ECE 445

SENIOR DESIGN LABORATORY

PROJECT PROPOSAL

ML-based Weather Forecast on Arduino

Team #26

XUANYU CHEN
(xuanyuc2@illinois.edu)
ZHEYU FU
(zheyufu2@illinois.edu)
ZHENTING QI
(qi11@illinois.edu)
CHENZHI YUAN
(chenzhi2@illinois.edu)

Sponsor: Cristoforo Dimartino

TA: Jingyu Wei

March 29, 2023

Contents

1 Introduction

1.1 Problem

Weather forecasting is crucial in our daily lives. It allows us to make proper plans and get prepared for extreme conditions in advance. However, weather forecasts in areas far away from weather stations are not always reliable, since the real-time weather conditions are simulated according to the distance and altitude instead of direct measurements. To overcome the limitations of traditional weather forecasting, machine learning models have become increasingly important in weather forecasting. Building our own weather forecast machine learning system is a perfect idea for us to analyze vast amounts of area data and generate more accurate and timely weather predictions on the go in our surrounding areas.

1.2 Solution

A weather forecast system can be created by using a few different hardware components and software tools. Our solution mainly consists of two parts: weather stations and forecasts. As to the weather measurement and data collection part, temperature, humidity, barometric pressure, and rain sensors are considered the main components. For the other part, a machine learning (ML) based algorithm is to be applied for data analysis and weather predictions. Also, if time permits, we plan to visualize our timely weather prediction on a website based on Flask or other web application frameworks.

1.2.1 Solution Components

1. Hardware Subsystem
   Due to the complexity of weather conditions, our system incorporates the following weather indicators and their corresponding collectors: a humidity and temperature sensor, a barometric pressure sensor, a rain sensor, a light sensor, and an anemometer for wind speed. The aforementioned equipment will be integrated into Arduino, covered with a waterproof enclosure. The power supply for the weather station consists of batteries and a solar panel charging circuit. The outdoor weather data collector will send to an indoor Arduino receiver with a display screen via wireless communication. Then, the receiver will send the collected data to the computer via the serial interface.

2. Software Subsystem
   A practically usable weather forecast system is supposed to make reliable predictions for real-world multi-variable weather conditions. We apply machine learning techniques to suffice such generalization to unseen data. To this end, a high-quality dataset for training and evaluating the machine learning model is required, and a specially designed machine learning model would be developed on such a dataset. After a preliminary investigation, we have downloaded Haining's weather datasets for the past 40 years from the OpenWeather platform. And we believe auto-

regressive models could serve as practical solutions for our model. Once a well-trained machine-learning model is obtained, we will deploy the model on portable devices with easy-to-use APIs.

1.3 Visual Aid

Visual aid is modified.

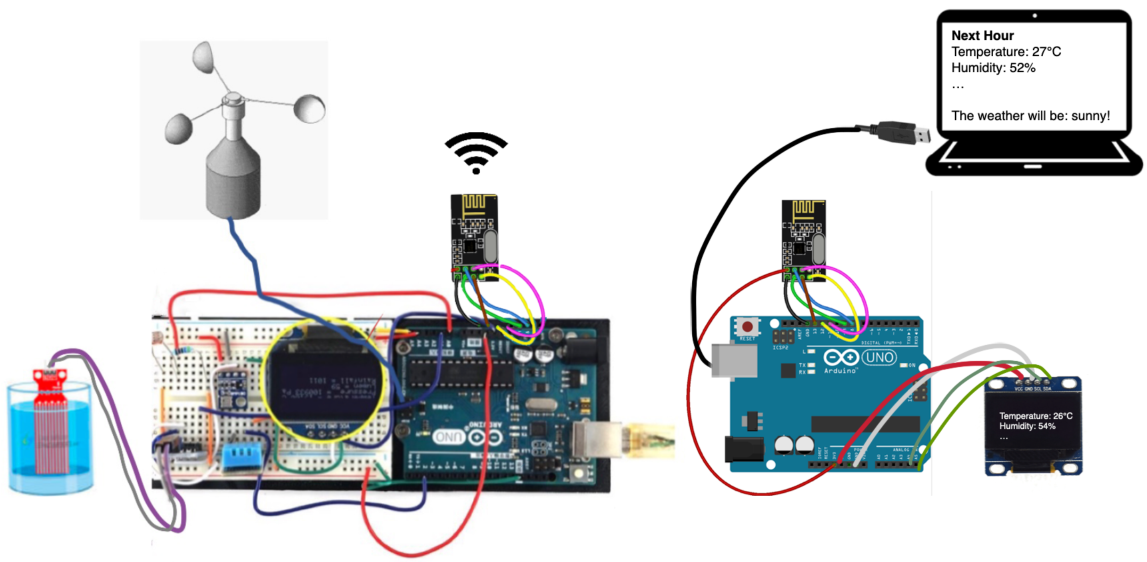

Figure 1: The entire system looks like this.

1.4 High-level requirements list

- The weather measurement prototype with sensors should be able to accurately collect the temperature, humidity, and barometric pressure. etc. To be more specific, our collected weather parameters will be compared to real-time data from a meteorological station and should be within a reasonable margin of error. For example, temperature error should be within 5%, humidity error, and barometric error within 10%.
- A machine learning algorithm should be successfully trained to make predictions on the weather conditions: rainy, sunny, thunderstorm, etc. This is a multi-classification problem and we hope our macro-average precision could reach 0.75 or higher.
- Our system can collect and forecast the weather in Haining, in real-time, and/or longer-period forecast. To be specific, our hardware subsystem could collect and upload real-time weather information to the software subsystem per 30 minutes. And our software subsystem could predict the following 24-hour's weather condition based on the recently inputted 48 hours weather parameters. For longer-period prediction, our system will be able to predict the weather up to 7 days later, but it could be a great challenge to predict accurately in this case as you can imagine.

- The forecast weather information could be demonstrated elegantly through some UI interface. A display screen would be a baseline, and a web application on a phone or PC would be extra credit if time permitted.

2 Design

2.1 Block Diagram

Figure 2 illustrates our entire design.

The block diagram is modified.

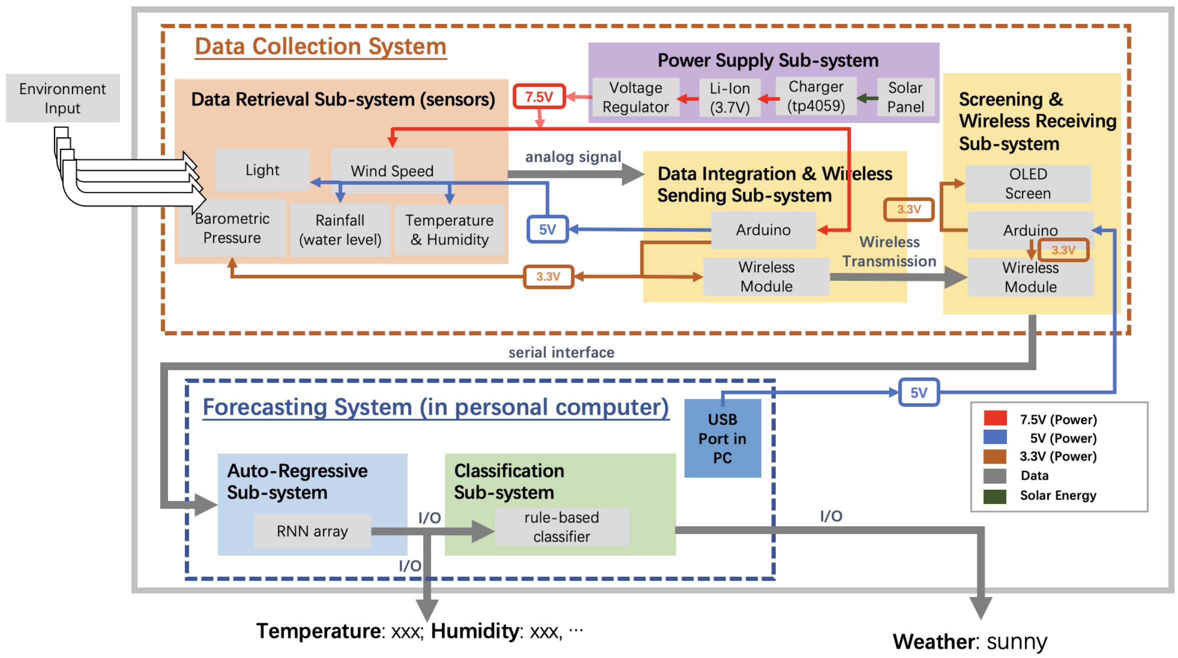

Figure 2: Block Diagram.

2.2 Subsystem Overview

Our weather forecast system mainly consists of two parts: software for the machine learning training model and hardware for the weather station.

2.2.1 Software: ML-Based Forecasting Subsystem

Deep learning (DL) models have shown their extraordinary ability to discover complex patterns and features of various types of data. Auto-regressive models, an important branch of DL models, are specifically designed for processing and generating time series, and since our algorithm's goal is to predict the following values in a sequence of values after taking in the current ones, we leverage auto-regressive models' ability to achieve our goal. The model is trained with an MSE loss function and is expected to predict one future value based on existing ones (Fig. 3).

Other than forecasting each weather indicator (temperature, humidity, etc.), we hope to predict an overall description of the weather, such as sunny and rainy. To this end, we apply an extra classifier (implemented with a multi-layer perceptron or a decision tree) to perform k-class classification (Fig. 4).

Requirements

- Predicted values for each weather indicator should deviate from true values in an acceptable range. Data predicted would be compared with 1) previously measured data and 2) online weather forecasts. We expect a 10% and 50% relative difference from online values and the last measured values, respectively, for all five parameters.

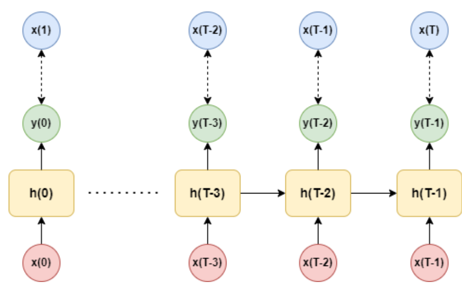

Figure 3: Training an Auto-Regressive Model with Many-to-Many Sequence Generation.

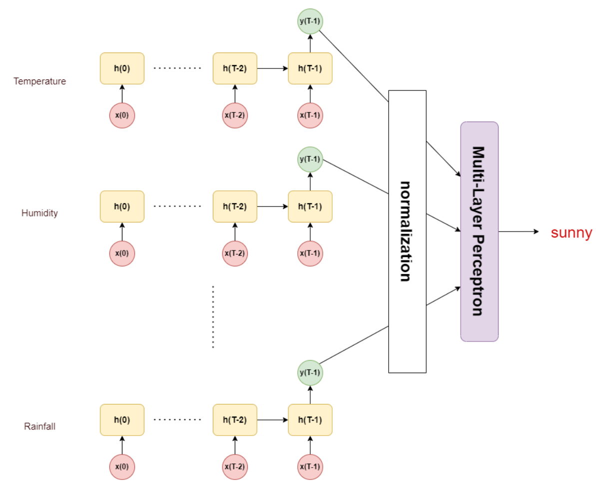

Figure 4: The Entire Forecasting System.

2.2.2 Hardware: Data Retrieval Subsystem

The whole subsection is changed.

Our weather data retrieval subsystem is used for collecting real-time weather data and transmitting them to our software subsystem for prediction. This subsystem will be deployed on Arduino Uno, a microcontroller board based on the ATmega328P. Applied sensors include a temperature and humidity sensor, a barometric pressure sensor, a raindrop sensor, a light sensor, and an anemometer. The real-time collected data will be transmitted wirelessly to the indoor receiver.

Firstly, we have to make sure that the selected weather indicators are comprehensive enough to be generalized to an output conclusion of weather. Secondly, to ensure that the measured weather parameters are accurate and perfectly match our needs, we pay great attention to the sensor selections. After a detailed survey and thorough discussion, we decided to measure data on temperature, humidity, barometric pressure, rain level, sunlight intensity, and wind speed. Based on these demands, we select the following sensors connected with Arduino via analog signal interfaces. Our selections are based on historical weather parameters' ranges shown in table 1, measurement accuracy (at most $\pm 5\%$), and power consumption.

	Min	*Max*
Temperature (Celsius)	-10.94	45.56
Humidity (%)	3.2	1.2
Pressure (hPa)	979	1044
Rain (mm/h)	0	22.38
Wind Speed (meter/sec)	0.01	16.12

Table 1: Historical Weather Data in Haining

- Temperature and humidity are definitely the most important weather parameters. We choose the DHT11 sensor [1] that measures the two parameters synchronously.
- For barometric pressure, we choose to use BMP 180 module [2] to sense because of its high sensibility and super low power consumption. The chip consumes less than 1mA during measurements.
- To measure the rainfall per hour, we plan to design a rain barrel to collect the raindrops. And the water level sensor [3] is used to measure the accumulated rain level.
- We choose a photoresistor [4] and an anemometer [5] as our light and wind sensors respectively.

To protect the sensor circuit, we will also design a highly waterproof shell for this subsystem. Since the shell must be placed under sunlight, the shell should have considerable UV resistance.

Requirements

- The sensor subsystem should be able to measure and output accurate data. To ensure these components function correctly, we have to support a suitable voltage source. The 5V logic microcontroller, Arduino Uno will act as the main voltage source. Specifically, the voltage applied to the anemometer should be above 7V.
- The temperature and humidity sensor should be able to send out an 80μs-long low-voltage-level response signal to indicate data preparation and then pull up the voltage.
- Our collected data should reach certain accuracy compared with the data from OpenWeather, the platform where we retrieve Haining's historical weather data. Data readings will be compared to the weather station's reports in our measured area. We expect a 5% relative difference in temperature, and 10% in other parameters.
- The enclosure of the box should totally prevent the dust and water jet into the outfit. Taking the IP waterproof as a reference, our outfit enclosure should at least reach

the IP65 waterproof. This IP65 waterproof means no ingress of dust whatsoever permitted, and no ingress from precipitation permitted.

2.2.3 Hardware: Wireless Communication Subsystem

This subsection is added.

The sensor-collected data needs to be transmitted to our personal computers for machine-learning weather prediction. Since our sensors will be placed in an outdoor environment that probably gets caught in rain, we decide to use wireless communication instead of a traditional wired connection to transmit the data to indoors. Also, an OLED screen is installed to demonstrate the sensing data in real time. As a consequence, we design our data transmission system to be two subsystems, i.e. the wireless sending and receiving subsystem shown in the block diagram.

- **Wireless transmission subsystem using transceiver**: An SPI-connected digital transceiver NRF24L01 is used to wirelessly transmit weather data from the outdoor Arduino connected with weather sensors to the indoor Arduino near our personal computer.
- **Wired transmission subsystem using serial port**: We use the serial port to transmit data from the indoor Arduino to our personal computer. Also, we will display all of the sensed weather data on an OLED screen installed on the indoor Arduino.

Requirements

- Our whole transmission module should be able to transmit the data correctly without any data loss with a wireless transmission distance to be at least $6m$.
- Our OLED screen installed on the indoor Arduino could demonstrate the real-time weather data successfully.

2.2.4 Hardware: Power Supply Subsystem

This subsection is added.

For our outdoor weather station, the power source of the Arduino Board is serial batteries (3.7V) with a DC-to-DC Converter Module to boost the voltage supply from the battery. Three voltage levels are needed, including 3.3V for the water-level sensor, 7.5V (VIN) for the anemometer, and 5V for the remaining components. For the indoor receiver, Arduino will be charged with the USB cable.

To ensure that our system won't suffer from failure because of a dead battery, a solar panel charging system is included to ensure the sensor module with the Arduino is powered continuously. The Lithium-Ion battery will be charged through a charger, the TP4059, powered by the solar panel. The charger is chosen for its affordability and thermal feedback regulations.

Requirements

- The battery-power supply for the sensing module should provide over 3V for at least 24-hour duration.

- The solar charging circuit should supply at least 4.2V to charge the battery.

2.3 Tolerence Analysis

The whole section is changed.

2.3.1 Hardware

Power Supply Analysis for the data retrieval subsystem We roughly estimate the working power for our remotely placed sensing subsystem as the following table states. It is essential to do the estimation since our sensing subsystem will be placed outside in an outdoor environment and will be powered by our self-designed power system.

Components	**Power**
Arduino Uno	Power: $\approx 450mW$ from Youtube [6]
Wind speed sensor	Power: $\leq 300mW$ from its data sheet [5]
Water level sensor	Working Voltage: $5V$ Working Current: $\leq 20mA$ Power: $\approx 100mW$
DTH11: Temperature and Humidity Module	Working Voltage: $5V$ Working Current: $0.3mA$ when measuring, $60\mu A$ when standby Power: $\approx 0.3mW$
BMP180 Digital pressure sensor	Working Voltage: $3.3V$ Working Current: $5\mu A$ Power: $\approx 0.02mW$
Light detector sensor	Working Voltage: $5V$ Working Current: $\leq 5\mu A$ Power: $\approx 0.03mW$
nRF24L01 Wireless Module	Working Voltage: $3.3V$ Working Current: $\leq 13.5\mu A$ Power: $\approx 44.55mW$
Other auxiliary resistors	Power: $\leq 100mW$

Table 2: Power Usage Analysis

As shown in table 2, the total power consumption is estimated to be 1W. Our battery component provides a 3.7V voltage supply and has $4000 * 2 = 8000mAh$, thus providing $3.7V * 8000mAh * 3600s = 106560J$. Our battery power supply can work for at most $\frac{106560J}{1W} = 29.6h$, which will be sufficient for our basic goal of a whole-day usage. Moreover, our chosen solar panel can supply at most 1.1A under the sun. Assume that sunlight is available for 4 hours on average (the assumption is reasonable referring to Haining's weather dataset). We are able to refill the battery $\frac{4000mAh*2}{1.1A*2} = 3.63$ with double-panel once a day at the best case. Based on the aforementioned estimation, our power supply system is able to produce adequate energy for our weather station.

2.3.2 Software

In the case of the forecasting model, potential problems may arise due to a lack of accuracy or precision in the data inputs.

Denote the model as a function f. Given weather data sequences $S = \{X_{temp}, X_{humidity}, ...\}$, where $X = [x_t, x_{t+1}, ..., x_{t+n-1}]$ of length n at time step t, the model predicts the next-step values $Y = \{y_{temp}, y_{humidity}, ...\}$ by $Y = f(S)$. Therefore, each predicted value y is contingent on all five types of measured data (temperature, humidity, pressure, rainfall, and wind speed). Since we use MSE (mean square error) to evaluate the prediction quality, which is calculated by $\mathbf{MSE}(Y, \hat{Y}) = \frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2$, the error caused by wrongly measured data would be of **second order**. So the most important tolerance in the software part comes from the robustness of the model against the incorrectness of measured data.

If we do nothing to the input data, the error would be magnified to a greater extent due to the difference in units. If both deviate the same percentage from actual values, the error caused by pressure (of order 1×10^3) would be $10^{(3-1)*2}$ times as much as the temperature (of order 1×10^1). We leverage **normalization** on the input data to erase the dimensional differences (e.g. 19°C of temperature and 1031 hPa of pressure) so that each weather feature contributes to the prediction results to the same extent. The L_p normalization for an input vector v scales the vector into range $[0, 1]$, which is calculated by:

$$v = \frac{v}{\mathbf{max}(||v||_p, \epsilon)},$$

where ϵ is a small value to avoid division by zero. Since all features are mapped to the same range, the difference in units would not be a problem anymore.

Besides, we continuously monitor and validate the model's performance and recalibrate it according to previously learned data patterns as necessary. We set the threshold of predicted values' deviation from the real values (measured in the next time step) to be **10%**. That is, the predicted value for the next time step $\hat{y}$ should satisfy $\frac{|y-\hat{y}|}{|y|} \leq 10\%$ (after normalization). If this threshold has been exceeded for certain times, we would be informed by the system and improve the model design. Incorporating more data sources or modifying the model's architecture may also be necessary to improve its accuracy and robustness.

Overall, anticipating and addressing potential failures in both hardware and software components is crucial to ensuring the reliability and accuracy of the weather forecasting system. It requires a comprehensive approach that combines proper maintenance, monitoring, redundancy measures, and continuous improvement to ensure optimal performance.

3 Ethics and Safety

3.1 Ethics

According to the IEEE Code of Ethics 1 [7], we should hold paramount the safety, health, and welfare of the public, and strive to comply with ethical design and sustainable development practices. Following this, We are committed to designing our weather forecast equipment to be both practical and ethical. We hope our design could be a next-generation solution for a mini-size meteorological station, and we will try to prevent any form of behavior that may endanger the public who order our products.

Also, the IEEE Code of Ethics states that engineers shall maintain confidentiality and protect the privacy of others. We ensure that we use legally available datasets, like Haining's weather conditions from the OpenWeather platform, for training our machine learning models. To avoid ethical breaches, we may implement appropriate measures to safeguard the privacy of data, such as encryption and secure storage.

Moreover, In line with the IEEE Code of Ethics 4 [7], it is important to actively seek, accept, and provide sincere critiques of technical work while acknowledging and rectifying any mistakes. We should be truthful and practical when making statements or predictions based on available data, and give appropriate recognition to the contributions of others.

3.2 Safety

According to the safety guidelines from the ECE445 course website [8], our team will abide by the rules below:

1. Minimum of two people must be present in the lab at all times to ensure our safety. Especially when doing some procedures with electricity, we should pay extra attention to the safety of ourselves as well as our teammates. Also, when we are doing soldering, material cutting, and 3d printing, we should be cautious, to avoid potential burns and cuts.

2. Everyone in our team will complete the mandatory online safety training, and submit our certificates of completion on Blackboard. This is to ensure our team has the necessary common sense for laboratory safety.

3. Generally, our project does not involve many dangerous operations. However, we should always be careful and cautious to ensure that our safety and health will not be affected.

References

[1] A. Electronics. “”Temperature and Humidity Module DHT11 Product Manual”.” (2014), [Online]. Available: https://components101.com/sites/default/files/component_datasheet/DHT11-Temperature-Sensor.pdf (visited on 03/07/2023).

[2] B. Sensortec. “”BMP180 Digital pressure sensor Datasheet”.” (2013), [Online]. Available: https://cdn-shop.adafruit.com/datasheets/BST-BMP180-DS000-09.pdf (visited on 03/07/2023).

[3] K. Robots. “”Water Sensor Module User’s Manual”.” (), [Online]. Available: https://curtocircuito.com.br/datasheet/sensor/nivel_de_agua_analogico.pdf (visited on 03/23/2023).

[4] S. H. S. Co. “”Ambient light detector Photosensitive sensor”.” (2014), [Online]. Available: https://www.arduino.cc/documents/datasheets/HW5P-1.pdf (visited on 03/07/2023).

[5] Q. Electronics. “”Anemometer Wind Speed Sensor Datasheet (in Chinese)”.” (), [Online]. Available: https://cdn-shop.adafruit.com/product-files/1733/C2192_datasheet.pdf (visited on 03/07/2023).

[6] M. Klements. “”How Long Can An Arduino Run On Batteries? I Tested 6 Of The Most Common Boards”.” (), [Online]. Available: https://www.youtube.com/watch?v=5cYN5-Spnos (visited on 03/24/2023).

[7] IEEE. “”IEEE Code of Ethics”.” (2016), [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html (visited on 03/07/2023).

[8] ECE470. “”ECE470 Safety Guidelines”.” (2023), [Online]. Available: https://courses.grainger.illinois.edu/ece445zjui/guidelines/safety.asp (visited on 03/07/2023).