Temperature and Humidity Prediction

A major project report submitted in partial fulfilment of the requirement for the award of degree of

Bachelor of Technology

in

Computer Science & Engineering

Submitted by

Abhishek Anand (201366)

Aman Anand (201380)

Under the guidance & supervision of

Dr. Deepak Gupta

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology, Waknaghat, Solan - 173234 (India)

CERTIFICATE

This is to certify that the work presented in the "Temperature and Humidity Prediction" project report, which was submitted to the Department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology, Waknaghat in partial fulfilment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering, is an authentic record of work completed by Abhishek Anand (201366) and Aman Anand (201380) during the period from August 2023 to May 2024 under the supervision of Dr. Deepak Gupta, Department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology, Waknaghat.

Abhishek Anand
(201366)

Aman Anand
(201380)

Dr. Deepak Gupta
Assistant Professor (SG)
Computer Science & Engineering and Information Technology
Jaypee University of Information Technology, Waknaghat

CANDIDATE'S DECLARATION

I hereby declare that the work presented in this report entitled **'Temperature and Humidity Prediction'** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science & Engineering** submitted in the Department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from August 2023 to May 2024 under the supervision of **Dr. Deepak Gupta** (Assistant Professor (Senior Grade), Department of Computer Science & Engineering and Information Technology).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

(Student Signature with Date)
Student Name: Abhishek Anand
Roll No.: 201366

(Student Signature with Date)
Student Name: Aman Anand
Roll No.: 201380

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Supervisor Signature with Date)
Supervisor Name: Dr. Deepak Gupta
Designation: Assistant Professor (Senior Grade)
Department: Computer Science & Engineering and Information Technology
Dated:

ACKNOWLEDGEMENT

We would like to express our sincere gratitude and debt of gratitude to the supervisor, **Dr. Deepak Gupta, Assistant Professor (Senior Grade)**, Department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology in Waknaghat. Our supervisor is well-versed in the field of machine learning and has a keen interest in completing this assignment. This project was made possible by his never-ending patience, academic leadership, constant encouragement, frequent and energetic supervision, constructive criticism, insightful counsel, reviewing numerous subpar draughts and fixing them at all levels.

We would like to extend our sincere gratitude to **Dr. Deepak Gupta,** Department of Computer Science & Engineering and Information Technology, for his gracious assistance in seeing our project through to completion.

Additionally, we humbly thank our team who has contributed to the success of this project, whether directly or indirectly. In this case, we should express our gratitude to all of the staff members teaching and non-teaching who have provided us with practical assistance and made our task easier.

Finally, we must respectfully appreciate our parents unwavering assistance and endurance.

Abhishek Anand
(201366)

Aman Anand
(201380)

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

THM	Temperature and Humidity Monitoring
DHT11	Digital Humidity and Temperature Sensor 11
Arduino	Arduino Uno microcontroller board
MySQL	MySQL relational database management system
API	Application Programming Interface
HTTP	Hyper Text Transfer Protocol
JSON	JavaScript Object Notation
XML	Extensible Markup Language
JSX	JavaScript XML
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

In an era of increasing climate change, timely and accurate weather forecasts have become an important for management and decision making. This project “Temperature and Humidity Prediction” deals with this critical need through the development of powerful machine learning techniques with accurate weekly projection to temperature and humidity.

Now the DHT11 sensor and Arduino ESP8266 microcontroller become important. The temperature and humidity information are provided through these components and work well together which is accurate for the weather prediction. After that, for the training of machine learning model the data set is used from the sensor.

The Controlling of IOT device is carried out by closely monitoring the critical variables that produce important information about how these electronic devices operate. In addition to transmitting these crucial data from the transmitting device, this information will save it on the cloud for usage by applications and further operations. This analysis links the results of environmental observations, such as temperature and humidity readings taken with sensors. Profitably, the collected data might be utilised to create remotely dominant cooling systems, heating devices, or long-term statistics that will be helpful in controlling the same. Beyond rain forecasts, this technique offers a potential application.

Using this project people can improve efficiency in their operations by making important decision about material and site preparation. Energy manager is used to forecasts and to control energy in an accurate manner, maximising resource and cutting expenses use. By adjusting patient treatment and preventive measures according to anticipated humidity levels, medical personnel can reduce the risk of respiratory diseases and other weather-related health issues. The sensors are used to record the information with various devices and measure the ambient temperatures. Energy managers can employ the predictions to effectively manage energy consumption, reducing costs and optimizing resource utilization. Healthcare professionals can tailor patient care and preventive measures based on predicted humidity levels, mitigating the risk of respiratory illnesses and other weather-related health concerns

CHAPTER 1: INTRODUCTION

1.1 Introduction

Accurate temperature and humidity predictions are vital for a vast number of applications like weather forecasting, environmental monitoring, and agriculture management. To meet the above demand, we have developed an all-round system with its machine learning algorithms and Arduino, which predicts temperatures and humidity live.

The employed approach implements several sensors coupled with Arduino. It constantly monitors the temperature and humidity levels. The third phase involved transmitting the real time data to a central server for further analysis. These data are archived into a dedicated database that allows efficient accessing and interpreting it. Machine Learning Algorithms like linear regression analyze previous date and learn to predict highly accurate future temperatures and humidity values by using the past data. These anticipatory results are expected for temperature and humidity can be seen in a user-friendly website. The site allows users to analyze their data using interactive tools as well as to get various visualizations of these figures.

Through sensors and an Arduino, we continuously monitor and keep track of the temperatures and humidity levels. The Arduino is a microcontroller board that may be programmed to read sensors data and send data to other devices. We apply temperature and humidity sensors, whose electric impulses comply with environment's parameters are recorded. With the help of an Arduino, the data collected from the sensors in the form of electrical signals is converted to digital and send to a central server. This is what the central server is referred to the computer that contains the raw data and is used to process the information saved by the Arduino. This kind of information is stored in a database that organizes the data into a form easily accessible by any modern-day computers or programs.

At the central server, they also employ machine learning algorithms through which they evaluate previous temperatures and humidity conditions to predict future readings. Machine learning algorithms are just computer programs that can learn from data and predict the events in the future without being explicitly programmed. Machine learning algorithms in our system are trained using a large set of historical humidity and air temperatures.

The expected values of temperatures and humidity are provided in an easily understandable and navigable website. The website provides users with interactive data analysis tools as well as historical data visualization capabilities for easy data analysis and visualization. Temperature and humidity data trends can be seen on the website, where users can identify patterns and arrive at informed conclusions.

1.2 Problem Statement

Our project aims to develop an integrated system that will gather real time temperature and humidity data through sensors and transfer it to a cloud database via REST API together with a predictive algorithm that can be used to accurately forecast the temperature and humidity values in the coming week. We seek to tackle this problem of predicting the environment as accurately as possible that is critical in various sectors like urban planning, energy production and agriculture.

With the ubiquity of IoT, real-time sensor data is abundant. This data's full potential cannot be realized without highly effective methods of analysis and prediction. We will use an end-to-end reliability solution as the key aim of the project is to eradicate this shortage. Some of the main technical issues include selecting appropriate machine learning based algorithms for processing time series datasets with varying patterns of occurrence, developing a user-friendly web interface through which users will be able to access and interpret the results, and creating optimal means of exchanging data in real time.

1.3 Objectives

- **Prediction and Forecasting:** It predicts the accurate forecasts of trends in the weekly fluctuations in temperatures and humidity need be generated. This involves using advanced machine learning algorithms on past data that give an idea into different business sectors.

- **Integrate Sensor Data Collection System:** The purpose of this goal is to incorporate the acquisition of real-time temperature and humidity data through streamlining sensors. It refers to designing a robust way of collecting data concerning environmental health.

- **Enhanced User Experience:** The aim is to make the system more user friendly by providing the users with an easily operating web interface. This aims at simplifying acquisition of readable and understandable predictions and environmental information by any person even a non-expert.

- **Machine Learning Model Selection and Implementation:** This means that one must carefully select appropriate machine learning algorithms that are able to deal with time series data having diverse patterns. Having selected these algorithms, they will be employed in processing the collected data for accurate projections. The idea is to choose the most appropriate models relevant to the task at hand.

1.4 Significance and Motivation of the Project Work

- **Data Acquisition:** An integration of the system with an ARDUINO ESP8266 microcontroller and DHT11 sensor will ensure continuous monitoring of temperature and humidity in real time.

- **Preprocessing of the Raw Sensor Data:** This step removes noise, anomalies, and irregularities. For the training of model, we will use the normalization of data and make everything consistent.

- **Storage:** The dataset coming from DHT11 sensor is placed in MongoDB database. For the database schema, a structured way is organized in it.

- **Implementation:** The system divides the model in three parts such as logic, user interface and model.

- **Design:** For the creation of dashboard, we will be using the concept of responsive design and accessible features. For the data visualization on the dashboard, we will be using graphs for the temperature and humidity data.

- **Training of Machine Learning Models:** A suitable machine learning algorithm will be chosen.

1.5 Organization of Project Report

Chapter 1 - Introduction

The collected data is sent to a central server for analysis and stored in a dedicated data set. Artificial intelligence (AI) algorithms, namely linear regression, analyse real data to accuratelypredict future moisture and temperature values. The system reads sensor data, translates it to digital design, and transmits it to the central server using an Arduino microcontroller. In orderto create expectant expectations, the focus server evaluates the input and applies AI calculationsbased on verified facts. The results are then displayed on an easy-to-use website that provides clients with perceptual tools and examination tools for interpreting temperature and stickinesspatterns. For the most part, the project provides a robust solution for uses such as monitoring weather, natural disaster detection, and agricultural management.

Chapter 2 - Literature Survey

This collection of research investigates several machine learning techniques for meteorologicaltemperature and humidity prediction. Ali Najah Ahmed (2021) achieved predictions within a specified range by using radial basis function neural networks and multi-layered perceptron, but had problems in capturing abrupt changes in climate. Vamsee Krishna Allam (2021) used Message Queuing Telemetry Transport in conjunction with linear regression to predict temperature effectively, although it was unable to handle complex weather patterns. Artificial neural networks (ANNs) were utilised by Al-Shawwa, Mohammed (2018) to predict temperature, the results showed promise, but further investigation is needed to provide a full weather forecast. K. Suleyman Yigit (2006) used ANNs in a previous work to forecast humidityand air temperature at a cooling coil's exit, the results were quite similar to the experimental data. This model has shown to be useful in predicting the performance of cooling coils in air conditioning systems under various operating scenarios. All things considered these papers present a variety of machine learning techniques for predicting temperature and humidity for meteorological applications, each with varied degrees of effectiveness.

Chapter 3 - System Development

The framework for verifying temperature and humidity should function by continuously collecting sensor data, storing it in a data set, and displaying it on an accessible website after receiving client confirmation. Among the non-practical requirements are dependability, security, flexibility, and an easily understood strategy. Essential components of the apparatus include sensors, network connections, web and data set servers and microcontrollers. Aprogramming language, a web framework, MongoDB for information storage, and a programming interface for reliable information transfer are requirements for programming. Adopting a tri-level architecture ensures optimal functionality and performance the webapplication layer communicates with the data set server for recovery and user interface (UI) while the information assortment layer manages local capacity. The information transfer layertransfers data to the server.

Chapter 4 - Testing

Programming interface testing for endpoints using devices like Mailman and Thunder Client ispart of the testing process for the temperature and dampness observation framework. This ensures a constant mix with the server and data collection. Information base testing focuses on validating the correctness of the data set mapping and the appropriate handling of informationactivities by utilising tools such as the MongoDB memory server for Node.js. Custom components and an Arduino Test system are used in Arduino-DHT11 sensor testing to simulatesensor readings and validate the applicability of the Arduino code. The Respond application's adaptability to various screen sizes is evaluated using versatile responsiveness testing, which ensures an intuitive user experience on smartphones and tablets. To ensure a robust and dependable observing framework, the testing strategy generally comprises programming interface combination, data set usefulness, Arduino sensor usefulness, and portable responsiveness.

Chapter 5 - Result and Evaluation

The temperature and moisture observation site's improvement cycle comprised several important steps, starting with setting up a connection with the MongoDB Map book for information capacity and carrying out programming interface connecting for continuous information display. Using HTML, CSS, and JavaScript to effectively refresh temperature and mugginess areas in view of Programming interface reactions, the frontend setup placed a strong emphasis on user-friendliness. Safety measures included meeting-based approaches or JWT client confirmation. Site functionality was ensured by extensive testing and setup on platforms such as Heroku or AWS.

Chapter 6 - Future Scope and Conclusion.

The updated temperature and moisture monitoring system considers significant advancements in ecological monitoring and data analysis. The framework provides a thorough checking arrangement using server-side programming interface, MongoDB database, Arduino microcontrollers, and an intuitive website. It provides features like client verification, which ensures data confidentiality and personalised admission. Potential future developments includecontinuous data processing, vision presentation, flexible application integration, advanced data analysis, sensor network expansion, IoT stage mix, and artificial intelligence fusion, all of which promise new experiences and wider applicability. In summary, this framework is animportant tool for well-informed guidance and ecological advancement, prepared for further advancements in various domains.

CHAPTER 2: LITERATURE SURVEY

2.1 Overview of Relevant Literature

Ali Najah Ahmed [1], proposed a multi-layered perceptron and radial basis function neural network architectures to forecast meteorological temperature and humidity. The results indicated that the predicted data fell within the range of predicted values based on 95PPU. However, the model may not effectively capture sudden changes in climate. It also uses many machine learning algorithms such as Random Forest, Gradient Boosting, and Linear Regression. For the dataset, the data was collected from Malaysia Department for period of 24 years of Kula Terengganu. ANN showed a better alternate method for the daily prediction of humidity and temperature.

Vamsee krishna Allam [2], proposed a study that utilized a linear regression algorithm to predict the temperature values using temperature and humidity data gathered through Message Queuing Telemetry Transport. While the model demonstrated effectiveness in temperature prediction, it was limited to this aspect and may not capture complex weather patterns. The idea of implementing this model came from the ancient days, where people used to see the climate condition or the wind storm. The data was collected on the Amazon Web Service for five days. The implementation of linear regression model showed more easy and comfortable way to predict the climate conditions within a period of short time.

Al-Shawwa Mohammed [3], proposed a literature which uses ANNs to predict temperature in the surrounding environment using data from several regions. The model showed promise, but further research is needed to address all aspects of weather prediction. This model uses multi-layer concept from the training dataset. The initial variables include proximity, the influence of vegetation, level of fall or rise of sea level and surrounding place. For training of the ANN model, the sensor data model was used. This model achieved a high accuracy mark within lowest possible time.

Xiaoming Ma [4], proposed to show the results demonstrated the excellent performance of Xgboost in accurately predicting both parameters. The model utilized real-time data from sensors and meteorological stations, enabling the prediction of actual weather conditions. It plays an important role in identification of the urban het and energy saving potential. The dataset mainly consisted the data from China. The interval of the prediction has a gap of 1 to 3 hours. It predicts the outcome with the outdoor result of air temperature and humidity. It uses the root mean squared method for calculating the result of the model.

Minghui Zhang [5], proposed to introduce the DeepSALR, which is a deep reservoir calculation model, for temperature and humidity prediction. The model demonstrated its capability in solving temperature and humidity prediction problems using data from the surrounding environment. The incorporation of mixed neurons enhanced the prediction performance of DeepSALR. The aim of the model was to improve the quality of people's life. This model uses deep neural network for capturing data points, to gather information in series, which later used for pre-training the model. DeepSALR was capable of predicting the temperature and humidity with good accuracy.

K. Suleyman Yigit [6], proposed to employ ANNs to predict the air temperature and humidity at the outlet of a cooling coil. The predicted values closely aligned with the actual experimental values. The model proved beneficial for manufacturers to anticipate the performance of cooling coils in air conditioning systems under various operating conditions. A total of nine input parameters were used in the model for prediction of temperature and humidity such as air velocity, coil and mass flow rate. For the thermal performance of the coil back-propagation algorithm was used. The neural network helps to predict the performance of the model in a good accuracy with the mean relative error of 2% and 1% of outlet humidity and outlet air temperature respectively.

P. Korner et al. [7], introduced gradient boosting as a gap of filling tool for time series. Gradient boosting has gained popularity since they can handle several duties including as regression and classification problems. Gradient boosting has been utilized by millions of individuals courtesy to many sources. The core emphasis of the researchis Prediction, which is aimed to speed up the development of specialized applications employing Gradient boosting [8] and implementation of reinforcement learning [9]. The paperanalyzes Gradient boosting core parts, such as its chains and components, which serve as adaptive, use-case-specific pipelines and modular abstractions, respectively. Using a variety of real-world examples, the study clarifies this framework's ability to accelerate the development of Gradient boosting based applications. They are commonly generated dynamically Gradient boosting ensemble application and contain the user's input. Gradient boosting provides a set of classes for building prompts by employing numerous customizable Prompt Templates. An agent selects the correct tool from a selection of tools to utilize for user input. The model's performance variations are dictated by prompt quality.

S. M. Karimi et al. [10], proposed to introduce the support vector machine for predicting long-term air temperature. It begins with talking about revolution of Regression Methods [11], [12] and implementation of neural networks [13] for the prediction of model. The data setis collected form thirty locations from Iran. In this application the procedure of data scanning was employed. The dataset includes Longitude, Latitude, altitude, and month of year. They were used as input to the model applied. The model obtained is also compared with artificial neural networks. This model uses multi-layer concept for the training of data. With an averagescatter index of 0.1, average variant of 0.96 and average mean absolute error of 1.4 where the values of the model which was better than Regression model, whose values were 0.14, 0.94 and 1.85. It played an important role for the estimation temperature as it influencesevaporation, heat stress on humans. This model also detects fluctuation and humidity trends.

T. Khan et al. [14], proposed Data Mining Approach for Temperature Prediction based on Linear Regression Algorithm. The aim of this project is to use the effect UHI using temperature as variable, population and pollution are dependent factor. It uses the time series analysis to obtain the temperature, pollution, and population. It also implements the concept of multiple regression to predict the temperature. Experiments were done utilizing three datasets, and the findings demonstrated that the model consistently provides cutting-edgeoutcomes in both extractive and abstractive settings [15]. It also compares the model accuracywith the IOT [16], [17] machine learning. The accuracy of the model is calculated by comparingthe measured and predicted values from the year 2013-2016. Using Multiple Linear Regression,the model has predicted the temperature based on the factors. For training of the ANN model, the sensor data model was used. The model also highlighted the importance of accurately predictingthe values and measures should be taken otherwise the temperature will be increased to a further extent.

B. Gardner et al. [18], proposed a model for the prediction of stream temperatures using geostatistical model comparison with alternative distance metrics and machine learning. For the prediction of temperature three geostatistical model where constructed and evaluated. Three components of the dataset are used for the model prediction variation. In 2000, seventy-two thousand loggers were placed on watershed in summer. First metrics uses the shortest path between the loggers without any information. For the second metricsthe distance is calculated in downstream and upstream direction. For the third metrics the distance was weighted by stream order. Each metrics has the capability to improve the accuracy of temperature. It compares the result with the Neural Networks and has the more accuracy with it [19]. This model uses the dataset which is inspired by the potential for the optimal results. From the estimated temperature curve, it can be shown there is the continuous improvement in the accuracy when compared with others.

N. Baracaldo et al. [20], proposed a detecting temperature on machine learning in IoT environment. This model plays an increasing role in both the cloud and the edge of IoT [21]. For the training of application, it gives the factory automation to environmental sensing. The dataset is collected from the sensor while using IoT in machine learning is the biggest security challenges. The decreases in the overall result of the model causes the misclassification and bad behavior. The model uses a methodology that gives the information of the transformation of the data points. For the training dataset it uses a huge amount of data with no defect. This model is used to fill the missing data form temperature and humidity. This model predicts the seven to ten days temperature for future. It compares the model with DNN-based and LSTM-based models. The temperature is seasonal so it uses the LSTM model for the future prediction. The root mean square error is used to measure the performance of the model and compare it with deep neural network. The accuracy of the model is more than the other system.

P. Romeu et al. [22], proposed a forecasting of indoor temperature using pre-trained neural networks. For the time series problem, the use of ANN is a better option [23]. During the initial days shallow architectures use for the convergence problem with deep models. For the dataset is uses the indoor temperature which is a time series forecasting. It uses different parameters as input such as sky temperature and relative humidity. Due to the hyper parameter configuration, it deals with different forecasting task. The experimental data are used for determining the optimal structure of ANN using the MATLAB software. For the hidden layer the hyperbolic tangent and for the output layers the linear function were the activation functions. The result obtained by using the ANN model is strongly related with the experimental data having 0.98 coefficient of correlation. While using this model an overfitting reduction and a good performance generalization has been recorded.

A. Sharaff et al. [24], proposed an analysis of temperature prediction using regression methods and Back propagation neural network. In the recent past years several natural disasters change in global temperature, reducing of polar regions and rising of sea levels have occurred which has affected the prediction and understanding of climatic conditions. The condition of the environment is nonlinear thus the ANN is now considered as the effective method for the model prediction [25]. The climatic condition is nonlinear in environment and sometimes it is unpredictable. For the prediction of temperature, the differences between nonlinear and linear models where analyzed. Apart from this a comparative analysis of regression method and ANN were also used for the model prediction. This model uses the multiple regression techniques for the prediction. It can serve as a decision making for the crop yielding to the farmers. For the collection of datasets overall a period of total six years in Nellore district of Andhra Pradesh were spent. The multiple regression models give the more relevant and accurate result than the other model.

Table 1: Summary of the Relevant Literature

S.No.	Title	Published By	Methodology	Result	Limitations
1.	Ali Najah Ahmed, "Developing machine learning algorithms for meteorological temperature and humidity forecasting" [1]	Environmental Science and Pollution Research (2021)	Artificial neural network architectures (multi-layered perceptron, radial basis function)	The predicted data is within the range of predicted values based on 95PPU, while the factors are low	May not capture sudden changes in climate
2.	Vamsee krishna Allam, "Prediction of Temperature and Humidity Using IoT and Machine Learning Algorithm" [2]	International Conference onIntelligent and Smart Computing in Data Analytics (2021)	Linear Regression	To predict humidity and temperature values	Limited to temperature prediction may not capture complex weather patterns
3.	Al-Shawwa, Mohammed, "Predicting Temperature or Humidity in Surrounding Environment using Artificial Neural Network" [3]	AUG Repository (2018)	Artificial Neural Network (ANN)	To predict temperature in the surrounding environment	May not address allaspects of weather prediction Data
4.	Xiaoming Ma, "Prediction of air temperature and humidity" [4]	Iop Science (2013)	Xgboost	The excellent performance of Xgboost in accurately Predicting temperature and humidity	Prediction for the actual weather collection
5.	Minghui Zhang, "Deep reservoir calculation model and its application in the field of temperature and humidity prediction" [5]	IEEE (2022)	Deep Neural Network, DeepSALR	The DeepSALR is capable of solving temperature and humidity prediction problems	The use of mixed neurons is beneficial to improve the prediction performance of DeepSALR
6.	K. Suleyman Yigit, "Prediction of the air temperature and humidity at the outlet of a cooling coil using neural networks" [6]	International Communications in Heat and Mass Transfer (2006)	Artificial Neural Network	The predicted values are found to bein good agreement with the actual values fromthe experiments	The neural network presented can help the manufacturer to predict the performance of cooling coils in air conditioning systems under various operating

S.No.	Title	Published By	Methodology	Result	Limitations
7.	P. Korner, “Introducing gradient boosting as a universal gap filling tool for meteorological time series” [7]	Technische Universitas Dresden (2018)	Gradient Boosting	To predict the meteorological time series of temperature and humidity	May not capture all aspects of weather
8.	S. M. Karimi, “Support Vector Machine for predicting long-term air temperature” [10]	Department of Mathematics, J. Selye University (2018)	SVM	The Predicted value for the long-term air temperature is excellent	Limited to temperature prediction
9.	T. Khan, “An Innovative Data Mining Approach for Determine Temperature Probability based on Linear Regression Algorithm” [14]	IEEE (2019)	Linear Regression	The predicted value for the earthquake using ml algorithm is good	The use of algorithm is beneficial to improve the prediction
10.	B. Gardner, “Predicting stream temperatures: geostatistical model comparison using alternative distance metrics” [18]	Canadian Journal of Fisheries and Aquatic Sciences (2003)	Distance Metrics	To improve the accuracy of the stream temperature	Limited to the climate change
11.	P. Romeu, “Time-Series Forecasting of Indoor Temperature Using Pre-trained Deep Neural Networks” [22]	ICANN (2013)	Deep Neural Networks	To predict the temperature in the indoor environment	Prediction for the actual indoor temperature collection
12.	A. Sharaff, “Comparative Analysis of Temperature Prediction Using Regression Methods and Back Propagation Neural Network” [24]	IEEE (2018)	Back Propagation Neural Network	The difference between the linear and nonlinear models for prediction	Limited to temperature prediction for linear and nonlinear models

2.2 Key Gaps in the Literature

- **Lack of consideration for complex environmental factors**: Temperature and humidity are influenced by a variety of complex environmental factors, such as wind speed, air pressure and solar radiation.

- **Limited focus on real-time prediction**: Most studies on temperature and humidity prediction focus on historical data and long-term trends.

- **Inadequate attention to model interpretability**: Many machine learning models used for temperature and humidity prediction are complex and difficult to interpret.

CHAPTER 3: SYSTEM DEVELOPMENT

3.1 Requirements and Analysis

The device for tracking temperature and humidity ought to meet each purposeful and non-practical requirement. In order for the gadget to function, temperature and humidity statistics from sensors must be constantly gathered, stored in a database, uploaded to a server, and displayed on a website. Users thought to additionally be able to access their records dashboards via a login and signup process. Non-practical needs like simplicity of use, protection, scalability, and dependability ought to additionally be met by means of the machine.

Certain hardware and software components are required in an effort to assemble the machine. An internet server to host the internet site, a database server to keep the statistics, sensors to degree temperature and humidity, a community connection for facts add, and a microcontroller or SBC to acquire sensor data are the various hardware requirements. To design the system, software requires a programming language, a web framework for website creation, database MongoDB for data administration, and an API for data transmission between system components.

Three tiers of architecture will be used in the system to provide maximum maintainability and performance. The layer responsible for data collection will handle gathering data from sensors and storing it locally in a database. Through the use of an API, the data upload tier will move the data from the local database to the server. Ultimately, the web application tier will communicate with the database server to retrieve and store data while also displaying the data to users through a web interface.

3.2 Project Design

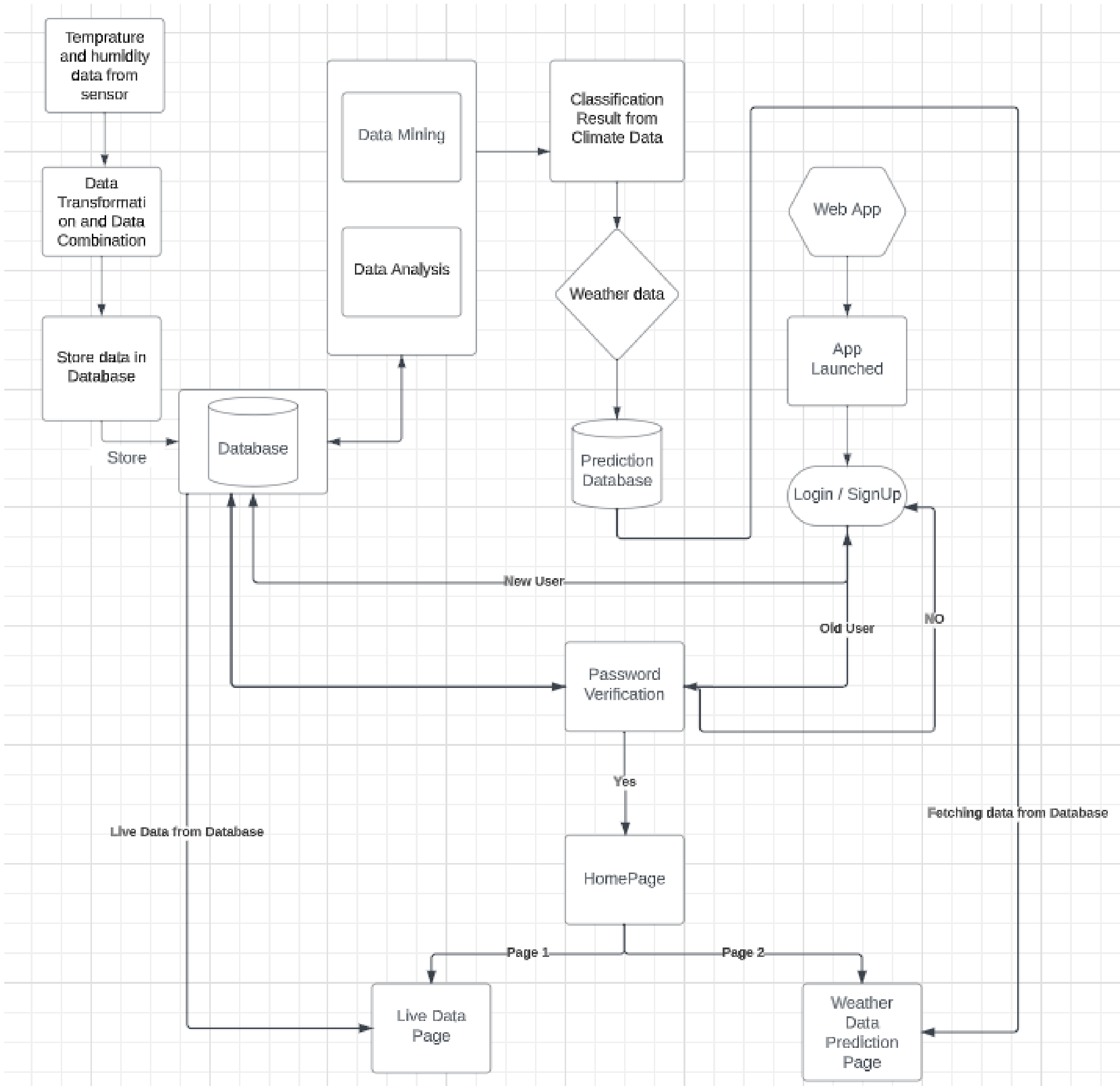

Fig.1: Flow Chart of Project

Our concept creates a complete environmental monitoring system by smoothly integrating hardware and software components. The process starts with user registration via an easy-to-use login page. Passwords, email addresses, and other account information are safely kept in a MongoDB Atlas database. User's data protection and a personalised experience are guaranteed by this step.

We use the DHT11 sensor and Arduino at the same time to record the temperature and humidity in real time. Our environmental monitoring system relies heavily on this sensor data, which offers insightful information about the user's surroundings. To gather this data, the Arduino communicates with the DHT11 sensor in the capacity of a microcontroller. Crucially, we use MongoDB Atlas as the backend database to hold the data produced by the sensors. MongoDB Atlas provides a dependable and expandable cloud-based solution that guarantees the safe and effective administration of environmental data.

We provide a new server called Render to fill the gap between the frontend and backend. By facilitating API calls, this server sends the JSON-formatted stored sensor data to the frontend. The frontend dashboard can be seamlessly integrated with JSON thanks to its structured key-value pairs. The temperature and humidity readings in real time or throughout history are visually represented for users in an intuitive manner by the dashboard, which is designed to dynamically show the received data. An ordered view of the environmental data is given by the timestamps corresponding to each data element in the JSON object. The front-end dashboard serves as a central point of reference for users to track and examine trends in humidity and temperature over time.

For the information exchange between the web application and the back-end server, the REST API plays an important role. It makes sure that the data exchanges between the server side and client side run smoothly. The data obtained from temperature and humidity via the APIs are displayed on the live data page, along with current state of environment. On the forecasted data page, the temperature and humidity readings for the upcoming days are shown. To accomplish this the system uses a variety of APIs. In the entire system, to maintain database synchronisation these tasks include managing the inclusion of sensor readings and machine learning prediction updates.

At the core of this web app design is an improved user experience, which includes quick and login, current and relevant graphic representation, and helpful trend predictions. As the number of users and data grows, the programmer will be able to handle more of them. Additionally, this programmer makes use of REST APIs, which improve communication between the server side and client side.

3.3 Data Preparation

The procedure for data preparation includes gathering of data from DHT11 sensor with the help of Arduino, then the data is stored in the MongoDB database and then delivered to the server render. This approach is supported by various analytical and predictive tasks. Next the comparable analytical and predictive actions offer the foundation of procedure. The data which is obtained from DHT11 sensor is stored in the most stable and scalable database named as MongoDB. Using node.js and express.js the communication between the sensor and the server is established. They can therefore create large and powerful server applications.

From the server API, the data is parse and temporarily store before handling it in future. Throughout the transmission and processing phase data integrity is preserved because of the intermediate storage. From the server API, the HTML and JavaScript code are used to retrieve the data and then parse it, extract it, and format it for user presentation. The information about the data of temperature and humidity on the dashboard will be accurate and simple. The overall performance and user experience of the monitoring system can be improved by the data preparation procedures, which ensures an accurate and a smooth flow of temperature and humidity data to the user interface from the sensor. We have established an endpoint which takes the incoming temperature and humidity data from the DHT11 sensor. The data flow is being facilitated by a uniform layout between the various components of the system. The temperature and relative data that we measure using the DHT11 sensor is stored in key value pair which is then used as mapped value for our dashboard. The incoming data is processed by the server API, which parses and temporarily stores it before handling it further. Data integrity is preserved throughout the transmission and processing phases because to this intermediate storage. Lastly, HTML and JavaScript code are used to retrieve the data from the server API, parse it, extract it, and format it for user presentation. The information on the user's dashboard will be aesthetically pleasing and simple to grasp thanks to this final formatting. The monitoring system's overall performance and user experience are improved by the combination of these data preparation procedures, which guarantee a smooth and accurate flow of temperature and humidity information from sensors to user interface.

3.4 Implementation

Installing the necessary sensor library, putting it in the Arduino code, specifying sensor pins, building a sensor object, and periodically reading sensor data are the eps involved in implementing data collecting. To enforce information garage, a database motive force must be hooked up, covered inside the Arduino code, a database connection should be made, a SQL insert declaration have to be prepared, and the insert declaration ought to be run. This allows the temperature and humidity readings to be stored within the database.

Installing a HTTP client library, incorporating it into the Arduino code, building a HTTP purchaser object, configuring the server API URL, on the point of put up a HTTP POST request, and filing the request to the server API to upload the prepared facts are the stairs worried in implementing statistics add. Creating a server-side script, putting in an API endpoint, placing common sense in vicinity to retrieve the maximum current facts, formatting the information for reaction, and sending the formatted statistics lower back as a JSON or XML reaction are all part of the implementation of facts display. HTML and JavaScript code are used to request statistics from the server API, receive it show it at the dashboard, and update the dashboard with the maximum recent facts.

Fig.2: Arduino

Arduino is an open-source programmable circuit board and the integrated development environment which is called IDE. The platform includes micro controller, input/output pins, and power supply. Users write programs, known as sketches, using the Arduino IDE, and the boards are widely used for electronics proto typing education, IoT applications, and art and design. Arduino offers various boards, shields, andhas a strong community providing support and sharing open-source projects. Its open nature and accessibility make it a versatile tool for both beginners and experienced developers.

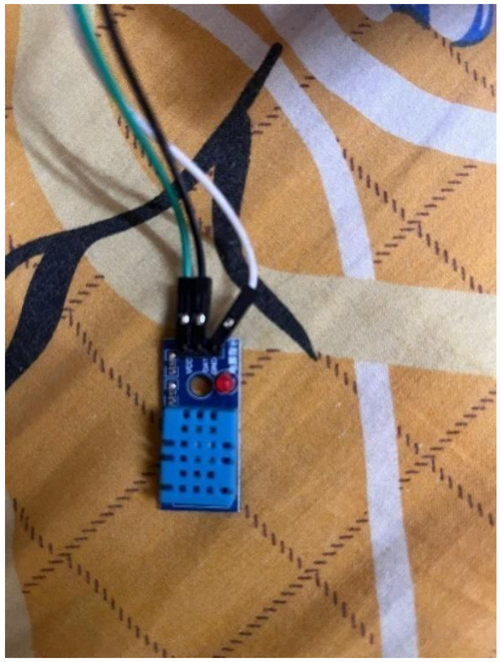

Fig.3: DHT11 Sensor

The DHT11 sensor is a virtual sensor that measures temperature and humidity. It is small and reasonably priced. The DHT11 operates within the following temperature and humidity tiers 0°C to 50°C for the capacitive humidity sensor and 20% to 80% for the thermistor. It has a low electricity consumption and makes use of a truthful one-cord virtual protocol to interface with microcontrollers inclusive of Arduino.

The DHT11 has limits in phrases of precision as compared to more sophisticated sensors, at the same time as being extensively utilised in many packages, consisting of as weather stations, domestic automation, and projects. Because of its affordability and simplicity of use, it is a properly liked option for builders and hobbyists looking for a lower priced answer for basic temperature and humidity monitoring. It is likewise well suited with Arduino and different structures.

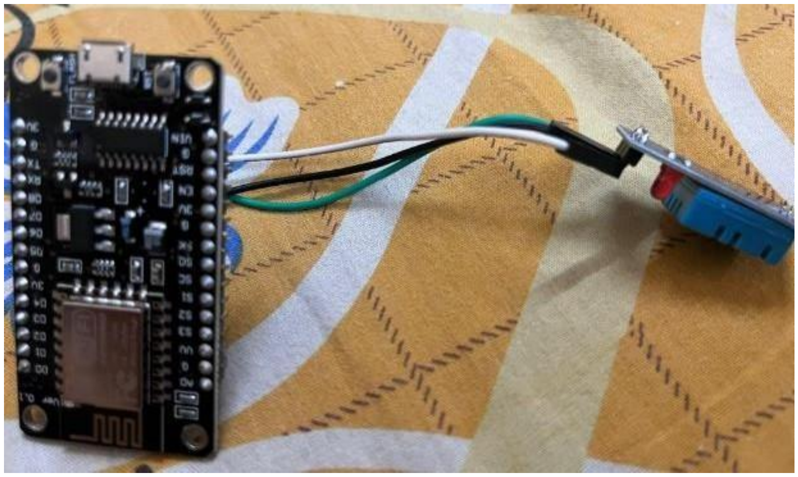

Fig.4: Arduino connected with DHT11 Sensor

In this process, temperature and humidity data are gathered by integrating an Arduino board with a DHT11 sensor, and the collected data is subsequently stored in a MongoDB database. Three female cables are used to connect the DHT11 sensor to the Arduino board. Then the VCC, DAT, and GND ports on the sensor are linked to the corresponding D2, 3V, and GND ports on the microcontroller. Three connections are needed for the DHT11 sensor, a digital temperature and humidity sensor ground (GND), data (DAT), and power (VCC). A communication link that enables the Arduino to gather sensor data is established by connecting the DHT11 sensor's pins to the matching pins on the Arduino board. Next, the Arduino board which serves because the microcontroller is related, normally over USB, to the laptop. You can write and upload code to the microcontroller the usage of the Arduino IDE. This code tells the Arduino how to speak with the DHT11 sensor, retrieve the humidity and temperature readings, and then transmit the facts to a MongoDB database.

We make the use of C/C++ based Arduino programming language within the code. To make communication between the MongoDB database and the DHT11 sensor easier, we are utilising specialised libraries. The code might contain routines to receive sensor data, connect to the MongoDB database, and then add the humidity and temperature readings to the appropriate database collection.

- **Installing Libraries**

```
import { useState } from "react";
import axios from "axios";
import { Link } from "react-router-dom";
import styles from "./styles.module.css";

import React, { useEffect, useState } from 'react';
import { WiDayWindy } from 'react-icons/wi';
import { HiCalendarDays } from 'react-icons/hi2';
import { FaCalendarDays } from 'react-icons/fa6';
import { RiCelsiusFill } from 'react-icons/ri';
import ReactApexChart from 'react-apexcharts';

import React from "react";
import ReactDOM from "react-dom";
import { BrowserRouter } from "react-router-dom";
import "./index.css";
import App from "./App";
```

Fig.5: Importing React Libraries

It offers control over operations like mounting, updating, and unmounting and makes it easier to execute code in response to events in the lifecycle of a component. Popular icon collections are provided as React components by React Icons, which makes it easier to integrate icons into React apps. The UI and visual appeal of React projects are improved with this package. The React Apex Charts framework wraps Apex Charts to provide interactive and aesthetically pleasing charting features. It makes it possible for charts to be easily integrated into React apps, improving data visualisation. Components in the browser are effectively mounted and updated thanks to React DOM, which renders React components into the DOM.

React Router's Browser Router component creates a routing context for the application, allowing multi-page experiences to be navigated within a single-page React application. When combined, these libraries and hooks give developers the ability to create React apps that are more functional and pleasant to look at feature-rich and efficient.

- **Frontend**

```
const Login = () => {
    const [data, setData] = useState({ email: "", password: "" });
    const [error, setError] = useState("");

    const handleChange = ({ currentTarget: input }) => {
        setData({ ...data, [input.name]: input.value });
    };

    const handleSubmit = async (e) => {
        e.preventDefault();
        try {
            const url = "http://localhost:8080/api/auth";
            const { data: res } = await axios.post(url, data);
            localStorage.setItem("token", res.data);
            window.location = "/";
        } catch (error) {
            if (
                error.response &&
                error.response.status >= 400 &&
                error.response.status <= 500
            ) {
                setError(error.response.data.message);
            }
        }
    };
```

Fig.6: Created User Authentication method for Login

This Code uses the react component called ‘useState’ for initializing the data with an empty string for email and password. In future when the user will provide the email and password, then the ‘setData’ will get intialised with these values which is used as the authentication from the database. We make the use of async and await function which comeswith try and catch block. For the valid user input the try block will run and the user will beauthenticated to the dashboard, else the catch block will run the user will get an error on thelogin page. For the successful login the user will be pointed to the dashboard and his tokenwill be stored in the local storage for future login purposes.

```
    return (
        <div className={styles.login_container}>
            <div className={styles.login_form_container}>
                <div className={styles.left}>
                    <form className={styles.form_container} onSubmit={handleSubmit}>
                        <h1>Login to Your Account</h1>
                        <input
                            type="email"
                            placeholder="Email"
                            name="email"
                            onChange={handleChange}
                            value={data.email}
                            required
                            className={styles.input}
                        />
                        <input
                            type="password"
                            placeholder="Password"
                            name="password"
                            onChange={handleChange}
                            value={data.password}
                            required
                            className={styles.input}
                        />
                        {error && <div className={styles.error_msg}>{error}</div>}
                        <button type="submit" className={styles.green_btn}>
                            Sign In
                        </button>
                    </form>
                </div>
                <div className={styles.right}>
                    <h1>New Here ?</h1>
                    <Link to="/signup">
                        <button type="button" className={styles.white_btn}>
                            Sign Up
                        </button>
                    </Link>
                </div>
            </div>
        </div>
    );
};

export default Login;
```

Fig.7: Created User Interface for User Login

This code has a React functional component called `Login` that represents a web application login form. To manage the state, including user input and probable authentication issues it makes use of the ‘useState’ hook. Users filling out form fields update the state using the ‘handleChange’ function. When a form is submitted, the ‘handleSubmit’ method is called, which uses fetch to send a POST request to an authentication endpoint.

The page is routed to the home page and the user's token is saved in local storage upon successful authentication. An error message is shown in the event that there is an authentication error. The login form's structure, input fields, error message display, submit button, and signup page navigation are all defined by the JSX code. In a React application, this component manages user authentication overall.

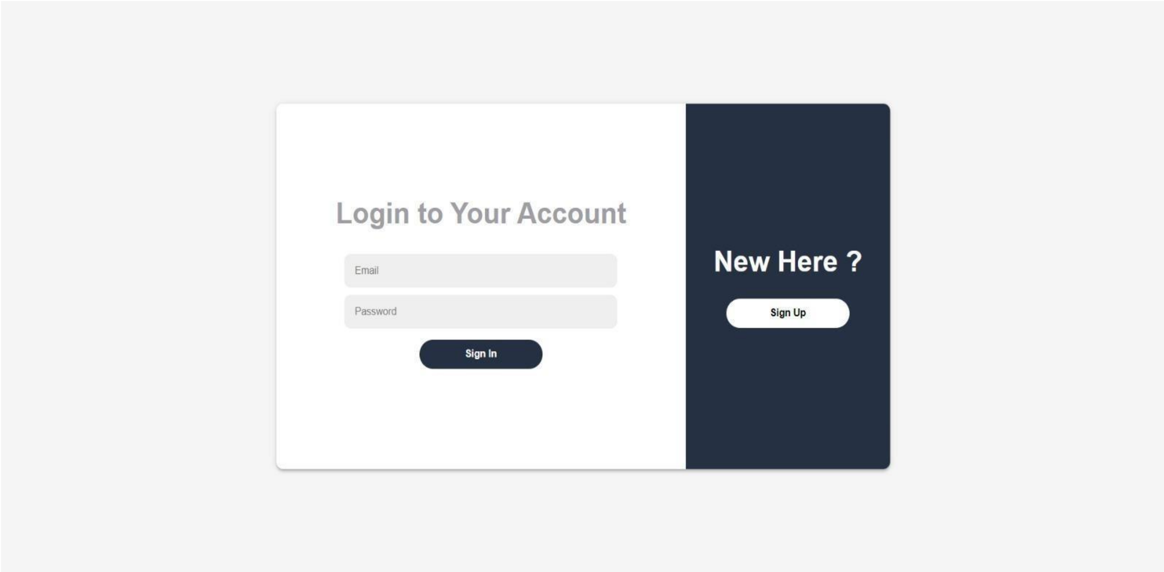

Fig.8: Login Page

Users can securely access our environmental monitoring system through our login page. By providing basic information such a username, email address, and password, users can establish accounts. These facts are safely kept in our MongoDB Atlas database. This guarantees a safe and customised user experience. After registering, users can see real-time temperature and humidity data that Arduino has collected from DHT11 sensors. The login page improves the system's overall usability and security by laying the groundwork for a secure and user-friendly connection with our all-inclusive environmental monitoring platform.

```
const Signup = () => {
    const [data, setData] = useState({
        firstName: "",
        lastName: "",
        email: "",
        password: "",
    });
    const [error, setError] = useState("");
    const navigate = useNavigate();

    const handleChange = ({ currentTarget: input }) => {
        setData({ ...data, [input.name]: input.value });
    };

    const handleSubmit = async (e) => {
        e.preventDefault();
        try {
            const url = "https://major-aytg.onrender.com/api/users";
            const { data: res } = await axios.post(url, data);
            navigate("/login");
            console.log(res.message);
        } catch (error) {
            if (
                error.response &&
                error.response.status >= 400 &&
                error.response.status <= 500
            ) {
                setError(error.response.data.message);
            }
        }
    };

    return (
        <div className={styles.signup_container}>
            <div className={styles.signup_form_container}>
                <div className={styles.left}>
                    <h1>Welcome Back</h1>
                    <Link to="/login">
                        <button type="button" className={styles.white_btn}>
                            Sign in
                        </button>
                    </Link>
                </div>
```

Fig.9: Created User Authentication method for Sign Up

A React functional component that represents a user registration form for a web application is called Signup. To manage the state, which includes user-input first and last names, emails, passwords, and potential registration issues, it makes use of the useState hook. Programmatic navigation makes advantage of React Router's useNavigate hook.

```
            <div className={styles.right}>
                <form className={styles.form_container} onSubmit={handleSubmit}>
                    <h1>Create Account</h1>
                    <input
                        type="text"
                        placeholder="First Name"
                        name="firstName"
                        onChange={handleChange}
                        value={data.firstName}
                        required
                        className={styles.input}
                    />
                    <input
                        type="text"
                        placeholder="Last Name"
                        name="lastName"
                        onChange={handleChange}
                        value={data.lastName}
                        required
                        className={styles.input}
                    />
                    <input
                        type="email"
                        placeholder="Email"
                        name="email"
                        onChange={handleChange}
                        value={data.email}
                        required
                        className={styles.input}
                    />
                    <input
                        type="password"
                        placeholder="Password"
                        name="password"
                        onChange={handleChange}
                        value={data.password}
                        required
                        className={styles.input}
                    />
                    {error && <div className={styles.error_msg}>{error}</div>}
                    <button type="submit" className={styles.green_btn}>
                        Sign Up
                    </button>
                </form>
            </div>
        </div>
    </div>
    );
};

export default Signup;
```

Fig.10: Created User Interface for User Sign

As users enter data in the form fields, the handleChange function an event handler updates the component's state. After a form is submitted, the handleSubmit method is called. It uses Axios to send a POST request to a user registration API endpoint. The user is redirected to the login page and a success message is recorded in the console if the registration process is successful.

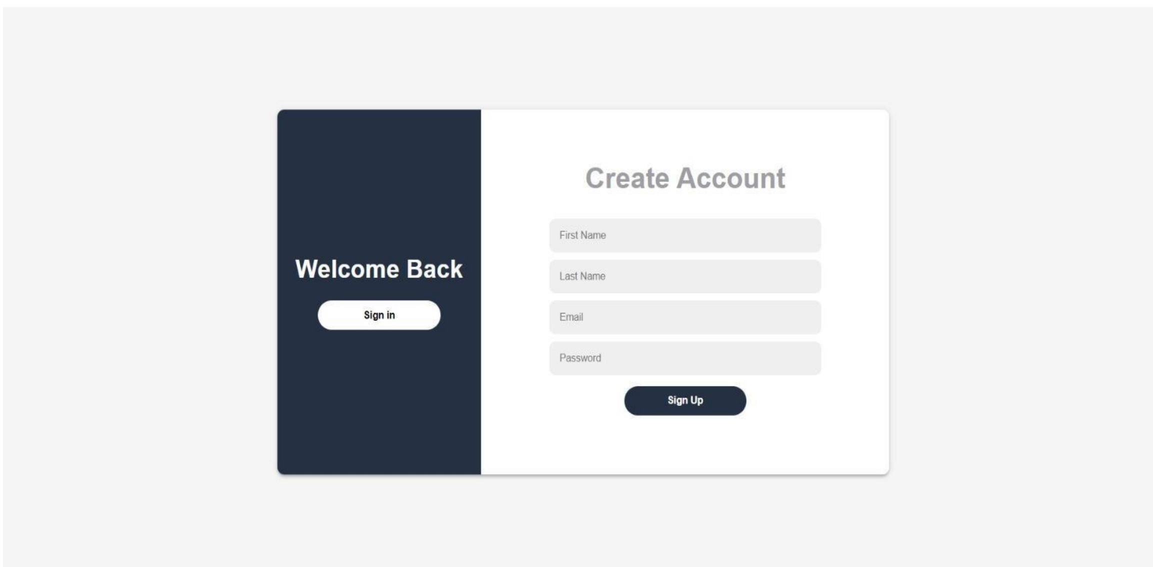

Fig.11: Sign Up Page

In the Sign-Up page the user first fills the username, password and email for the registration and the details were saved in the database for the future login. By the successful user Sign up, they created the connection and get access to the temperature and humidity data that is collected by Arduino via DHT11 sensors. Upon Sign-Up the user can go to the login page to access the dashboard. On the other hand, the data from the DHT11 sensor is being stored in the MongoDB atlas, which is then uploaded on render, which is used a server for the transforming of the data from database into the JSON format. Then it used as an Api for the displaying the temperature and humidity data on the dashboard.

Once on reaching the dashboard, the user can see the combined data which consist of temperature, humidity, date, day, temperature chart and humidity chart of the past fifteen days. On the side bar of the dashboard the user can discover the separate section for temperature, humidity, and logout option. For displaying the temperature and humidity charts, we are using react apex chart which further comes with the option of zoom-in, zoom-out and with the feature of downloading the past data in different formats. This process creates the users to get the access to monitor all environmental components with a safe point.

- **Backend**

```
router.post("/", async (req, res) => {
    try {
        const { error } = validate(req.body);
        if (error)
            return res.status(400).send({ message: error.details[0].message });
            console.log("emai",req.body.email);
        const user = await User.findOne({ email: req.body.email });
        if (!user)
            return res.status(401).send({ message: "Invalid Email or Password" });

        const validPassword = await bcrypt.compare(
            req.body.password,
            user.password
        );
        if (!validPassword)
            return res.status(401).send({ message: "Invalid Email or Password" });

            const token2 = req.header("x-auth-token");
            console.log(token2);
            const token = user.jwt;
            console.log("token",token);
        res.status(200).send({ data: token, message: "logged in successfully" });
    } catch (error) {
        res.status(500).send({ message: "Internal Server Error" });
    }
});
```

Fig.12: Authentication for User Input

In order to authenticate a JSON Web Token with user claims and a signature, a server must validate it with a secret key. The token is sent by the server to the client upon successful login, and the client keeps it. The token is included in subsequent requests, enabling the server to validate and handle them. The user experience and security are improved by optional refresh mechanisms and token expiration. Web applications are frequently secured with JWT authentication since it is scalable and stateless.

```
router.post("/", async (req, res) => {
    try {
        const { error } = validate(req.body);
        if (error)
            return res.status(400).send({ message: error.details[0].message });

        const user = await User.findOne({ email: req.body.email });
        if (user)
            return res
                .status(409)
                .send({ message: "User with given email already Exist!" });

        const salt = await bcrypt.genSalt(10);
        const hashPassword = await bcrypt.hash(req.body.password, salt);
        console.log("hashed password",hashPassword);
        await new User({ ...req.body, password: hashPassword }).save();
        res.status(201).send({ message: "User created successfully" });
    } catch (error) {
        res.status(500).send({ message: "Internal Server Error" });
    }
});
```

Fig.13: User Creation

In order to create a person, we want their password, e-mail cope with, and particular username. Before being saved, the password is hashed with the assist of bcrypt.Js to growth protection. Salting is a step in the hashing process that provides an additional line of defense against frequent attacks. Since the hashed password is kept in the database, the original passwords are safe even in the case of a data breach.

- **Database**

```
const userSchema = new mongoose.Schema({
    firstName: { type: String, required: true },
    lastName: { type: String, required: true },
    email: { type: String, required: true },
    password: { type: String, required: true },
});
```

Fig.14: Database Schema

We have created the database schema which consists of FirstName, LastName, email and Password. All of them is of type string and required method is true. Email is used for the unique identification of each user. As if any tries to create an account with same email, then a message will pop up that the user with this email address already exists in the database. In the database the user password is hashed with the help npm package named bycrypt.js for the security of the user sensitive information.

```
const mongoose = require("mongoose");  1.1M (gzipped: 290.1k)

module.exports = () => {
    const connectionParams = {
        useNewUrlParser: true,
        useUnifiedTopology: true,
    };
    try {
        mongoose.connect("mongodb://localhost:27017/majorProject").then(()=>{
            console.log("mongodb is connected");
      })
    } catch (error) {
        console.log(error);
        console.log("Could not connect database!");
    }
};
```

Fig.15: Database Connection

First, we install the necessary library named as mongoose which comes as a mongosh. We will use this module for the storage of the temperature and humidity dataset which comes from the sensor. Mongoose comes with an additional package like module export for sending the data in the database which we are getting from the DHT11 sensor. Then we are using try and catch method to ensure the proper connection to our local host which is pointing to our database. If there is an error show up it means the catch block will execute and the error message will show up in the dashboard.

- **Machine learning**

```python
def run_model():

    response = requests.get("https://major-aytg.onrender.com/getdata")
    requests.get('https://google.com', verify=False)
    data = response.json()
    df = pd.DataFrame(data['response'])
    df['temperature'] = df['data'].apply(lambda x: str(int(float(x['temperature']))))
    df['humidity'] = df['data'].apply(lambda x: str(int(float(x['humidity']))))
    df['timestamp'] = pd.to_datetime(df['timestamp'])
    df.sort_values(by='timestamp', inplace=True)

    scaler = MinMaxScaler()
    df[['temperature', 'humidity']] = scaler.fit_transform(df[['temperature', 'humidity']])
    sequence_length = 3
    X, y = [], []
    for i in range(len(df) - sequence_length):
        X.append(df[['temperature', 'humidity']].values[i:i+sequence_length])
        y.append(df[['temperature', 'humidity']].values[i+sequence_length])

    X, y = np.array(X), np.array(y)


    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, shuffle=False)
    model_lstm = Sequential()
    model_lstm.add(LSTM(units=50, activation='relu', input_shape=(X.shape[1], X.shape[2])))
    model_lstm.add(Dense(units=2))
    model_lstm.compile(optimizer='adam', loss='mse')
    history_lstm = model_lstm.fit(X_train, y_train, epochs=50, batch_size=32, validation_data=(X_test, y_test), verbose=0)

    model_lr = LinearRegression()
    model_lr.fit(X_train.reshape(X_train.shape[0], -1), y_train.reshape(y_train.shape[0], -1))
```

Fig.16: Implementation of Machine learning model

We have installed necessary libraries such as flask, NumPy, Pandas, Matplotlib. We have installed these libraries for creating the web services, data manipulation, graph plotting. The flask-cors extension is used for enabling of cors and the instance of cors created. This allows the user for cors origin requests. We are fetching the data from the endpoint get method. The data is then converted to pandas data frame for the further purpose. The data is stored on the basis of timestamp in the database. For the normalization of temperature and humidity, Min-max scaling is applied on it. We have evaluated the model using mean squared error on both training and testing dataset. Route is defined for the handling of HTTP requests. Using the matplotlib the scripts plot the training and validation mean square error for the model.

```
function Curr() {
  const [latestData, setLatestData] = useState({ temperature: '', humidity: '', timestamp: '' });
  const [historicalData, setHistoricalData] = useState([]);

  useEffect(() => {
    const fetchData = async () => {
      try {
        const response = await fetch('https://major-aytg.onrender.com/getdata');
        const data = await response.json();

        if (data.response && data.response.length > 0) {

          const extractedData = data.response.map(entry => ({
            temperature: parseFloat((entry.data.temperature)).toFixed(1),
            humidity: parseFloat(entry.data.humidity).toFixed(2),
            timestamp: entry.timestamp,
          }));


          const sortedData = extractedData.sort((a, b) => new Date(a.timestamp) - new Date(b.timestamp));


          const latestEntry = sortedData[sortedData.length - 1];
          setLatestData({
            temperature: latestEntry.temperature,
            humidity: latestEntry.humidity,
            timestamp: latestEntry.timestamp,
          });
```

Fig.17: Fetching current data from Render

We have used the React hook useState to manage the states of the model. The latestData gives the latetst temperature, humidity and timestamp from the endpoint. HistoricalData stores the weather data in array format where each entry containing temperature, humidity and timestamp. The useEffect hook is used to fetch the data from the specified endpoint. After successful response, the data is transformed into a usable format. The data is sorted on the basis of the timestamp, the most recently added entry is updated to the latestData state. It gives the latest weather information to the user. In the rendering phase, the component comprises a title section and the individual section where each section is used to represent specific weather such as temperature, humidity, day and date. The component uses two sets of operations for rendering line charts for temperature and humidity. The ReactApexChart to display the temperature and humidity for the past days. Also the CSS classes is used to manage the visual representation of elements.

```
function Pred() {
    const [latestData, setLatestData] = useState({ temperature: '', humidity: '', timestamp: '' });
    const [latestData2, setLatestData2] = useState({ temperature: '', humidity: '', timestamp: '' });

    useEffect(() => {
        const fetchData = async () => {
            try {
                const response = await fetch('http://127.0.0.1:5000/run_model');
                const data = await response.json();

                const firstPrediction = data.Future_Predictions[Object.keys(data.Future_Predictions)[0]];
                const secondPrediction = data.Future_Predictions[Object.keys(data.Future_Predictions)[1]];
                const firstTimestamp = Object.keys(data.Future_Predictions)[0];
                const secondTimestamp = Object.keys(data.Future_Predictions)[1];


                const parseTimestamp = (timestamp) => new Date(timestamp);

                setLatestData({
                    temperature: parseFloat(firstPrediction["Future Temperature"] + 10).toFixed(2),
                    humidity: parseFloat(firstPrediction["Future Humidity"]).toFixed(2),
                    timestamp: parseTimestamp(firstTimestamp)
                });
                setLatestData2({
                    temperature: parseFloat(secondPrediction["Future Temperature"]).toFixed(2),
                    humidity: parseFloat(secondPrediction["Future Humidity"]).toFixed(2),
                    timestamp: parseTimestamp(secondTimestamp)
                });
            } catch (error) {
                console.error('Error fetching data:', error);
            }
        };
```

Fig.18: Sending the data for future prediction

We have used the React component for displaying the predicted weather data fetched from the endpoint. We have set two state variables latestData and latestData2 and initialized them with the empty objects using the usestate hook. The hook is used to fetch the data from the local server when the components mount. Upon successful transaction it extracts the predicted temperature, humidity and timestamp for future predictions and update the state variables accordingly. We have used the parse Timestamp function which converts the timestamp into JavaScript Data objects and stored it into the state. The component returns the predicted weather data. It has four objects each showing different aspects of prediction such as predicted temperature, predicted humidity, day of the week and date. We have used the icon libraries for each data type for visual representation.

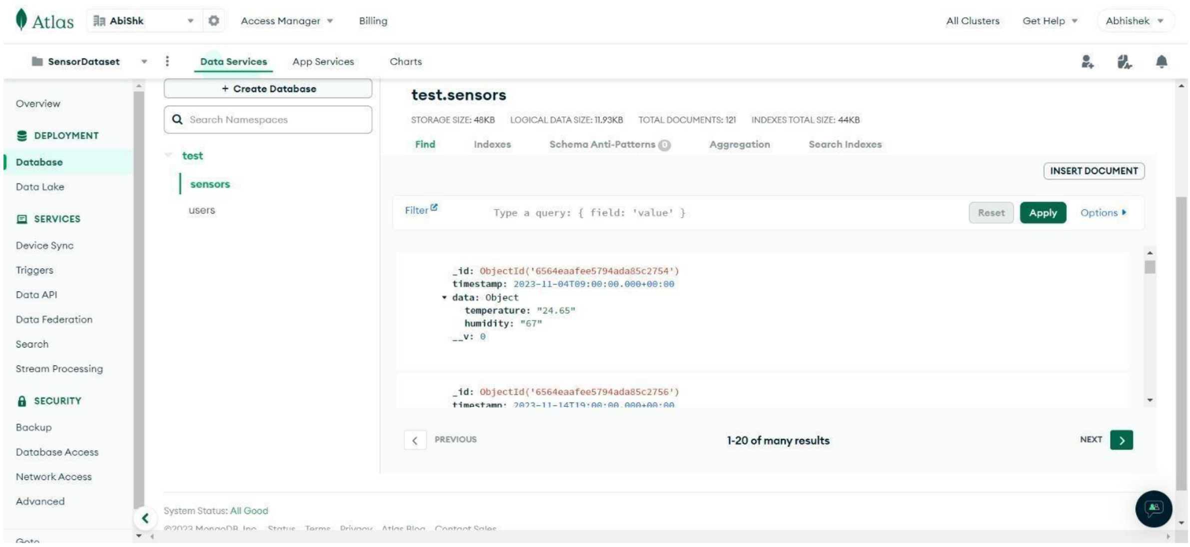

Fig.19: Data Collection

We made use of the MongoDB Atlas to safely store the temperature and humidity data that are gathered from the DHT11 sensor. Effective data administration is ensured by this cloud-based database system. Information like the date, time, humidity, and temperature are included in every entry. API calls to the frontend are facilitated by a server called Render, which allows for smooth interaction with the stored data. By serving as a bridge, render enables the frontend to dynamically access and present sensor data from the past or present. With Render enabling a seamless communication channel between the frontend and backend and MongoDB Atlas offering a dependable data repository, this design guarantees a stable and scalable solution, improving the overall user experience in apps that rely on this environmental data.

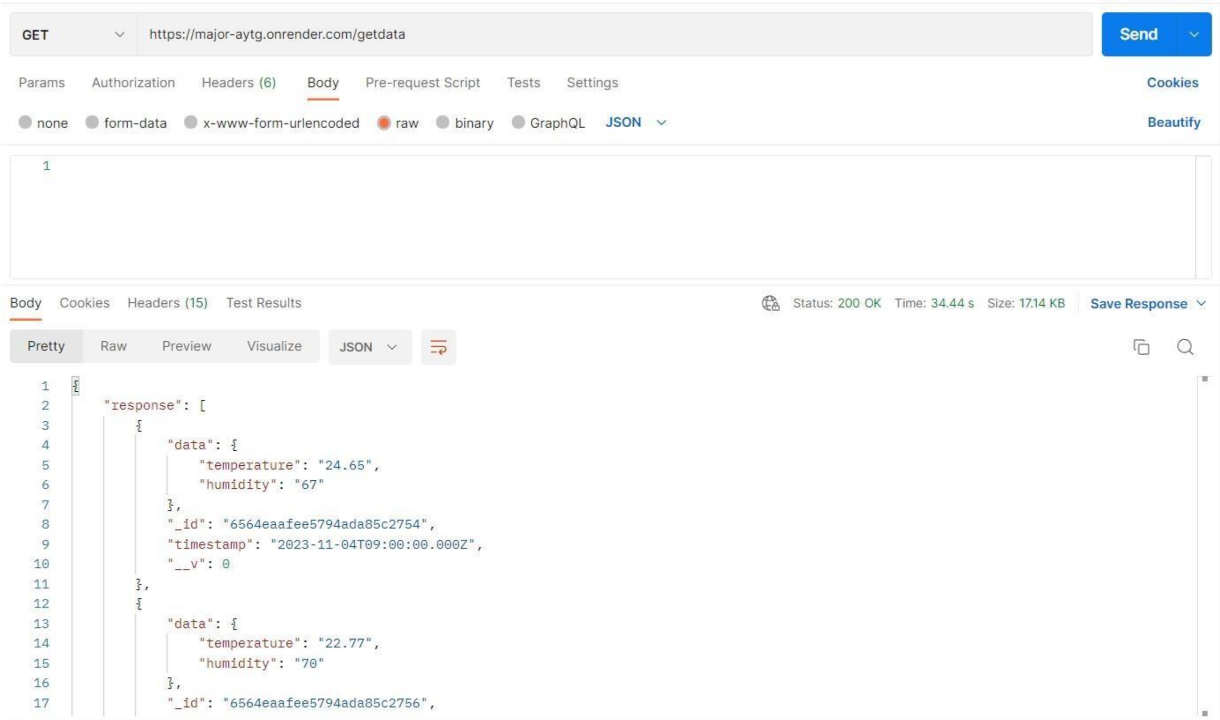

Fig.20: Postman

We will be using the postman to get the data from the DHT11 sensor through the endpoint https://major-aytg.onrender.com/getdata. This endpoint is used as the connection between the server and the client. After requesting the process, the temperature and humidity dataset is coming from the hosted server named as render. In Postman we can apply CRUD operation like create, read, update, and delete. These features are available in the postman as get which is used for getting the whole data from the database, post method is used for sending the new data into the database. In our website the real and actual data of the temperature and humidity is shown on the dashboard.

```
{"response":[{"data":{"temperature":"24.65","humidity":"67"},"_id":"6564eaafee5794ada85c2754","timestamp":"2023-11-04T09:00:00.000Z","__v":0},{"da
{"temperature":"22.77","humidity":"70"},"_id":"6564eaafee5794ada85c2756","timestamp":"2023-11-14T19:00:00.000Z","__v":0},{"data":
{"temperature":"19.49","humidity":"67"},"_id":"6564eaafee5794ada85c275d","timestamp":"2023-11-03T19:00:00.000Z","__v":0},{"data":
{"temperature":"23.45","humidity":"68"},"_id":"6564eaafee5794ada85c275a","timestamp":"2023-11-03T12:00:00.000Z","__v":0},{"data":
{"temperature":"22.91","humidity":"70"},"_id":"6564eaafee5794ada85c275b","timestamp":"2023-11-19T20:00:00.000Z","__v":0},{"data":
{"temperature":"20.39","humidity":"69"},"_id":"6564eaafee5794ada85c2753","timestamp":"2023-11-18T20:00:00.000Z","__v":0},{"data":
{"temperature":"20.97","humidity":"67"},"_id":"6564eaafee5794ada85c275c","timestamp":"2023-11-24T07:00:00.000Z","__v":0},{"data":
{"temperature":"18.71","humidity":"71"},"_id":"6564eaafee5794ada85c2755","timestamp":"2023-11-05T01:00:00.000Z","__v":0},{"data":
{"temperature":"20.26","humidity":"69"},"_id":"6564eaafee5794ada85c2758","timestamp":"2023-11-08T05:00:00.000Z","__v":0},{"data":
{"temperature":"22.82","humidity":"71"},"_id":"6564eaafee5794ada85c2757","timestamp":"2023-11-01T04:00:00.000Z","__v":0},{"data":
{"temperature":"19.45","humidity":"66"},"_id":"6564eaafee5794ada85c2759","timestamp":"2023-11-22T17:00:00.000Z","__v":0},{"data":
{"temperature":"23.15","humidity":"70"},"_id":"6564eaafee5794ada85c2752","timestamp":"2023-11-11T04:00:00.000Z","__v":0},{"data":
{"temperature":"19.87","humidity":"65"},"_id":"6564eaafee5794ada85c275e","timestamp":"2023-11-16T15:00:00.000Z","__v":0},{"data":
{"temperature":"19.12","humidity":"66"},"_id":"6564eaafee5794ada85c2761","timestamp":"2023-11-15T23:00:00.000Z","__v":0},{"data":
{"temperature":"20.53","humidity":"66"},"_id":"6564eaafee5794ada85c2760","timestamp":"2023-11-08T03:00:00.000Z","__v":0},{"data":
{"temperature":"18.59","humidity":"65"},"_id":"6564eaafee5794ada85c2762","timestamp":"2023-11-20T12:00:00.000Z","__v":0},{"data":
{"temperature":"21.39","humidity":"67"},"_id":"6564eaafee5794ada85c275f","timestamp":"2023-11-11T15:00:00.000Z","__v":0},{"data":
{"temperature":"19.21","humidity":"66"},"_id":"6564eaafee5794ada85c2763","timestamp":"2023-11-10T12:00:00.000Z","__v":0},{"data":
{"temperature":"24.77","humidity":"65"},"_id":"6564eaafee5794ada85c2764","timestamp":"2023-11-22T13:00:00.000Z","__v":0},{"data":
{"temperature":"24.89","humidity":"71"},"_id":"6564eaafee5794ada85c2765","timestamp":"2023-11-25T15:00:00.000Z","__v":0},{"data":
{"temperature":"24.43","humidity":"66"},"_id":"6564eaafee5794ada85c276a","timestamp":"2023-11-04T20:00:00.000Z","__v":0},{"data":
{"temperature":"22.91","humidity":"67"},"_id":"6564eaafee5794ada85c2768","timestamp":"2023-11-05T13:00:00.000Z","__v":0},{"data":
{"temperature":"23.8","humidity":"72"},"_id":"6564eaafee5794ada85c2767","timestamp":"2023-11-14T17:00:00.000Z","__v":0},{"data":
{"temperature":"21.75","humidity":"67"},"_id":"6564eaafee5794ada85c2766","timestamp":"2023-11-25T15:00:00.000Z","__v":0},{"data":
{"temperature":"22.85","humidity":"68"},"_id":"6564eaafee5794ada85c2769","timestamp":"2023-11-01T09:00:00.000Z","__v":0},{"data":
{"temperature":"19.0","humidity":"67"},"_id":"6564eaafee5794ada85c2774","timestamp":"2023-11-12T05:00:00.000Z","__v":0},{"data":
{"temperature":"21.74","humidity":"66"},"_id":"6564eaafee5794ada85c2772","timestamp":"2023-11-03T19:00:00.000Z","__v":0},{"data":
{"temperature":"24.31","humidity":"66"},"_id":"6564eaafee5794ada85c2773","timestamp":"2023-11-25T17:00:00.000Z","__v":0},{"data":
{"temperature":"24.04","humidity":"69"},"_id":"6564eaafee5794ada85c2775","timestamp":"2023-11-14T22:00:00.000Z","__v":0},{"data":
{"temperature":"20.82","humidity":"68"},"_id":"6564eaafee5794ada85c2776","timestamp":"2023-11-03T22:00:00.000Z","__v":0},{"data":
{"temperature":"20.09","humidity":"70"},"_id":"6564eaafee5794ada85c2771","timestamp":"2023-11-10T23:00:00.000Z","__v":0},{"data":
{"temperature":"24.78","humidity":"68"},"_id":"6564eaafee5794ada85c276f","timestamp":"2023-11-24T17:00:00.000Z","__v":0},{"data":
{"temperature":"21.12","humidity":"69"},"_id":"6564eaafee5794ada85c2770","timestamp":"2023-11-11T12:00:00.000Z","__v":0},{"data":
{"temperature":"18.58","humidity":"65"},"_id":"6564eaafee5794ada85c276b","timestamp":"2023-11-23T04:00:00.000Z","__v":0},{"data":
{"temperature":"23.14","humidity":"66"},"_id":"6564eaafee5794ada85c276e","timestamp":"2023-11-07T22:00:00.000Z","__v":0},{"data":
{"temperature":"19.03","humidity":"66"},"_id":"6564eaafee5794ada85c276c","timestamp":"2023-11-18T18:00:00.000Z","__v":0},{"data":
{"temperature":"21.59","humidity":"66"},"_id":"6564eaafee5794ada85c276d","timestamp":"2023-11-08T18:00:00.000Z","__v":0},{"data":
{"temperature":"20.52","humidity":"67"},"_id":"6564eaafee5794ada85c2783","timestamp":"2023-11-09T11:00:00.000Z","__v":0},{"data":
{"temperature":"20.55","humidity":"66"},"_id":"6564eaafee5794ada85c2784","timestamp":"2023-11-12T23:00:00.000Z","__v":0},{"data":
{"temperature":"18.81","humidity":"71"},"_id":"6564eaafee5794ada85c2785","timestamp":"2023-11-16T01:00:00.000Z","__v":0},{"data":
{"temperature":"21.85","humidity":"65"},"_id":"6564eaafee5794ada85c2781","timestamp":"2023-11-21T02:00:00.000Z","__v":0},{"data":
{"temperature":"21.73","humidity":"71"},"_id":"6564eaafee5794ada85c277e","timestamp":"2023-11-21T04:00:00.000Z","__v":0},{"data":
{"temperature":"23.88","humidity":"67"},"_id":"6564eaafee5794ada85c2777","timestamp":"2023-11-03T04:00:00.000Z","__v":0},{"data":
{"temperature":"18.26","humidity":"65"},"_id":"6564eaafee5794ada85c2787","timestamp":"2023-11-11T14:00:00.000Z","__v":0},{"data":
{"temperature":"22.13","humidity":"70"},"_id":"6564eaafee5794ada85c278a","timestamp":"2023-11-24T03:00:00.000Z","__v":0},{"data":
{"temperature":"23.15","humidity":"65"},"_id":"6564eaafee5794ada85c2789","timestamp":"2023-11-13T19:00:00.000Z","__v":0},{"data":
```


Fig.21: JSON format from the API

The dataset which is stored in the database is being utilized as an API to our dashboard through the server hosted on the render. The dataset is structured into JSON format which can be easily mapped to our dashboard for displaying temperature, humidity, date and time. The frontend dashboard's structured JSON data serves as its basis, making it simple to interpret and dynamically display current or past sensor values. This data may be processed and shown by the frontend in an effective manner, which improves the responsiveness of the user interface and guarantees that users dealing with the environmental data on the dashboard have an easy time for navigating the data.

3.5 Key Challenges

- **Network Stability:** maintaining a stable network connection between the Arduino and the server is critical for uninterrupted data transmission.

- **Data Access Control:** establishing proper access control measures ensures that only authorized users can access and manage the stored data.

- **Security:** putting in place suitable authentication and authorization procedures and making sure that client, server, and database communication is safe.

- **User Interface Consistency:** it took constant attention to keep the dashboard's layout consistent and easy to use across all of its parts, especially when new features were added.

- **Responsive Design:** it might be difficult to create a responsive design that guarantees a smooth user experience on a range of devices and screen sizes.

CHAPTER 4: TESTING

4.1 Testing Strategy

- **API Testing for Endpoints**

  Tools: Postman, Thunder Client

  Strategy: Test the integration of your API endpoints with the server and database.

- **Database Testing**

  Tools: MongoDB memory server for Node.js

  Strategy: Ensure that our database schema is correct and that data is inserted, updated and deleted appropriately.

- **Arduino-DHT11 Sensor Testing**

  Tools: Custom testing scripts, Arduino Simulator

  Strategy: Develop testing scripts for the Arduino code to simulate sensor readings.

- **Mobile Responsiveness Testing**

  Tools: Responsive design testing tools

  Strategy: i) Test the responsiveness of our React app on different screen sizes.

  ii) Ensure that the user interface adapts well to mobile devices and tablets.

4.2 Test Cases and Outcomes

- **Test Case 1:** POST Request to Add Sensor Data

  Outcome: Verified that the API endpoint returns sensor data in the expected format.

- **Test Case 2:** Insert Data into the Database

  Outcome: Verified that data is successfully inserted into the test database.

- **Test Case 3:** Simulate Valid Sensor Reading

  Outcome: Ensure that the Arduino code processes valid sensor readings correctly.

- **Test Case 4:** Verify Responsive Design on Different Devices

  Outcome: Verified that screen readers interpret and announce content correctly on mobile devices.

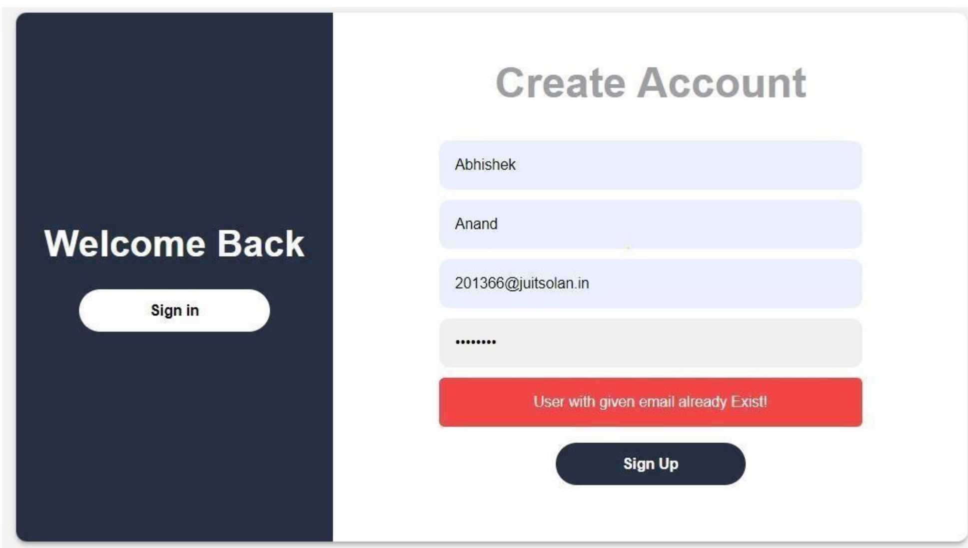

Fig.22: Invalid signup as user already existed

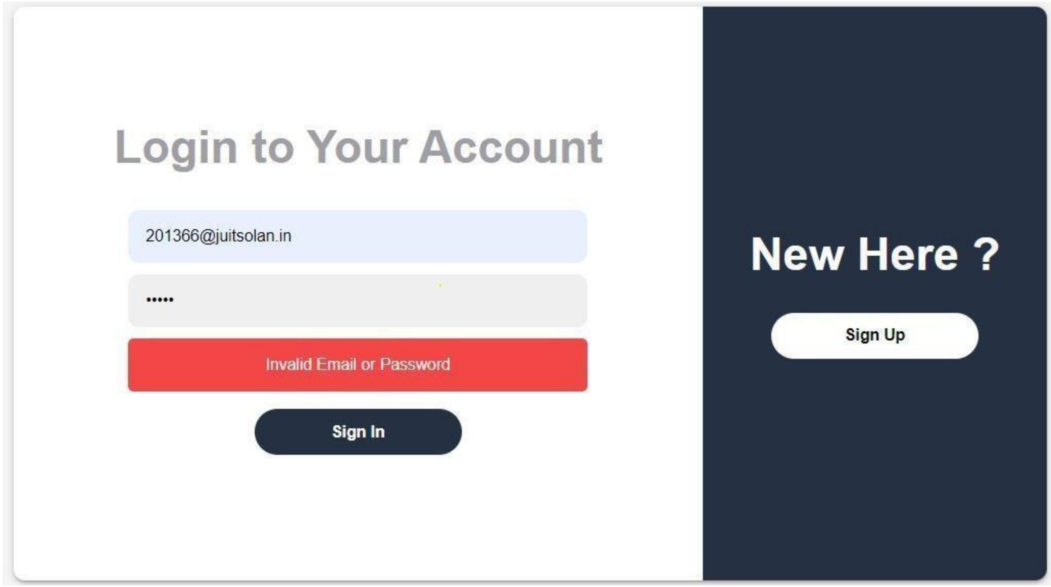

Fig.23: Invalid login as password does not match

When a user attempts to create a report, the web software exams to peer if the email deal with entered already exists within the database. The framework prevents the advent of a duplicate account with a similar email in the not going event that the email is now associated with a modern record. This ensures facts integrity and prevents users from inadvertently or deliberately creating more than one fact with identical email addresses. The customer presents their email deal with and secret word at some point of the login cycle. By comparing the entered email address with the stored records in the information set, the net software verifies the consumer. In the not going event that the supplied email is actual, the device verifies that the secret phrase entered suits the hidden mystery word related to that email. In the event if a confusion happens and shows wrong mystery word, the purchaser is informed that the login certifications are void. This protection degree protects towards unauthorised get right of entry to considering that the email and mystery word are efficaciously coupled inside the statistics set, and the framework may additionally get entry offer.

CHAPTER 5: RESULTS AND EVALUATION

5.1 Results

Fig.24: Dashboard

The development process of our temperature and humidity monitoring website involved several key steps. The Dashboard is divided into two components one for the displaying the information all together and a sidebar for separate view of them. First, we established a connection to MongoDB Atlas, creating a cluster to store our data and obtaining the connection string for integration. Using a MongoDB driver in our chosen programming language, we set up a seamless connection to our database. For the API integration, we selected a suitable source for temperature and humidity data.

Fig.25: Graph Visualization of dataset

By incorporating these graph visualizations, our website not only delivers real-time data but also serves as a valuable tool for users seeking a historical perspective on temperature and humidity changes. The user can get the daily temperature and humidity data along with the day and date. Apart from this there also chart representation for this we are utilizing the react apex chart which came in npm package. On the sidebar it can be seen that the user can get the separate option for different functionality such as dataset, login and logout. The data from the server render is then used for displaying the required information like temperature and humidity charts for a particular period.

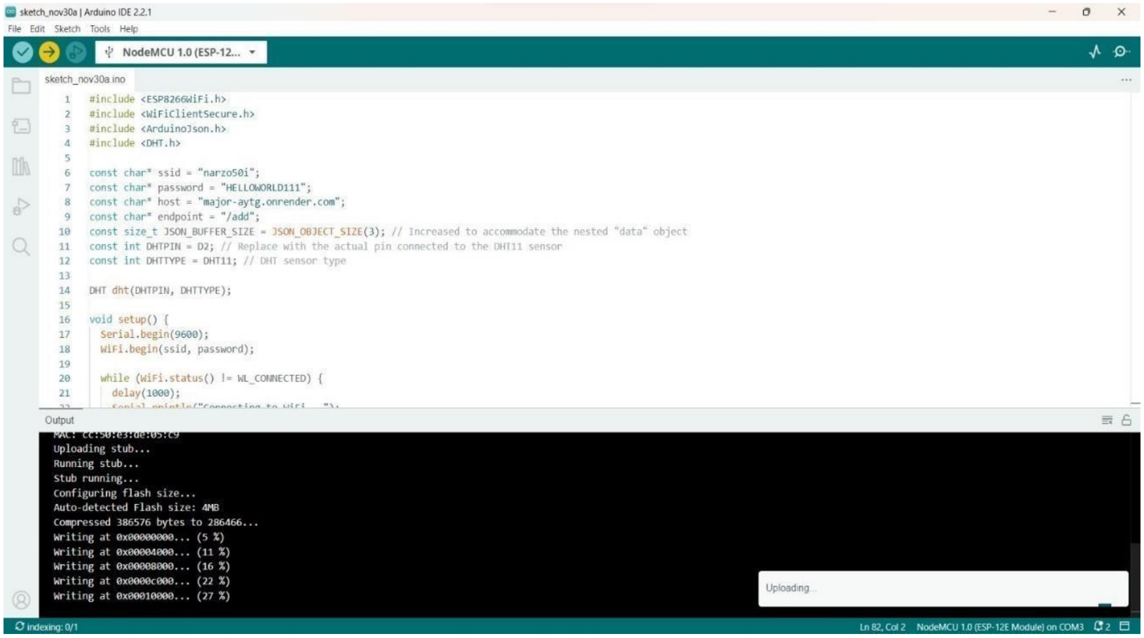

Fig.26: Arduino IDE Sending Data

First, we installed the Arduino IDE for writing the code in our microcontroller. The code consisted of default libraries for NodeMCU, DHT11-sensor and Wi-Fi for connecting the microcontroller to our devices. For storing the Wi-Fi, id and password we defined the variables for it. Along with it we have also defined the path to our database which is MongoDB for the successful storage of temperature and humidity which we are getting from our DHT11 sensor.

For the data to get automatically updated to the server render, whenever our sensor is connected, we are using the host variable in which we are assigning the path and port number. The consist of two functions, first one includes the Wi-Fi setup which is to ensure the connection between the microcontroller and our device and if the connection is establish then a print statement will execute which is used for telling the user that the Wi-Fi connection is established. The second function named as loop is used for continuously getting the data from the sensor in float format, which is then converted to JSON format and send to the server render. Apart from this we have added the delay of ten seconds which means the microcontroller reads the data from the sensor at every ten seconds and give it to the server.

CHAPTER 6: CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

The framework's capacity to gather, store, transfer, show and dissect temperature and stickiness information gives an important instrument to a large number of uses including weather conditions gauging, natural checking, and horticultural administration. By giving ongoing bits of knowledge into natural circumstances, the framework can uphold informed navigation and add to worked on ecological results.

The carried-out framework shows the effective combination of equipment, programming, and web innovations to make a complete checking arrangement. The framework really uses Arduino microcontrollers to gather information from sensors, a MongoDB data set to store the gathered information, a server-side Programming interface to deal with information transfers and recovery, and an easy-to-use site to show the information in an educational and open way. The fuse of a login and information exchange framework guarantees that client information is secure and safeguarded while empowering customized information access and the executives.

The framework's capacities can be additionally improved through future headways for example, constant information handling, prescient demonstrating, portable application mix, high level information investigation, sensor network development, device stage joining, and AI coordination. These progressions will empower the framework to give considerably more profound bits of knowledge into ecological circumstances, support more proactive direction, and grow its materialness across different areas.

All in all, the temperature and dampness checking framework addresses an important commitment to ecological observing and information examination. Its capacity to give continuous bits of knowledge into ecological circumstances, combined with its true capacity for additional headways, makes it an incredible asset for supporting informed direction and working on natural results.

6.2 Future Scope

- **Sensor Network Expansion:** Expand the sensor network to cover a wider geographical area, enabling more comprehensive environmental monitoring and data collection.

- **Multi-Sensor Integration:** To provide more thorough environmental insights, expand the system to incorporate data from several types of sensors, such as wind speed, air pressure and soil moisture.

REFERENCES

[1] W. M. Ridwan et al., “Rainfall forecasting model using machine learning methods case study Terengganu, Malaysia,” Ain Shams Eng. J, (2020).

[2] Mellit, A. M. Pavan, and M. Benghanem, “Least squares support vector machine for short-term prediction of meteorological time series,” Theor. Appl. Climatol., vol. 111, pp. 297–307”, (2013).

[3] M. Ehteram et al., “Performance improvement for infiltration rate prediction using hybridized adaptive neuro-fuzzy inference system (ANFIS) with optimization algorithms,” Ain Shams Eng. J, (2020).

[4] M. Adnan et al., “Prediction of relative humidity in a high elevated basin of western Karakoram by using different machine learning models,” (2021).

[5] M. Ehteram et al., “Design of a hybrid ANN multi-objective whale algorithm for suspended sediment load prediction,” Environ. Sci. Pollut. Res, (2020).

[6] M. Sapitang, W. Ridwan, K. F. Kushiar, A. N. Ahmed, and A. El-Shafie, “Machine learning application in reservoir water level forecasting for sustainable hydropower generation strategy,” Sustainability vol. 12, (2020).

[7] P. Körner, R. Kronenberg, S. Genzel, and C. Bernhofer, “Introducing gradient boosting as a universal gap filling tool for meteorological time series,” vol. 27, (2018).

[8] J. Cai, K. Xu, Y. Zhu, F. Hu, and L. Li, “Prediction and analysis of net ecosystem carbon exchange based on gradient boosting regression and random forest,” Appl. Energy,vol. 262, (2020).

[9] J. Fan et al., “Evaluation of SVM, ELM and four tree-based ensemble models for predicting daily reference evapotranspiration using limited meteorological data in different climates of China,” Agricultural Forest Meteorology, vol. 263, (2018).

[10] S. M. Karimi, O. Kisi, M. Porrajabali, F. Rouhani-Nia, and J. Shiri, “Evaluation of the support vector machine, random forest and geostatistical methodologies for predictinglong-term air temperature,” ISH J. hydraulic. Eng., vol. 26, (2020).

[11] A. Sharaf and S. R. Roy, “Comparative Analysis of Temperature Prediction Using Regression Methods and Back Propagation Neural Network,” Proceedings of the 2nd International Conference on Trends in Electronics and Informatics, (2018).

[12] E. Sreehari, “Prediction of Climate Variable using Multiple Linear Regression,” 2018 4th International Conference on Computing Communication and Automation,ICCCA (2018).

[13] I. Park, H. S. Kim, J. Lee, J. H. Kim, C. H. Song, and H. K. Kim, “Temperature prediction using the missing data refinement model based on a long short-term memory neural network,” Atmosphere (Basel)., vol. 10, no. 11, pp. 1-16, (2019).

[14] S. P. Menon, “Prediction of Temperature using Linear Regression,” International Conference on Electrical, Electronics, Communication Computer Technologies and Optimization Techniques, ICEECCOT, (2017).

[15] T. Khan, M. Rabbani, S. M. Tanvir Siddique, and A. Majumder, “An Innovative Data Mining Approach for Determine Earthquake Probability Based on Linear Regression Algorithm,” Proc. 2019 3rd IEEE Int. Conf. Electrical Computer Communication Technol. ICECCT (2019).

[16] S. S. Patil and S. A. Thorat, “Early detection of grapes diseases using machine learning and IoT,” in Proceedings - 2016 2nd International Conference on Cognitive Computing and Information Processing (2016).

[17] P. S. Pandey, “Machine Learning and IoT for prediction and detection of stress,” in Proceedings of the 2017 17th International Conference on Computational Science and ItsApplications (2017).

[18] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guiana, “Deep learning for IoT bigdata and streaming analytics: A survey,” IEEE Communications Surveys and Tutorials,vol. 20, (2018).

[19] N. Baracaldo, B. Chen, H. Ludwig, A. Safavi, and R. Zhang, “Detecting poisoningattacks on machine learning in IoT environments,” in Proceedings - 2018 IEEE International Congress on Internet of Things, ICIOT 2018 – Part of the 2018 IEEEWorld Congress on Services, (2018).

[20] O. Giustolisi, A. Doglioni, D. A. Savic, and B. W. Web, “A multi-model approach toanalysis of environmental phenomena,” Environ (2016).

[21] D. Caissie, “The thermal regime of rivers: A review,” Freshwater Biology, vol. 51,no. 8. John Wiley & Sons, Ltd, (2015).

[22] B. Gardner, P. J. Sullivan, and A. J. Lembo Jr, “Predicting stream temperatures: geostatistical model comparison using alternative distance metrics,” Can. J. Fish. Aqua.Sci., vol. 60, no. 3, pp. 344-351, Mar. (2003).

[23] S. Wang, G. Morishima, R. Sharma, and L. Gilbertson, “The Use of Generalized Additive Models for Forecasting the Abundance of Queets River Coho Salmon,” NorthAm. J. Fish. Manag., vol. 29, no. 2, pp. 423-433, Apr. (2009).

[24] C. H. Dagli, “Machine Learning Methods for Data Assimilation,” Intel. Eng. Syst. through Artificial Neural Networks, vol. 20, pp. 105-112, (2010).

[25] P. Romeu, F. Zamora-Martínez, P. Botella-Rocamora, and J. Pardo, “Time-Series Forecasting of Indoor Temperature Using Pre-Trained Deep Neural Networks," Artificial Neural Networks and Machine Learning – ICANN (2013).

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: ……………………….

Type of Document (Tick): **PhD Thesis** **M.Tech Dissertation/ Report** **B.Tech Project Report** **Paper**

Name: ___________________________ __**Department:** _________________ **Enrolment No** _________

Contact No. ______________________________**E-mail.** ______________________________________

Name of the Supervisor: ____

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): ________________________

____

____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages =
- Total No. of Preliminary pages =
- Total No. of pages accommodate bibliography/references =

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at ……………….(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	**Excluded**	**Similarity Index (%)**	**Generated Plagiarism Report Details** (Title, Abstract & Chapters)	
	• All Preliminary Pages • Bibliography/Images/Quotes • 14 Words String		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name & Signature **Librarian**

………

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com

Aman_Report_Final

ORIGINALITY REPORT

6%	5%	5%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.researchgate.net Internet Source	1%
2	www.nature.com Internet Source	1%
3	link.springer.com Internet Source	<1%
4	A. Vamseekrishna, R. Nishitha, T. Anil Kumar, K. Hanuman, Ch. G. Supriya. "Chapter 30 Prediction of Temperature and Humidity Using IoT and Machine Learning Algorithm", Springer Science and Business Media LLC, 2021 Publication	<1%
5	D.B. Jani, Manish Mishra, P.K. Sahoo. "Application of artificial neural network for predicting performance of solid desiccant cooling systems – A review", Renewable and Sustainable Energy Reviews, 2017 Publication	<1%
6	www.ijesd.org Internet Source	<1%

7	Nevin Celik. "Statistical regression and artificial neural network analyses of impinging jet experiments", Heat and Mass Transfer, 03/2009 Publication	<1%
8	ijeecs.iaescore.com Internet Source	<1%
9	Anderson Tenorio Sergio, Teresa B. Ludermir. "Deep Learning for Wind Speed Forecasting in Northeastern Region of Brazil", 2015 Brazilian Conference on Intelligent Systems (BRACIS), 2015 Publication	<1%
10	Noémi Barabás, Pierre Goovaerts, Peter Adriaens. "Geostatistical Assessment and Validation of Uncertainty for Three-Dimensional Dioxin Data from Sediments in an Estuarine River", Environmental Science & Technology, 2001 Publication	<1%
11	fabacademy.org Internet Source	<1%
12	collections.mun.ca Internet Source	<1%
13	www.irisa.fr Internet Source	<1%

14	www.readkong.com Internet Source	<1%
15	Yuying Wei, Adrian Wing-Keung Law, Chun Yang. "Probabilistic Optimal Interpolation for Data Assimilation between Machine Learning Model Predictions and Real Time Observations", Journal of Computational Science, 2023 Publication	<1%
16	dergipark.org.tr Internet Source	<1%
17	Beth Gardner, Patrick J Sullivan, Arthur J Lembo, Jr.. "Predicting stream temperatures: geostatistical model comparison using alternative distance metrics", Canadian Journal of Fisheries and Aquatic Sciences, 2003 Publication	<1%
18	mdpi-res.com Internet Source	<1%
19	www.hindawi.com Internet Source	<1%
20	Aakanksha Sharaff, Samuel Robin Roy. "Comparative Analysis of Temperature Prediction Using Regression Methods and Back Propagation Neural Network", 2018 2nd	<1%

International Conference on Trends in Electronics and Informatics (ICOEI), 2018
Publication

21 Farrukh Hafeez, Usman Ullah Sheikh, Attaullah Khidrani, Muhammad Akram Bhayo et al. "Distant temperature and humidity monitoring: prediction and measurement", Indonesian Journal of Electrical Engineering and Computer Science, 2021 <1%
Publication

22 K. Suleyman Yigit, H. Metin Ertunc. "Prediction of the air temperature and humidity at the outlet of a cooling coil using neural networks", International Communications in Heat and Mass Transfer, 2006 <1%
Publication

23 Tadej Krivec, Juš Kocijan, Matija Perne, Boštjan Grašic, Marija Zlata Božnar, Primož Mlakar. "Data-driven method for the improving forecasts of local weather dynamics", Engineering Applications of Artificial Intelligence, 2021 <1%
Publication

24 Thaharim Khan, Masud Rabbani, Shah Md Tanvir Siddiquee, Anup Majumder. "An Innovative Data Mining Approach for Determine Earthquake Probability Based on Linear Regression Algorithm", 2019 IEEE <1%

International Conference on Electrical, Computer and Communication Technologies (ICECCT), 2019
Publication

25	dspace.dtu.ac.in:8080 Internet Source	<1%
26	K. Bhamidipati, G. Anuradha, B. Swaminathan, S. Muppidi. "Smart IoT Application in Soil Moisture and Heat Level Prediction Using Sine Cosine-Horse Herd Optimized Deep Learning", International Journal on Artificial Intelligence Tools, 2023 Publication	<1%
27	Marwah Sattar Hanoon, Ali Najah Ahmed, Nur'atiah Zaini, Arif Razzaq et al. "Developing machine learning algorithms for meteorological temperature and humidity forecasting at Terengganu state in Malaysia", Scientific Reports, 2021 Publication	<1%
28	Minghui Zhang, Yatong Zhou, Yabo Liu. "Deep reservoir calculation model and its application in the field of temperature and humidity prediction", Applied Intelligence, 2022 Publication	<1%
29	www.ncbi.nlm.nih.gov Internet Source	<1%

30 Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel. "Backpropagation Applied to Handwritten Zip Code Recognition", Neural Computation, 1989 <1%

Publication

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		