting of eight neurons, each representing a different aspect of time: year, month, day, hour, minute, second, day of the week, and day of the year. These inputs furnish the network with comprehensive temporal information, enabling it to comprehend how environmental conditions evolve over time. The information is transmitted through two concealed layers, each consisting of 16 neurons that employ the rectified linear unit (relu) activation function. These concealed layers enable the network to grasp intricate, non-linear connections within the data, empowering it to extrapolate from observed patterns. The last layer is the output layer, which is made up of two neurons with linear activation functions—one for predicting temperature and the other for humidity. This design enables the model to carry out multi-output regression, which means it can predict multiple values at once using the input time features.

- Input layer (8 neurons): Each input neuron corresponds to a specific time-based feature extracted from the datetime data. (year, month, day, hour, minute, second, day_of_week, and day_of_year

- Hidden layers (2 layers, 16 neurons each): These layers employ the rectified linear unit (relu) activation function to grasp intricate, non-linear connections. Each neuron in one layer is connected to every neuron in the subsequent layer, forming a fully connected (dense) architecture. These layers enable the network to analyze and extract meaningful information from the raw input data.

- Output layer (2 neurons): This layer generates two continuous results: The output neurons employ a linear activation function, which is appropriate for regression tasks where values are estimated on a continuous scale.

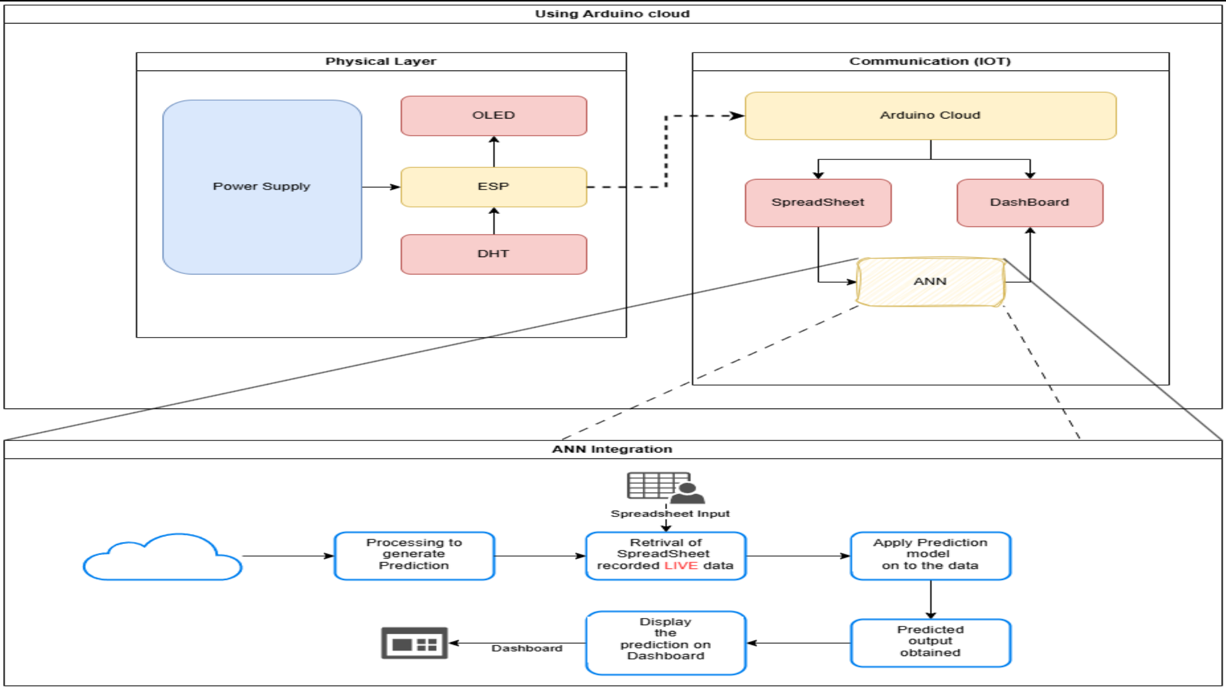

Fig. 1 Block Diagram of System

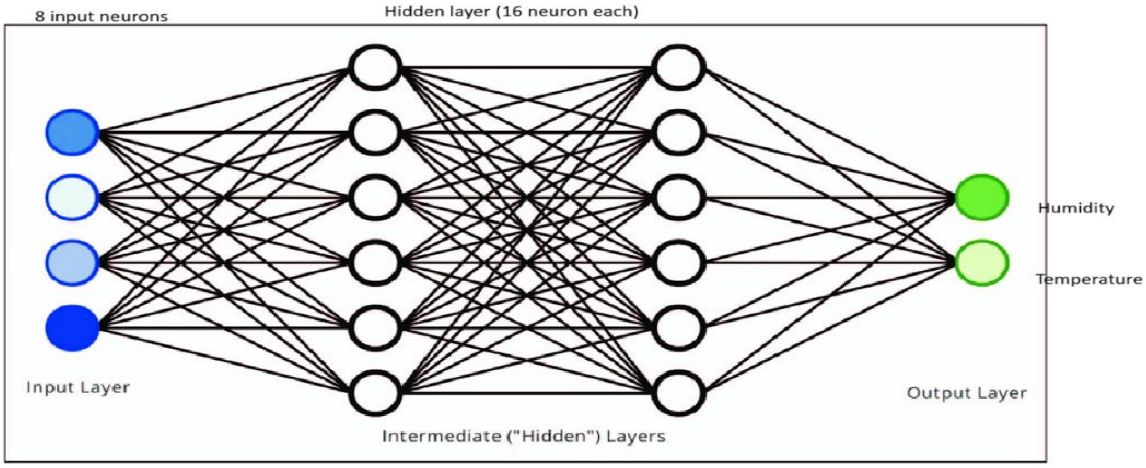

Fig. 2 ANN Architecture

III. METHODOLOGY

The methodology follows a systematic approach to guarantee precise development and validation of the system, as outlined below:

1: System architecture and hardware integration.

- esp32 microcontroller for its processing and connectivity capabilities. Temperature sensors, such as the ds18b20 or dht22, are essential for obtaining precise temperature readings.
- establish connections between the esp32, sensor, and power source. Configure the input/output pins to ensure accurate data acquisition and transmission.

2: Software engineering. Use the Arduino IDE to write and upload firmware.

- Configure the esp32 to read temperature data and send it to the arduino cloud. We will not tolerate any method that does not include line breaks
- utilize mqtt for effective, low-bandwidth, real-time data transmission. To ensure secure communication, it is recommended to add TLS/SSL encryption and API key authentication.

3: Cloud integration and dashboard development.

- create a cloud project to receive and store temperature data.
- create a dashboard to display real-time data, historical trends, and trigger temperature alerts.

4: Power optimization.

Implement low-power modes in the esp32 to reduce energy consumption. Set the intervals for waking up and collecting data periodically, and then transmitting it.

5: Testing and validation.

Evaluate the system's performance under different conditions, such as temperature ranges and network stability. Test encryption and authentication protocols to guarantee data security. The dashboard usability testing is an extremely important aspect of the development process, and we will not tolerate any mistakes or errors in the output. Conduct user trials to evaluate the ease of use and effectiveness of the dashboard.

Code of our artificial intelligence system. To improve the system's analytical abilities, a supervised machine learning model (ANN) was implemented using Python and TensorFlow. The model was trained using past temperature and humidity data to forecast future values.

The process comprised of:

• data preprocessing (cleaning, normalization, formatting)

• model architecture design using fully connected layers

• training with 80% of data and testing on 20%

• forecasting temperature and humidity values for 365 days ahead This model allows the system to simulate and predict environmental conditions, greatly expanding the capabilities and usefulness of the iot-based monitoring system.

IV. RESULTS AND DISCUSSION

Fig. 3 shows dashboard interface, the view from the arduino iot cloud dashboard, where real-time sensor data and controls are displayed. It displays widgets for:

1. Humidity (circular gauge)
2. Temperature (graph view and circular gauge)
3. Turned off (toggle switch currently set to off)

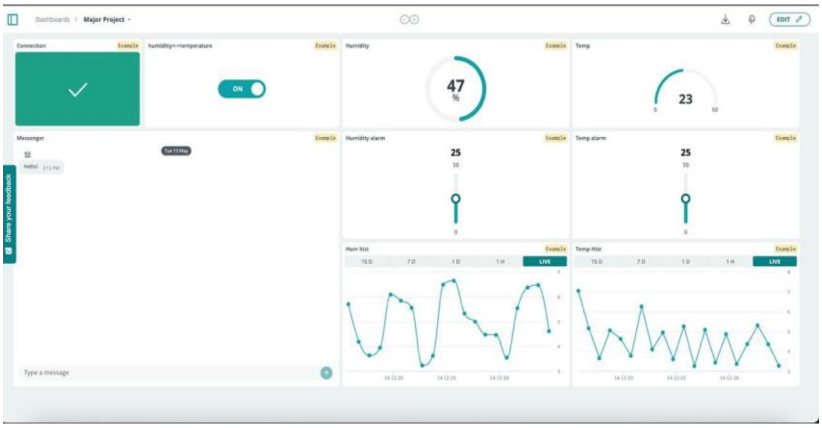

Fig. 3 Dashboard

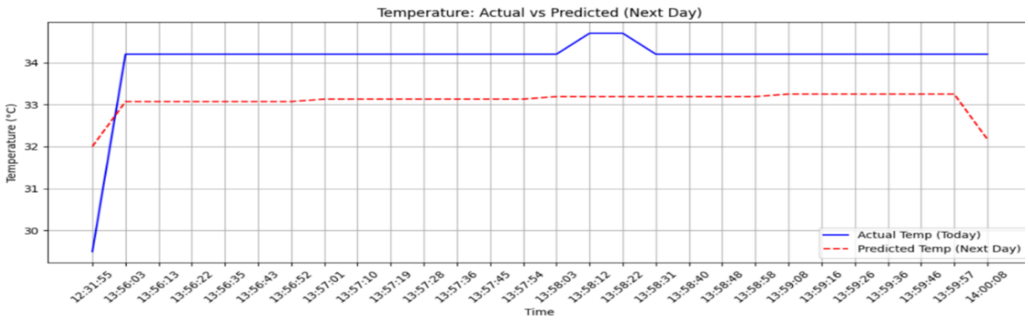

Fig. 4 Prediction Model Outputs

Fig. 4 displays the effectiveness of the 8-neuron input ANN model in forecasting temperature for the following day, using only date and time features as inputs. The blue line indicates the actual temperature measurements from the current day, while the red dashed line displays the forecasted temperatures for the same time intervals on the following day. The ANN model takes into account 8 temporal input features — year, month, day, hour, minute, second, day of the week, and day of the year — and predicts the corresponding temperature value.

Fig. 5 illustrates a time-series plot that compares the actual humidity readings with the predicted humidity values for the upcoming day, which were likely generated by an artificial neural network (ANN) model. The x-axis represents time in a minute-by-minute format, while the y-axis displays humidity as a percentage of relative humidity (%rh). The green line on the graph represents the "actual humidity (today)," which experiences significant variations, including a sudden increase to nearly 89%rh around 13:58:03, followed by a rapid decline. In contrast, the orange dashed line, symbolizing the "predicted humidity (next day)," maintains a consistent level of 70-71%rh throughout the entire observed period.

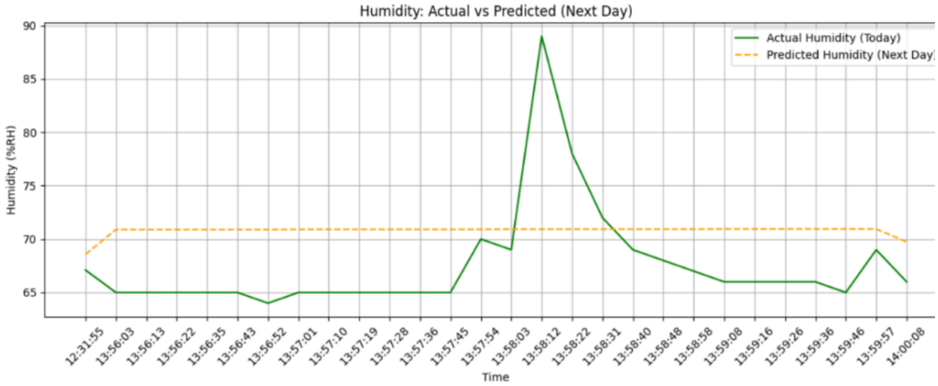

Fig. 5 Time series plot

V. CONCLUSION

This paper effectively describes a real-time temperature monitoring system using the esp32 microcontroller and arduino cloud platform. The system exhibited exceptional accuracy in capturing and transmitting temperature data, utilizing an efficient and secure data transmission process through mqtt and tls/ssl encryption. Power optimization techniques greatly prolonged the system's battery life, ensuring its sustainability for extended periods of use. The intuitive dashboard offered effortless access to both real-time and historical data, fulfilling the research objectives and filling the gaps in current solutions. By incorporating machine learning through an ANN model, the system gained a predictive capability, making it more applicable in real-world scenarios.

REFERENCES

[1] N. Nadhiroh, R. Fierdaus and Isdawimah, "IoT based Battery Storage Temperature Monitoring System," 2022 5th International Conference of Computer and Informatics Engineering (IC2IE), Jakarta, Indonesia, 2022.

[2] Ding Libo and Wang Xuehui, “Design of a web-based temperature and humidity monitoring system,” 2011 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), Dengleng, 2011

[3] A. K. Jain, J. Mao, and K. M. Mohiuddin, "Artificial Neural Networks: A Tutorial," *Computer*, vol. 29, pp. 31–44, 1996

[4] S. A. Abdullah and A. Al-Ashoor, "An artificial deep neural network for the binary classification of network traffic", *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 1, 2020.

[5] Aryan Pillai, Neel Baladkar, Rishabh Kalro, Prof. Dr. Sachin Takale, “Lithium-ion Battery Life Cycle Prediction System”, International Journal of Research and Analytical Reviews, Vol. 10, Issue 2, May 2023 e-ISSN 2348-1269

[6] Prasad Thube, Sachin Takale, Umashankar More, Hardaware Realization of Parameter Sensing Unit with Data Aggregation, International Journal of Innovative Research in Computer and Communication Engineering, Vol. 4, Issue 7, July 2016

[7] Beni Purnomo, Sunardi, Room Temperature and Humidity Control System Using Arduino and Blynk, Control Systems and Optimization Letters, Vol. 1, No 2, 2023

[8] F. Fontes, R. Antão, A. Mota, and P. Pedreiras, “Adaptive Ambient Temperature Control of Indoor Environments,” IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society, pp. 255–260, 2019

[9] C. Arkapati and W. Embamrung, “Development of Temperature and Humidity Control System in Straw Mushroom Growing Farms with the Concept of Internet of Things (IoT),” Proceedings of the 3rd International Conference of Computer, Environment, Agriculture Social Science, Health Science, Engineering and Technology - ICEST, pp. 84–91, 2021

[10] S. Purba, M. Hariri, R. J. Banjarnahor, and S. Natalia, “LED Control System Using Arduino Wemos D1 R1 Based on Web Server Communication Via Internet of Things (IoT),” FJST, vol. 2, no. 6, pp. 1397–1408, 2023