# Motilal Nehru National Institute of Technology Allahabad

**Department of Electronics and Communication Engineering (ECED)**

Project report on

# Long-Range Connectivity and Machine Learning: A Comprehensive Study on LoRa in Environmental Monitoring


By

Team Members

1). Sarthak Dwivedi (20215102)

2). Sattwik Ranjan (20215030)

Under the Guidance of

**Mr. Asim Mukherjee**

Associate Professor, ECED, MNNIT Allahabad

# Undertaking

We declare that the work presented in the hardware project titled, **"Long-Range Connectivity and Machine Learning: A Comprehensive Study on LoRa in Environmental Monitoring"**, submitted to the department of Electronics and Communication Engineering, Motilal Nehru National Institute of Technology, Allahabad is our own work.

**Sarthak Dwivedi – (20215102)**

**Sattwik Ranjan – (20215030)**

# Motilal Nehru National Institute of Technology Allahabad

# CERTIFICATE

In embarking upon this project, it would be remiss not to express our sincere gratitude to the guiding forces that have shaped its course. At the outset, we humbly thank the divine forces and our parents, whose unwavering encouragement and blessings have been the cornerstone of our endeavors. Their support has been a guiding light, propelling us towards the successful completion of this project. In particular, we owe a debt of gratitude to Mr. Asim Mukherjee Sir, whose generosity in providing us with the opportunity to undertake this project in our third year has been instrumental. His discerning guidance, selection of an excellent project topic, and constant inspiration, coupled with timely assistance, have been invaluable contributors to our growth and learning. Their accurate and swift deliveries played a crucial role in ensuring the seamless execution of our project. Lastly, we express our deepest regards to the Electronics and Communication Department of our Institute. Their recognition of our efforts and continuous guidance have been instrumental in shaping the direction of our project. This report is a testament to the collective efforts and unwavering support of all those who have contributed to our growth and success in the field of electronics engineering.

**Date:** **Mr Asim Mukherjee**

**Place:** **Associate Professor, Department of ECED**

# PREFACE

In the realm of modern electronics engineering, the exploration of cutting-edge technologies and innovative methodologies is paramount. This project unfolds against the backdrop of LoRa communication, a revolutionary wireless technology that forms the foundation of our exploration. The core objective involved the creation of a robust environmental monitoring system employing two Arduino boards, each equipped with LoRa shields operating at a frequency of 915 MHz. The journey commenced with the development of a broadcasting device, seamlessly connecting to a laptop through the Arduino IDE, incorporating a DHT11 sensor for precise temperature and humidity readings. The broadcasting device, featuring a LoRa shield and an antenna, transmitted data to a receiver device, mirroring its counterpart. The acquired temperature and humidity data, enriched with timestamps, were meticulously transformed into a structured CSV file, paving the way for in-depth analysis. Delving into the realm of predictive analytics, this project embarked on the construction of machine learning models. The initial foray involved the utilization of linear regression, laying the foundation for subsequent advancements. A neural network model, characterized by two hidden layers employing ReLU activation functions, brought forth enhanced complexity and accuracy. Further strides were taken with the incorporation of a random forest classifier, culminating in a remarkable 89% accuracy, as validated through a comprehensive confusion matrix. The project's narrative takes an intriguing turn with the introduction of adversarial attacks—specifically, the FGSM and Carlini Wagner attack methods. These attacks, often viewed as adversaries, served as catalysts for refinement, elevating the robustness of our models. The iterative process of model enhancement through adversarial attacks underscored our commitment to relentless improvement and adaptability. As the pages unfold, each facet of this project contributes to a mosaic of discovery, innovation, and practical application. The journey traversed from hardware integration to predictive modeling and adversarial resilience is a testament to the resilience and collaborative spirit of the project team. This report encapsulates the essence of our exploration—a fusion of technology, creativity, and the relentless pursuit of excellence.

# ABSTRACT

This project ventures into the realm of LoRa communication to develop a robust environmental monitoring system. Utilizing two Arduino boards equipped with LoRa shields, operating at a frequency of 915 MHz, the project unfolds with the creation of a broadcasting device. Paired with a DHT11 sensor for temperature and humidity readings, this device transmits data to a receiving counterpart, generating a comprehensive dataset with timestamps. The journey extends into predictive analytics, incorporating linear regression, a neural network model with ReLU activation functions, and a random forest classifier achieving an 89% accuracy. Notably, the introduction of adversarial attacks, including FGSM and Carlini Wagner, refines model robustness. The project culminates in a synthesis of hardware integration, predictive modeling, and adversarial resilience, underscoring the project team's commitment to technological innovation.

# TABLE OF CONTENTS

# LIST OF FIGURES

# Chapter 1

# INTRODUCTION TO THE PROJECT

## 1.1. OBJECTIVE

### 1.1.1 Implement LoRa Communication:

Establish a reliable communication system using LoRa technology between two Arduino-based devices for efficient data transmission.

### 1.1.2 Environmental Monitoring:

Develop a comprehensive environmental monitoring system capable of capturing real-time temperature and humidity data.

### 1.1.3 Hardware Integration:

Integrate DHT11 sensors with Arduino boards and LoRa shields to create broadcasting and receiving devices for data collection.

### 1.1.4 Data Processing and Timestamping:

Implement a data processing mechanism that transforms raw sensor data into a structured format, incorporating timestamps for temporal analysis.

### 1.1.5 Predictive Modelling:

Construct predictive machine learning models, including Linear Regression, Neural Network, and Random Forest, to forecast temperature at a later time based on environmental conditions.

### 1.1.6 Model Evaluation and Improvement:

Evaluate model performance, employing metrics such as accuracy and confusion matrix, and iteratively improve models using adversarial attacks (e.g., FGSM, Carlini Wagner) to enhance robustness.

### 1.1.7 Create a User-Friendly Interface:

Design a user-friendly interface for practical implementation, allowing easy interaction and visualization of the obtained temperature and humidity predictions.

### 1.1.8 Explore Real-world Applications:

Investigate and showcase the real-world applicability of the project, with potential use cases in agriculture, smart homes, and industrial monitoring.

## 1.2. WHAT IS LORA

LoRa, short for Long Range, is a wireless communication technology designed to provide long-range, low-power connectivity for the Internet of Things (IoT) devices. Developed to address the unique challenges of IoT applications, LoRa operates on unlicensed frequency bands, allowing for efficient, cost-effective, and scalable communication in diverse environments.

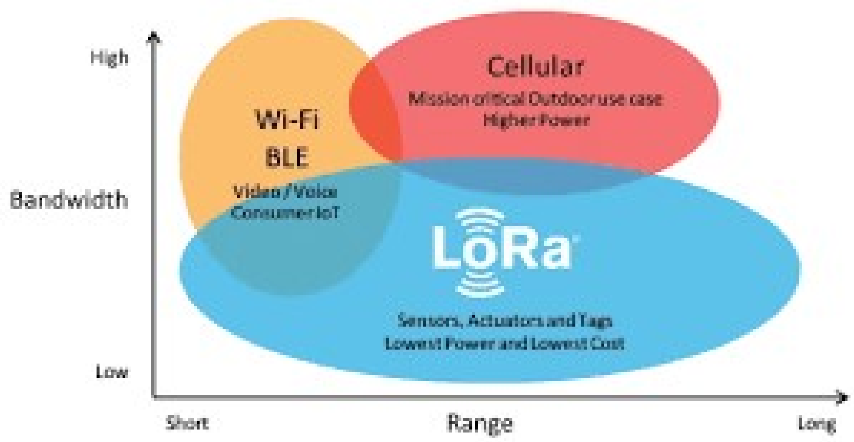



*Figure 1 - LoRa*

### Key Characteristics of LoRa:

#### i. Long Range:

LoRa technology enables communication over extended distances, making it well-suited for applications that require connectivity over large geographic areas

.

ii. **Low Power Consumption:**

LoRa devices are designed to operate on minimal power, extending battery life for connected devices and enabling long-term, energy-efficient deployments.

iii. **Low Data Rates:**

While not designed for high-bandwidth applications, LoRa excels in scenarios where low data rates are sufficient, making it ideal for applications like environmental monitoring.

iv. **Scalability:**

LoRa networks can scale efficiently to accommodate a large number of devices, offering flexibility for both small-scale and large-scale IoT deployments.

**Applications of LoRa:**

i. **Smart Agriculture:**

LoRa is utilized in agriculture for soil monitoring, weather stations, and crop health tracking, enabling efficient and remote farm management.

ii. **Smart Cities:**

LoRa networks are employed for smart city applications, including waste management, parking solutions, and environmental monitoring.

### iii. Asset Tracking:

LoRa is employed for tracking and managing assets, providing real-time location information for valuable items or fleet management.

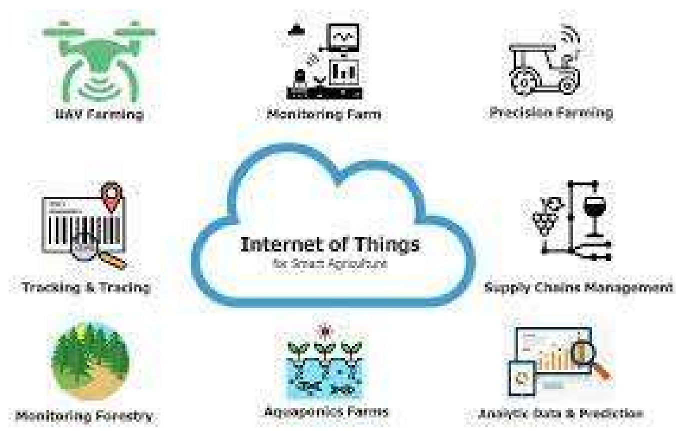



*Figure 2 – IOT in Smart Agriculture*

# Chapter 2

# HARDWARE SETUP

## 2.1. INTRODUCTION

In this chapter, we delve into the hardware setup of our environmental monitoring project, outlining the components that form the backbone of our system. A comprehensive understanding of the hardware is essential for the successful implementation of the project. Let's begin by introducing the key components and their respective roles in our setup.

## 2.2. COMPONENTS USED

### i. Arduino Boards

The project utilizes two Arduino boards as the central processing units for the broadcasting and receiving devices. Arduino serves as the brain of the system, orchestrating data collection and processing.

### ii. LoRa Shields (Frequency: 915 MHz)

Two LoRa shields operating at a frequency of 915 MHz are integrated with the Arduino boards. These shields facilitate long-range communication between the broadcasting and receiving devices.

### iii. DHT11 Sensor

The DHT11 sensor is employed for real-time measurement of temperature and humidity. Connected to the broadcasting device, this sensor provides crucial environmental data for analysis.

### iv. Antenna

An antenna is attached to both the LoRa shields, enhancing the range and efficiency of communication between the broadcasting and receiving devices.

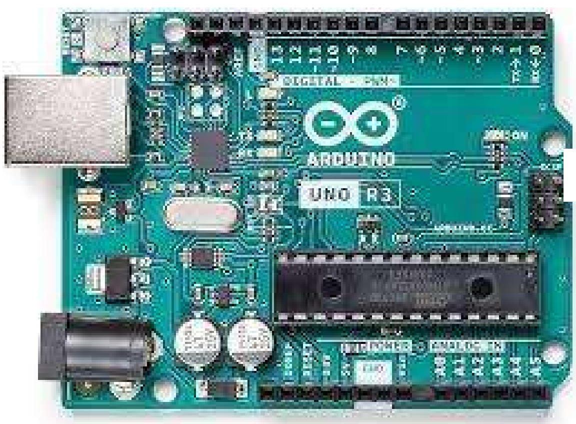



*Figure 3 – Arduino Board*

*Figure 4 – LoRa Shield*

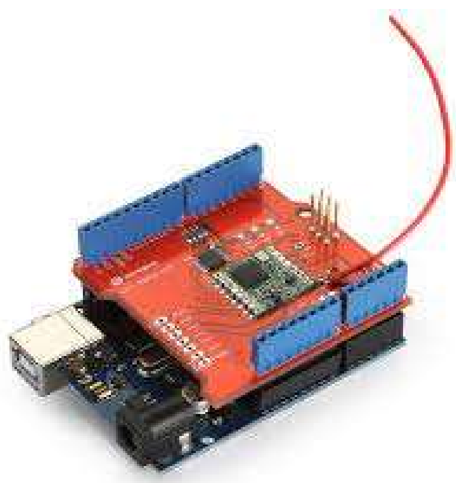


*Figure 5 – Hardware Setup*

## 2.3. Hardware Connection

### i. Arduino and LoRa Shield Integration

Connect the LoRa shield to the Arduino board, ensuring a secure and stable connection. Pay attention to pin configurations and compatibility to avoid communication issues.

### ii. DHT11 Sensor Connection

Wire the DHT11 sensor to the broadcasting Arduino board, establishing the necessary connections for obtaining temperature and humidity readings.

### iii. Antenna Attachment

Connect the antenna to the designated port on both LoRa shields. Properly secure the antenna to optimize signal transmission.

### iv. Power Supply

Ensure adequate power supply to both Arduino boards and associated components. Consider the power requirements of the LoRa shields and sensors for stable operation.

## 2.4. CONFIGURATION SETTINGS

### i. Arduino IDE Setup

Configure the Arduino Integrated Development Environment (IDE) for both broadcasting and receiving devices. Ensure the correct selection of board type, port, and programming language.

### ii. LoRa Communication Parameters

Set up the LoRa communication parameters, including frequency and transmission settings, to establish a reliable and efficient communication link.

# Chapter 3

# DATA COLLECTION

## 3.1. INTRODUCTION

This chapter delves into the crucial aspect of data collection in our environmental monitoring project. The reliability and accuracy of the data acquired directly impact the effectiveness of subsequent analyses. In this section, we discuss the methods and procedures employed to collect temperature and humidity data from the broadcasting device and its transmission to the receiving device.

## 3.2. Data Collection Process

### i. Sensor Calibration

Before initiating data collection, ensure the calibration of the DHT11 sensor to guarantee accurate temperature and humidity readings. Discuss any specific calibration procedures undertaken for the project.

### ii. Sampling Rate and Frequency

Define the sampling rate at which the DHT11 sensor collects data. Discuss considerations for choosing an appropriate sampling rate and the frequency of data transmission between the broadcasting and receiving devices.

### iii. Timestamping

Implement timestamping for each data point to establish a temporal context. Describe the method used for timestamping, whether it's done locally on the Arduino board or during data processing.

## 3.3. DATA TRANSMISSION

### i. LoRa Communication Protocol

LoRa, which stands for Long Range, is a communication protocol and technology designed for long-range, low-power communication between devices in the context of the Internet of Things (IoT). LoRa operates in the sub-gigahertz frequency bands, such as 868 MHz in Europe, 915 MHz in North America, and 433 MHz in some other regions.

### ii. CSV File Creation

The process of converting raw sensor data into a structured CSV file involves collecting data from sensors, preprocessing for cleaning and filtering, associating each data point with a timestamp, and organizing the data into rows and columns with headers. The CSV file is created with a consistent timestamp format (e.g., ISO 8601), and considerations are made for time zones if applicable. The file is named with a meaningful convention, and documentation includes details on column meanings and data collection. Timestamps are crucial for temporal analysis, allowing chronological tracking of sensor readings over time for diverse applications in IoT, agriculture, or industry.

## 3.4. DATA VALIDATION

### i. Data Quality Checks

To validate the quality of collected data, several procedures are implemented, including range checks, outlier detection, and other methods. Range checks involve verifying that data values fall within expected and acceptable ranges for each sensor parameter, ensuring data consistency and relevance. Outlier detection techniques are employed to identify and address anomalies or extreme values that may skew the dataset. Statistical methods, such as mean and standard deviation analysis, can be used to detect outliers. Additionally, data validation may include cross-referencing with predefined patterns or criteria to identify inconsistencies. Regular audits and comparison with known standards or ground truth data help maintain data accuracy. Implementing rigorous validation processes ensures the reliability and integrity of collected data, enhancing its utility for meaningful analysis and decision-making in various fields, including scientific research, IoT applications, and industrial processes

## ii. Data Logging

Data logging mechanisms are implemented to monitor the status of data collection and identify errors or irregularities. These mechanisms involve the systematic recording of key events and metrics throughout the data collection process. They enable the tracking of timestamps, sensor readings, and system status, providing a comprehensive log for analysis. In the event of errors or irregularities, the logs serve as a valuable tool for diagnosing issues, understanding the context of anomalies, and implementing corrective measures. This proactive approach to data monitoring enhances the overall reliability and quality of the collected data, ensuring a robust and accountable data collection process for various applications, from industrial processes to scientific research and beyond.

# Chapter 4

# PREDICTIVE MODELLING

## 4.1. INTRODUCTION

In this chapter, we delve into the realm of predictive modeling, a cornerstone of our environmental monitoring project. Predictive modeling involves the creation of algorithms that leverage historical data to make informed predictions about future outcomes. In our context, we employ machine learning techniques to forecast temperature at a later time based on various environmental conditions. This section provides a comprehensive overview of the models employed, the rationale behind their selection, and the methodology used for their implementation.

## 4.2. Machine Learning for Environmental Prediction

### i. Objective of Predictive Modelling

The primary objective of predictive modeling in this project is to harness the power of machine learning algorithms to anticipate future temperature values based on a comprehensive analysis of both current and historical data. By leveraging the inherent patterns and relationships within the dataset, our aim is to develop models that can provide accurate and reliable predictions of temperature trends over time. The predictive models serve as invaluable tools for foreseeing environmental changes, offering a proactive approach to managing and responding to fluctuations in temperature. This objective aligns with the broader goal of enhancing environmental monitoring systems, contributing to more informed decision-making in various applications.

ii. **Significance in Environmental Monitoring**

Predictive modeling holds profound significance in the realm of environmental monitoring, presenting a paradigm shift in how we interpret and utilize data. The application of these models extends across diverse sectors, including agriculture, smart homes, and industrial settings, revolutionizing the way we approach environmental sustainability and resource management.

## 4.3. Models Employed

### i. Linear Regression

Linear regression in machine learning is a supervised learning algorithm used for predicting a continuous output variable based on one or more input features. The model assumes a linear relationship between the input features and the output, represented by a straight line. The goal is to find the best-fitting line that minimizes the sum of squared differences between the predicted and actual values. The model is trained by adjusting parameters (slope and intercept) to optimize the fit. Once trained, it can make predictions on new data. Linear regression is widely employed for tasks such as forecasting, trend analysis, and understanding the relationships between variables, providing a simple yet effective tool for regression problems in machine learning.

### ii. Neural Network Model

Neural network models in machine learning are a class of algorithms inspired by the structure and functioning of the human brain. Comprising interconnected nodes organized in layers, neural networks can learn complex patterns and representations from data. Input data is fed into the input layer, processed through hidden layers using learned weights and activation functions, and produces an output in the output layer. Training involves adjusting these weights through backpropagation to minimize the difference between predicted and actual outputs. Neural networks are versatile and excel in tasks like image recognition, natural language processing, and complex pattern recognition, making them a powerful tool for solving a wide range of machine learning problems. In our model there were two hidden layers. The first layer had 64 nodes and the second hidden layer had 32 nodes. Both the layers used the ReLU activation function. The model was trained by dividing the data into training and testing. The Mean squared error (MSE) was calculated for them to calculate the accuracy of the models.

### iii. Random Forest Classifier

The random forest classifier is an ensemble learning method in machine learning that constructs a multitude of decision trees during training. Each tree in the forest independently classifies input data, and the final prediction is determined by a majority vote or averaging among the individual tree predictions. Randomness is introduced in the tree-building process through features and data sampling, enhancing the model's robustness and reducing overfitting. Random forests are effective for both classification and regression tasks, and they excel in handling large, high-dimensional datasets. Known for their accuracy and versatility, random forests are widely used in applications such as image classification, medical diagnosis, and financial forecasting.

## 4.4. Model Training and Evaluation

### i. Data Splitting

The process of splitting a dataset into training and testing sets is crucial for effective model training and evaluation in machine learning. Typically, a random portion of the dataset, often around 80%, is assigned to the training set, where the model learns patterns and relationships from the data. The remaining portion, usually 20%, forms the testing set, serving as unseen data for assessing the model's performance. This separation allows the evaluation of the model's generalization to new, unseen data, helping to identify potential overfitting or underfitting issues. By assessing performance on an independent test set, practitioners can gauge the model's effectiveness and make informed decisions regarding its deployment and optimization.

## ii. Training the Models

The training phase of our predictive models involves a systematic approach to harness the potential of historical temperature and humidity data. Initially, the training dataset is carefully prepared by strategically splitting it into training and testing sets, laying the foundation for effective model evaluation. As the models are initialized, parameters such as weights, biases, or hyper parameters are set to kick-start the learning process. To ensure uniform contributions from all features, a crucial step of feature scaling is performed, normalizing the input data. The training procedure unfolds iteratively, with the model processing the training dataset to make predictions, calculating prediction errors, and adjusting weights or hyper parameters to minimize these errors. Hyper parameter tuning is introduced for models with configurable parameters, involving a meticulous process of experimentation to identify the optimal combination of values. Cross-validation techniques, such as K-fold cross-validation, are employed to gauge the model's generalization across different subsets of the training data. Model evaluation becomes a recurring practice, with regular assessments on the testing set to monitor the model's performance on unseen data. Utilizing regression metrics like Mean Squared Error (MSE) and Mean Absolute Error (MAE), the accuracy of predictions is quantified. The process is characterized by iterative optimization, where model parameters are adjusted, further training is conducted, and hyper parameters are fine-tuned based on evaluation results. In addition to hyper parameter tuning, regularization techniques are implemented to prevent overfitting and enhance model robustness. Techniques like dropout in neural networks or adjusting tree depth in random forests contribute to preventing the model from memorizing the training data excessively. Throughout the training iterations, convergence is consistently checked to ensure that the model continues to improve its performance and does not get stuck in local minima. This meticulous training process ensures that each predictive model effectively captures underlying patterns in the dataset, resulting in accurate and reliable temperature predictions.

## 4.5. Adversarial Attacks

Adversarial attacks in the context of machine learning refer to the deliberate manipulation of input data to mislead or deceive a trained model. These attacks exploit the model's vulnerabilities, aiming to cause misclassifications or errors. Adversarial examples are carefully crafted inputs that may appear normal to humans but can lead the model to make incorrect predictions. Adversarial attacks can take various forms, such as adding subtle perturbations to images or altering input features. Understanding and defending against adversarial attacks is crucial for ensuring the robustness and reliability of machine learning models, especially in sensitive applications like security, finance, and healthcare. Researchers and practitioners develop defense mechanisms and robust models to mitigate the impact of adversarial attacks and enhance the overall security of machine learning systems.

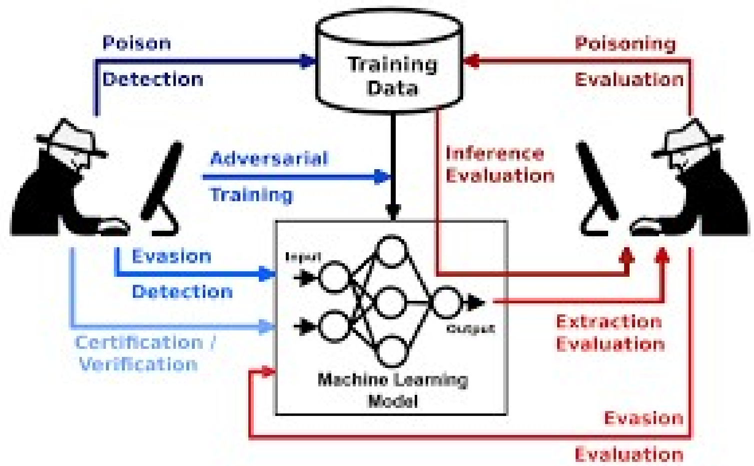



*Figure 6 – Machine Learning Models*

### 4.5.1. Attacks Implemented

#### i. Fast Gradient Sign Method (FGSM)

The Fast Gradient Sign Method (FGSM) is a type of adversarial attack commonly used in the context of machine learning, especially for image classification tasks. In an FGSM attack, an adversary crafts an adversarial example by perturbing the input data in the direction that maximizes the loss or error of the model. The perturbations are computed by taking the gradient of the loss with respect to the input data and then scaling the gradient by a small value (epsilon). The goal is to create a modified input that looks similar to the original to the human eye but leads the model to misclassify the example. FGSM attacks are known for their simplicity and efficiency, making them a popular choice for studying and understanding the vulnerabilities of machine learning models. Defenses against FGSM and other adversarial attacks often involve incorporating robustness measures during model training or using adversarial training techniques to expose models to such attacks during their learning process.

## ii. Carlini Wagner

The Carlini-Wagner attack, proposed by researchers Nicholas Carlini and David Wagner, is an advanced adversarial attack designed to generate subtle perturbations on input data, particularly in the context of neural networks. Unlike simpler attacks like FGSM, the Carlini-Wagner attack aims to find the minimal perturbation that leads to misclassification while simultaneously minimizing the perceptibility of the perturbation. It formulates the attack as an optimization problem, introducing a custom objective function that balances the trade-off between the size of the perturbation and the confidence of the misclassification.. Carlini-Wagner attacks highlight the challenges in developing robust machine learning models, as they demonstrate the existence of highly effective, imperceptible adversarial examples that can bypass traditional defense mechanisms. Researchers use such attacks to assess and improve the resilience of machine learning models against sophisticated adversarial manipulation.

# Chapter 5

# CONCLUSION AND FUTURE ENHANCEMENTS

## 5.1. PRACTICAL SIGNIFICANCE

In concluding this project, it's crucial to highlight the practical significance and real-world implications of our environmental monitoring system based on LoRa communication and predictive modeling. The project's outcomes extend beyond the technical intricacies, manifesting in tangible benefits for various sectors.

- **Environmental Monitoring and Precision Agriculture**

  Our system provides a robust solution for monitoring environmental conditions in agriculture. Farmers can leverage accurate temperature predictions for optimal crop management, ensuring timely planting and harvesting.

- **Smart Homes and Energy Efficiency**

  In smart homes, the system contributes to energy efficiency by enabling proactive control of heating and cooling systems based on forecasted temperature trends. This not only enhances comfort but also minimizes energy consumption.

- **Industrial Operations and Maintenance**

  Industries can deploy our environmental monitoring system to predict temperature changes, facilitating proactive maintenance and minimizing downtime. This predictive capability is pivotal for preventing equipment failures and optimizing operational efficiency.

## 5.2. FUTURE ENHANCEMENTS

While the current implementation marks a significant milestone, there are avenues for future enhancements and refinements to elevate the system's capabilities.

- **Integration with Additional Sensors**

  Expand the system by integrating additional sensors to capture a more comprehensive set of environmental parameters. This could include air quality, soil moisture, or other relevant metrics for a holistic monitoring approach.

- **Advanced Predictive Models**

  Explore advanced machine learning models or ensemble methods to further improve predictive accuracy. This could involve the incorporation of deep learning architectures or hybrid models to handle intricate relationships within the data.

- **Real-Time Monitoring and Alerts**

  Implement real-time monitoring capabilities and alerts, enabling users to receive instant notifications about significant environmental changes. This feature enhances the system's responsiveness and usefulness in dynamic scenarios.

## 5.3. SOCIETAL IMPACTS

Our environmental monitoring system not only addresses specific application scenarios but also holds the potential for broader societal impact.

- **Resource Conservation**

  By facilitating informed decision-making in agriculture and energy usage, the system contributes to resource conservation, aligning with sustainable practices.

- **Resilience to Environmental Changes**

  The predictive capabilities empower individuals and industries to build resilience against environmental changes, mitigating risks and optimizing resource utilization.

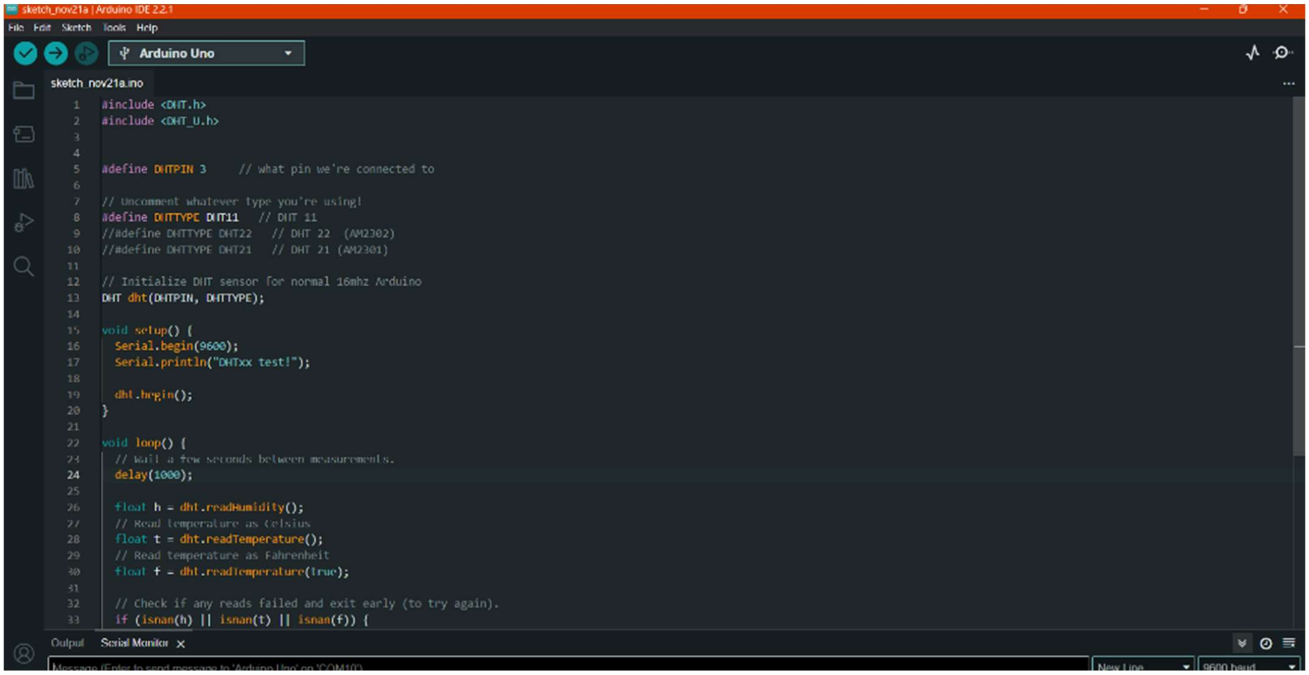
## 5.4. RESULT AND DISCUSSION

Various different models were applied successfully of which the Random forest Classifier gave the highest accuracy of 98%. The confusion matrix was also prepared for it. After applying the adversarial attacks like Fast Gradient Sign Method (FGSM) the accuracy of the model dropped to 48%. Further different models were applied to make our model resistant to these attacks. The usage of these models is vast and can be a future topic for research as well. The Model successfully predicted the temperature provided the humidity at any given time and region. For example it predicted a temperature of 24 degrees Celsius at 18% Humidity.

```
Total Number of Samples: 190502
Confusion Matrix:
[[18867   276]
 [  299 18659]]
Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.99      0.98     19143
           1       0.99      0.98      0.98     18958

    accuracy                           0.98     38101
   macro avg       0.98      0.98      0.98     38101
weighted avg       0.98      0.98      0.98     38101

Confusion Matrix (Adversarial Model):
[[ 7710 11433]
 [ 8253 10705]]
Classification Report (Adversarial Model):
              precision    recall  f1-score   support

           0       0.48      0.40      0.44     19143
           1       0.48      0.56      0.52     18958

    accuracy                           0.48     38101
   macro avg       0.48      0.48      0.48     38101
weighted avg       0.48      0.48      0.48     38101

The temperature at 18% Humidity is 24 degree celcius
```

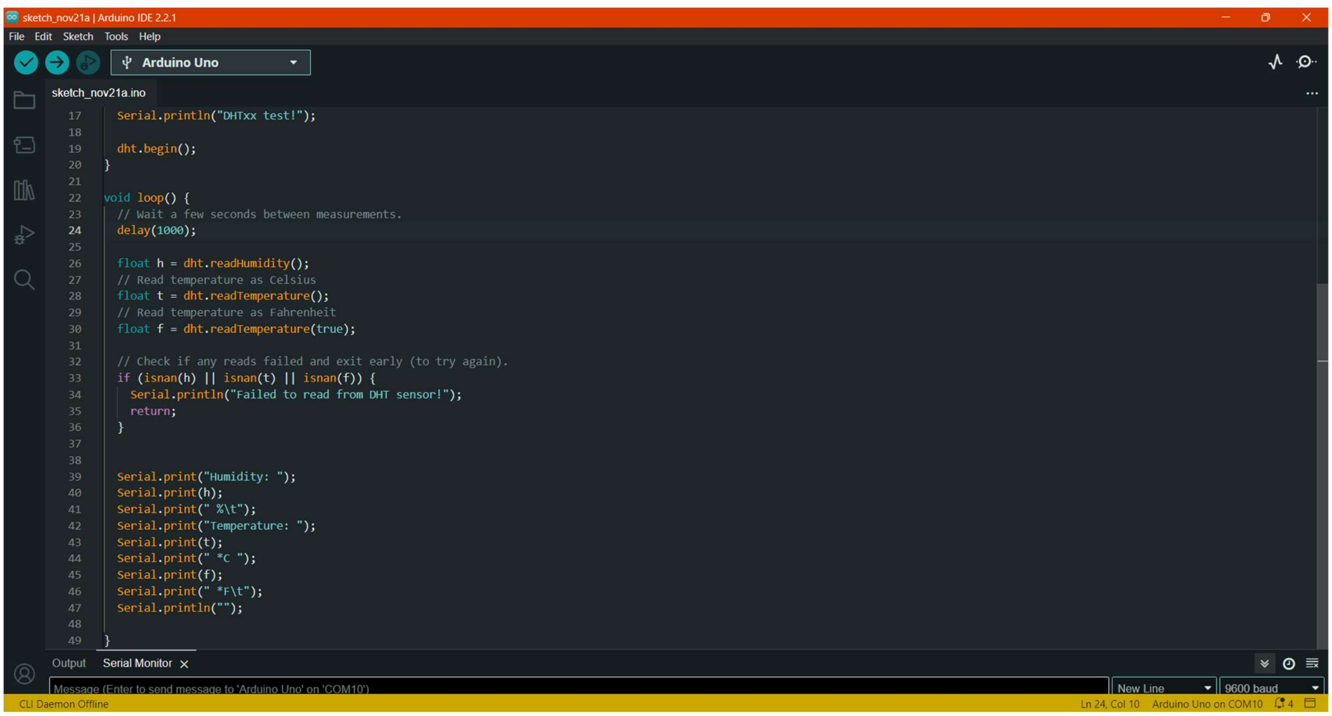
*Figure 7 – Output of the Random Forest Classifier Model*

Further the Arduino IDE was used to code the Arduino boards and develop a broadcasting and receiver device.



```
#include <DHT.h>
#include <DHT_U.h>


#define DHTPIN 3     // what pin we're connected to

// Uncomment whatever type you're using!
#define DHTTYPE DHT11   // DHT 11
//#define DHTTYPE DHT22   // DHT 22  (AM2302)
//#define DHTTYPE DHT21   // DHT 21 (AM2301)

// Initialize DHT sensor for normal 16mhz Arduino
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  Serial.println("DHTxx test!");

  dht.begin();
}

void loop() {
  // Wait a few seconds between measurements.
  delay(1000);

  float h = dht.readHumidity();
  // Read temperature as Celsius
  float t = dht.readTemperature();
  // Read temperature as Fahrenheit
  float f = dht.readTemperature(true);

  // Check if any reads failed and exit early (to try again).
  if (isnan(h) || isnan(t) || isnan(f)) {
```

*Figure 8 – Arduino Code*



```
  Serial.println("DHTxx test!");

  dht.begin();
}

void loop() {
  // Wait a few seconds between measurements.
  delay(1000);

  float h = dht.readHumidity();
  // Read temperature as Celsius
  float t = dht.readTemperature();
  // Read temperature as Fahrenheit
  float f = dht.readTemperature(true);

  // Check if any reads failed and exit early (to try again).
  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }


  Serial.print("Humidity: ");
  Serial.print(h);
  Serial.print(" %\t");
  Serial.print("Temperature: ");
  Serial.print(t);
  Serial.print(" *C ");
  Serial.print(f);
  Serial.print(" *F\t");
  Serial.println("");
}
```

*Figure 9 – Arduino Code*

The output that we got on the serial monitor of the Arduino IDE was the data collected by the DHT11 sensor. It showed us the values of the temperature and humidity.

```
Humidity: 61.00 %       Temperature: 28.70 *C 83.66 *F
Humidity: 61.00 %       Temperature: 28.70 *C 83.66 *F
Humidity: 60.00 %       Temperature: 28.60 *C 83.48 *F
Humidity: 60.00 %       Temperature: 28.60 *C 83.48 *F
Humidity: 60.00 %       Temperature: 28.60 *C 83.48 *F
Humidity: 60.00 %       Temperature: 28.60 *C 83.48 *F
Humidity: 60.00 %       Temperature: 28.60 *C 83.48 *F
Humidity: 60.00 %       Temperature: 28.60 *C 83.48 *F
Humidity: 60.00 %       Temperature: 28.60 *C 83.48 *F
Humidity: 60.00 %       Temperature: 28.60 *C 83.48 *F
Humidity: 60.00 %       Temperature: 28.60 *C 83.48 *F
Humidity: 60.00 %       Temperature: 28.60 *C 83.48 *F
Humidity: 60.00 %       Temperature: 28.70 *C 83.66 *F
Humidity: 60.00 %       Temperature: 28.70 *C 83.66 *F
Humidity: 60.00 %       Temperature: 28.70 *C 83.66 *F
Humidity: 60.00 %       Temperature: 28.70 *C 83.66 *F
Humidity: 60.00 %       Temperature: 28.60 *C 83.48 *F
Humidity: 60.00 %       Temperature: 28.60 *C 83.48 *F
Humidity: 60.00 %       Temperature: 28.60 *C 83.48 *F
Humidity: 60.00 %       Temperature: 28.60 *C 83.48 *F
Humidity: 60.00 %       Temperature: 28.60 *C 83.48 *F
Humidity: 60.00 %       Temperature: 28.60 *C 83.48 *F
Humidity: 60.00 %       Temperature: 28.60 *C 83.48 *F
Humidity: 60.00 %       Temperature: 28.60 *C 83.48 *F
Humidity: 60.00 %       Temperature: 28.60 *C 83.48 *F
Humidity: 60.00 %       Temperature: 28.60 *C 83.48 *F
Humidity: 59.00 %       Temperature: 28.60 *C 83.48 *F
Humidity: 59.00 %       Temperature: 28.60 *C 83.48 *F
Humidity: 59.00 %       Temperature: 28.60 *C 83.48 *F
Humidity: 59.00 %       Temperature: 28.60 *C 83.48 *F
Humidity: 59.00 %       Temperature: 28.60 *C 83.48 *F
Humidity: 59.00 %       Temperature: 28.60 *C 83.48 *F

emon Offline
```

*Figure 10 – Temperature and Humidity Data*

## 5.5. CONCLUSION

In conclusion, this project has successfully demonstrated the integration of LoRa communication, environmental data collection, and predictive modeling to create a versatile environmental monitoring system. The practical significance of the system across agriculture, smart homes, and industrial operations underscores its potential societal impact. Future enhancements promise even greater utility and precision, making this project a stepping stone toward innovative solutions in the field of environmental monitoring. As we look ahead, the journey of exploration and refinement continues, driven by the commitment to contribute meaningfully to societal and environmental well-being.

# REFERENCES


1. Chakraborty, A., Alam, M., Dey, V., Chattopadhyay, A. and Mukhopadhyay, D. (2021), A survey on adversarial attacks and defences. CAAI Trans. Intell. Technol, 6: 25-45. https://doi.org/10.1049/cit2.12028
2. Z. Yang, X. Liu, T. Li, D. Wu, J. Wang, Y. Zhao, and H. Han, "A systematic literature review of methods and datasets for anomaly-based network intrusion detection," Computers & Security, p. 102675, 2022
3. A. A. Pavate and R. Bansode, "Generation of adversarial mechanisms in deep neural networks: A survey of the state of the art," International Journal of Ambient Computing and Intelligence (IJACI), vol. 13, no. 1, pp. 1–18, 2022.
4. Z. Lin, Y. Shi, and Z. Xue, "IDSGAN: Generative adversarial networks for attack generation against intrusion detection," arXiv preprint arXiv:1809.02077, 2018
5. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in Proc. Adv. Neural Inf. Process. Syst., 2012, pp. 1097–1105
6. R. S. Sinha, Y. Wei, and S.-H. Hwang, "A survey on LPWA technology: LoRa and NB-IoT," ICT Exp., vol. 3, no. 1, pp. 14–21, 2017.
7. Dwivedi S, Tata Rao L, Goel S, Dubey SK, Javed A, Goel S. Statistical analysis and optimization of fuel cells using the design of experiment. *Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering*. 2022;0(0).
8. Modern digital and analog communication systems by B. P. Lathi, Zhi Ding