Weather-triggered Wireless Telemetry System

Senior Design May’25 Team 18

Alex Chambers, Alexander Christie, Adam Fields, Aidan Gull, Colin Kempf, Nisha Raj
Client: Daji Qiao
Advisor: Sarath Babu

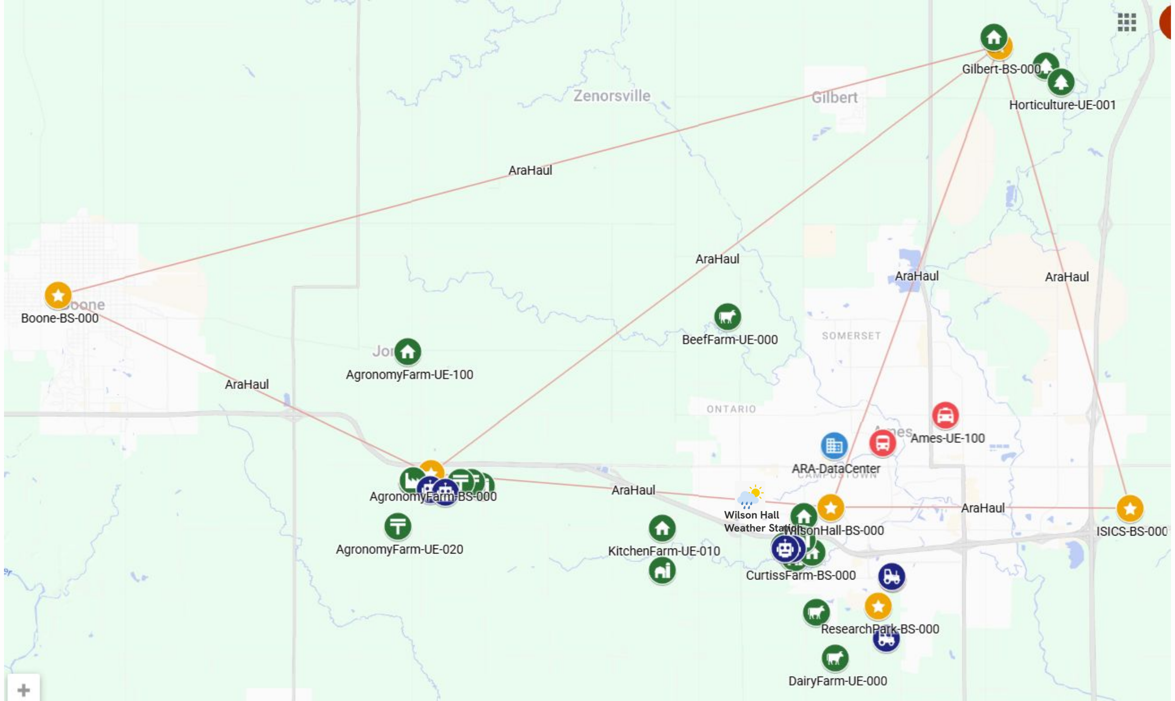
Client Information

- ARA: at scale advanced wireless research platform covering Iowa State University, Ames and surrounding rural areas
- ARA deployments span a 60km diameter
- Serves as a wireless lab enabling research into rural focused wireless technologies
- Research teams worldwide utilize ARA testbed to conduct wireless experiments
- ARA infrastructure includes
  - AraRAN: SDRs, mMIMO, Ericsson COTS &
  - AraHaul: mmWave, microwave and free space optical links
  - Weather Stations: used for weather-based wireless experimentation
- Offers speeds up to several hundred Gpbs

ARA Infrastructure

Key:

- Yellow: Base Stations
- Green: Fixed UE Sites
- Dark Blue/Red: Mobile UE Sites
- Light Blue: Data Center



Project Plan & Management

Project Overview

Purpose

- Analysis of ARA wireless platform performance during weather events
- Automatically generate datasets to enable researchers to study how weather events affect wireless platforms

Goal

- Develop a weather-triggered telemetry system that measures the performance of wireless technologies deployed on ARA before, during, and after weather events of interest

Tasks

- Collect network data during a variety of precipitation weather events such as rain or snow
- Store collected data so users can query and visualize formatted datasets

Approach

- Design and develop a telemetry system utilizing weather APIs
- System will continuously monitor for when a future weather events
- Recognize weather events of interest and set a lead-in time to trigger data collection
- Once a weather event ends, continue collecting data until sufficient lead-out time is complete
  - Lead-in and lead-out time shows the transitions between weather events
- Datasets collected will be formatted in a hierarchical ZIP file structure
  - Includes metadata such as time started/ended collecting and location of base station
- The datasets will be stored in a dedicated server using relational database model
- Create a user-friendly interactive UI so researchers can visualize collected datasets

Requirements

Functional

- Gather & store ARA data during weather events on a dedicated ARA server
- Explored publicly weather forecasting APIs to predict when a weather events will occur
- Trigger ARA data collection when weather is detected during lead-in time
- Consider lead-in and lead-out time where data is collected before and after weather event
- Create UI so users can visualize ARA data in graphical format
- Store data in ZIP file hierarchy and store ZIP path in database on a dedicated ARA server

Non-Functional

- Host telemetry system, database, and visualization framework on ARA
- Utilize ARA disdrometers and weather stations to collect fine grained weather data points
  - Examples: Precipitation, windspeed, particle velocity, and humidity
- Create accessible and intuitive UI that utilizes graphical visualization tools

Market Research - APIs

- Explored publicly available weather forecasting APIs that could be used for weather prediction metrics
- Six APIs stood out to us as potentially meeting our needs
- Determined Tomorrow Weather, Open-Meteo, and National Weather Service best met our needs

API	Issue/Feature
tomorrow.io	-Free -Localized weather data -Project integration well-documented -Many applicable data points
OpenWeather	-Requires subscription -Limited to 100 free API calls per day
Open-meteo	-No API key required -10,000 requests per day -Responds with data points chart
Weatherbit	-Limited to 50 API calls per day -Requires subscription -No data points for wind speed
NATIONAL WEATHER SERVICE NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION	-Free & open source -JSON formatting -Caches result of requests
AccuWeather	-Paid account needed for advanced features -Limited to 50 API calls per day -Free accounts only receive one API key

Project Risks & Mitigation

Risk	Mitigation
External API becomes unavailable	Utilize multiple different APIs
Unable to collect weather data from ARA framework	Work with ARA administrators to resolve data collection issue
Server running our script becomes unavailable	We will always have backups of our code; can request an additional server and have scripts running in parallel
Data corrupted or input not as expected	In data processing script create a check for corrupted of incorrectly formatted data
Causing hardware malfunctions	Ensure program is tested before merging with ARA testbed equipment

System Design

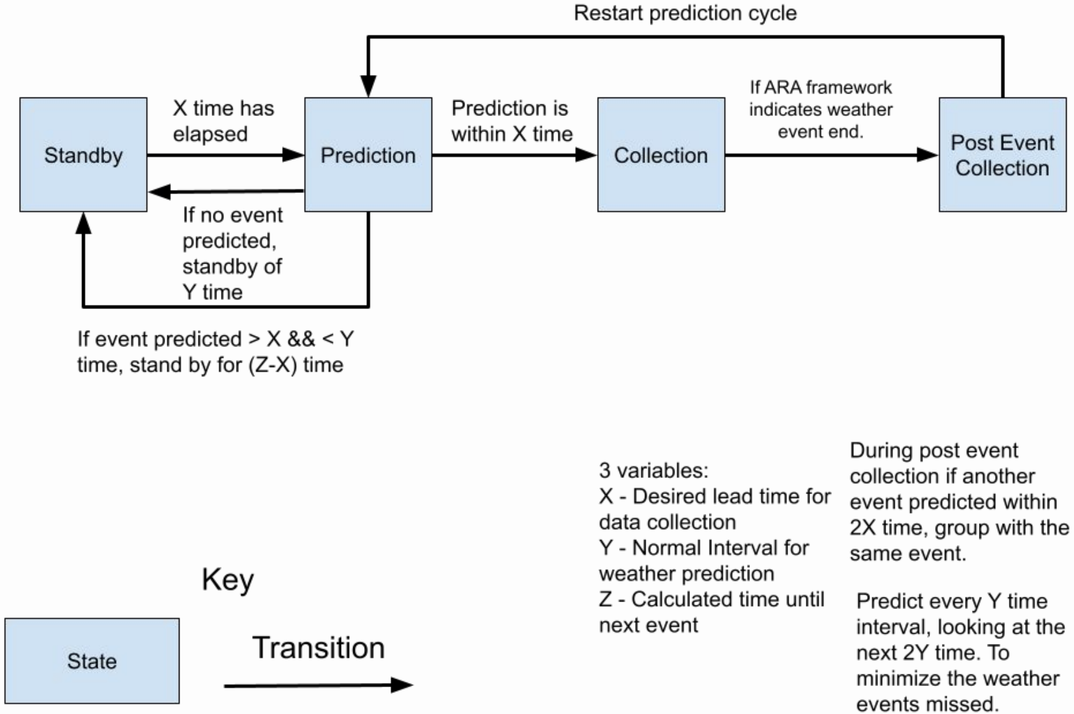
State Diagram

Standby: Passive state of software

Prediction: Software uses weather APIs to predict future events

Collection: Software state when weather data from ARA APIs is received to determine lead-in time for an event

Post Event Collection: Gathers data for lead-out time, stores collected data



Restart prediction cycle

Standby

X time has elapsed

Prediction

If no event predicted, standby of Y time

If event predicted > X && < Y time, stand by for (Z-X) time

Prediction is within X time

Collection

If ARA framework indicates weather event end.

Post Event Collection

3 variables:
X - Desired lead time for data collection
Y - Normal Interval for weather prediction
Z - Calculated time until next event

During post event collection if another event predicted within 2X time, group with the same event.

Predict every Y time interval, looking at the next 2Y time. To minimize the weather events missed.

Key

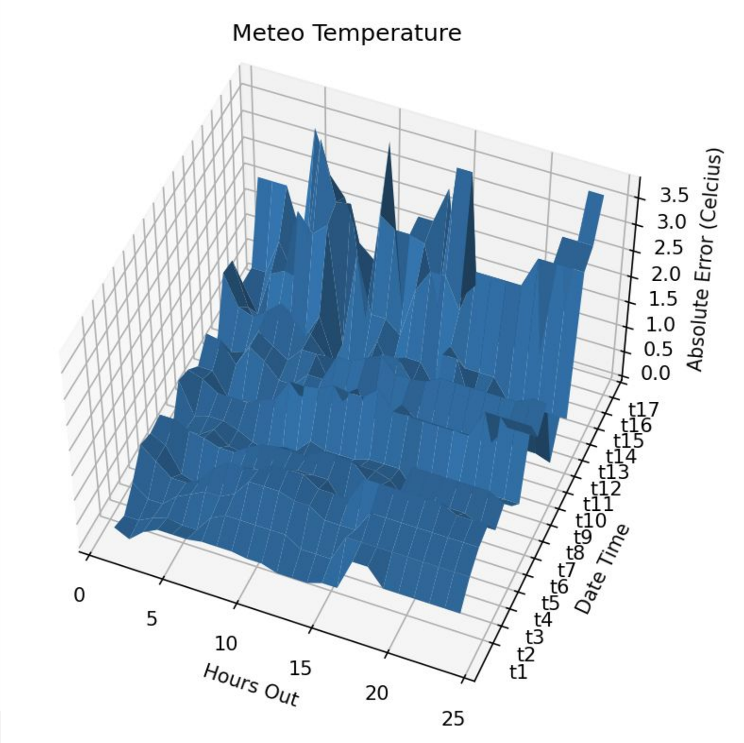
State

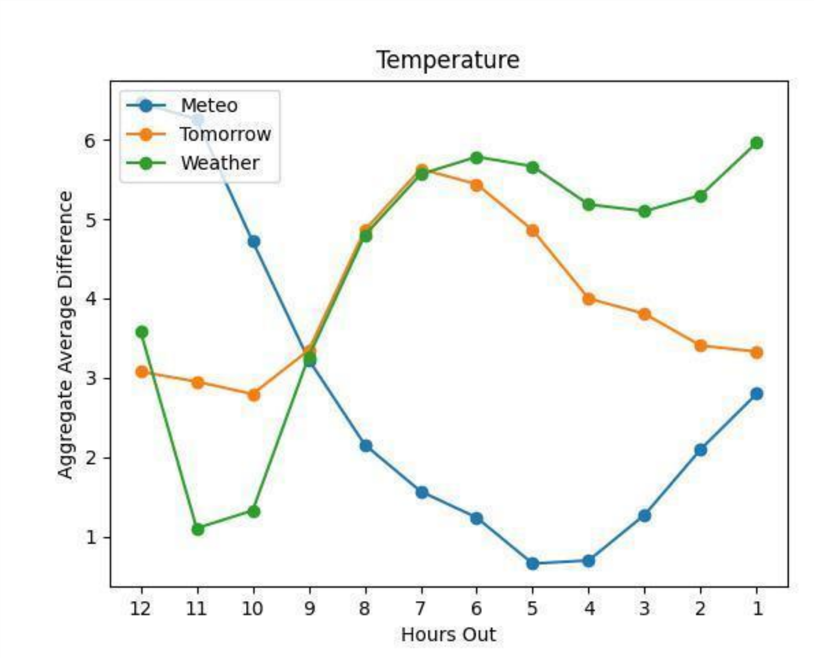
Transition

Initial Prototyping

- Created an initial prototype to gather forecasting data from the APIs
- Collects specific weather features from the APIs:
  - Temperature
  - Wind speeds
  - Humidity
  - Wind direction
- Collected data which is stored in csv files
- Used the collected data to analyze the accuracy of predicted features.
- Created graphs to assist in analysis, showing predicted features from APIs versus the actual weather data at that time

Prototype Visualization

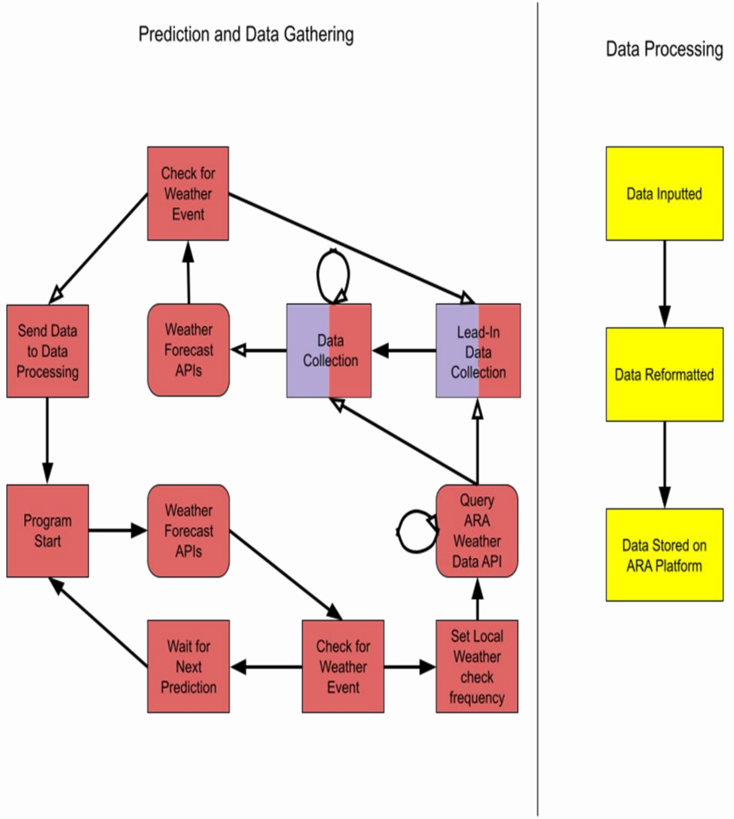




Implementation

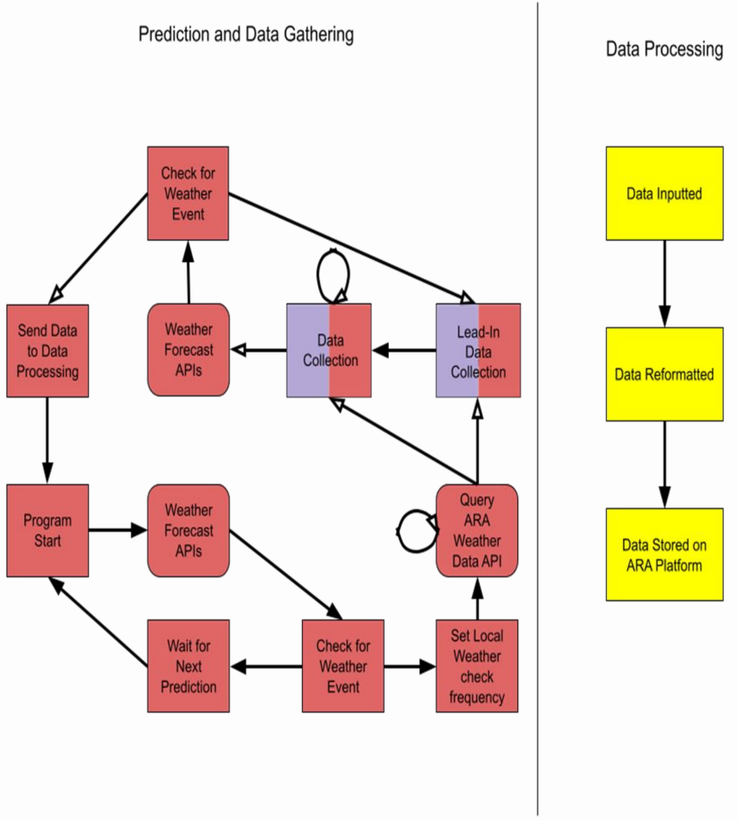
Component Breakdown

1. Weather Prediction and Collection
2. Wireless Data Collection
3. Data Storage and Visualization



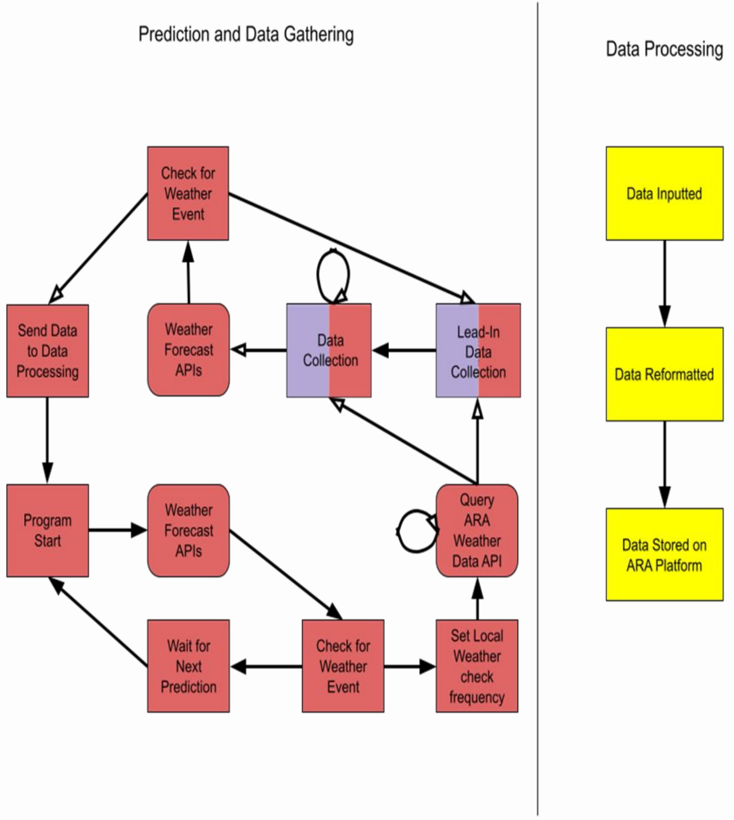
Weather Prediction and Collection

- Predict probability of a weather event occurring in the next hour based on APIs
- When ARA disdrometer detects weather event, start Wireless Data Collection subsystem
- Once weather event is over, send data to Data Storage and Display subsystem



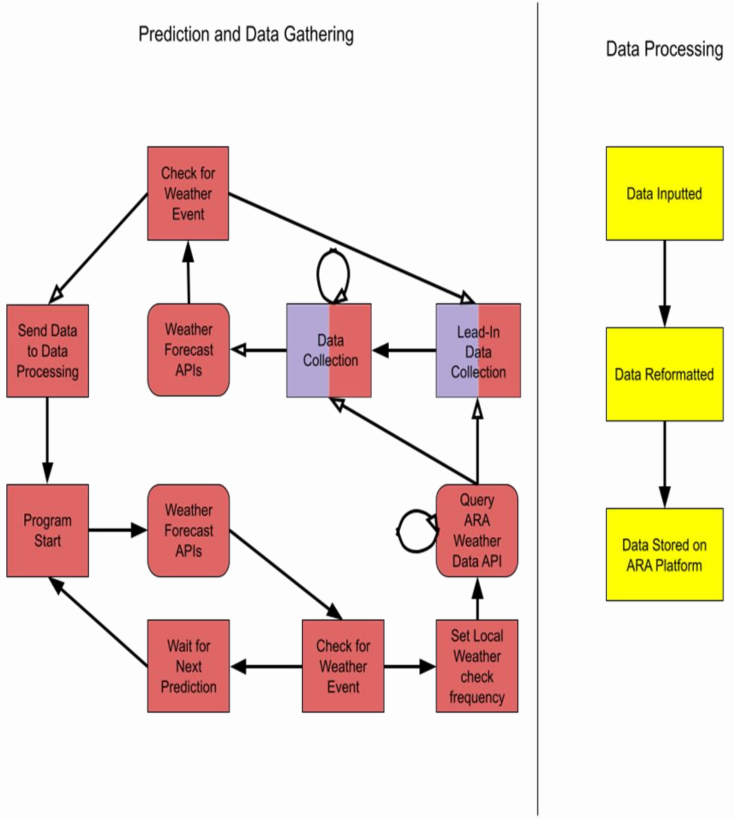
Wireless Data Collection

- Collecting and storing ARA Wireless Data from multiple devices
- When data collection begins ID generated and data collected in association
- ID returned to caller, when caller decides wireless data collection with associated ID is stopped



Data Storage and Visualization

- Handles collected data and format to be queried
- When weather event is complete, data is reformatted and stored in a hierarchical zip file and stored in ARA platform’s database

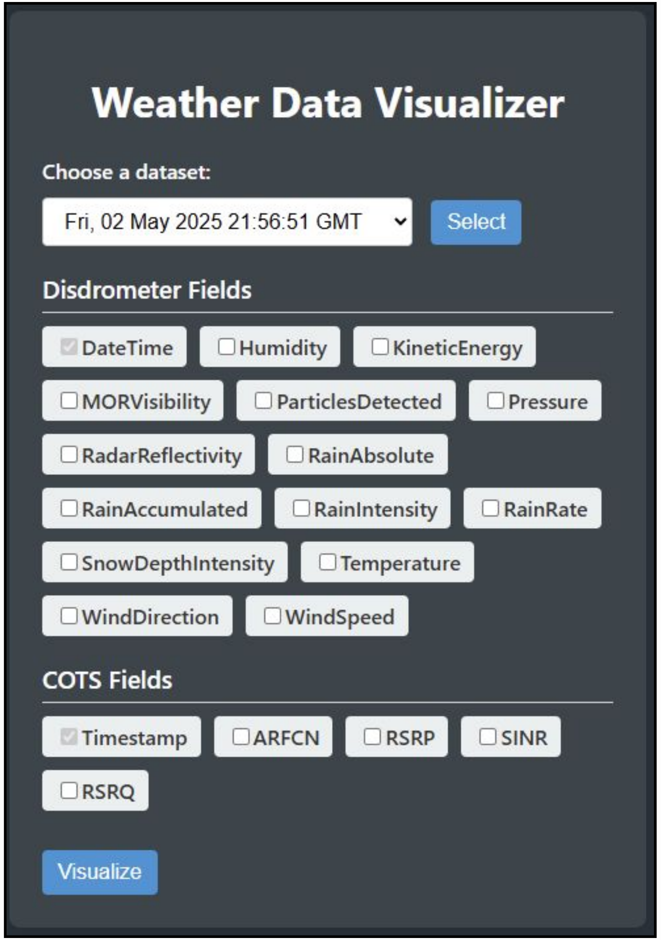


Telemetry System Deployment

- All code components run on the ARA server
- Weather prediction data will trigger wireless data collection
- The data collected is stored on a dedicated ARA server in a relational database
- ARA server runs HTML UI where Grafana visualization tool will be launched
- Server running local instance of Grafana
  - Graphs populated when user interacts with UI

UI Implementation

- UI running on the ARA server
- Accessible at URL while connected to VPN
- Drop down menu to choose which dataset to visualize
  - Data sets populated from the database
- User can choose which weather and wireless fields they would like to visualize
- Each field will populate its own graph in visualization tool



Weather Data Visualizer

Choose a dataset: Fri, 02 May 2025 21:56:51 GMT

Disdrometer Fields

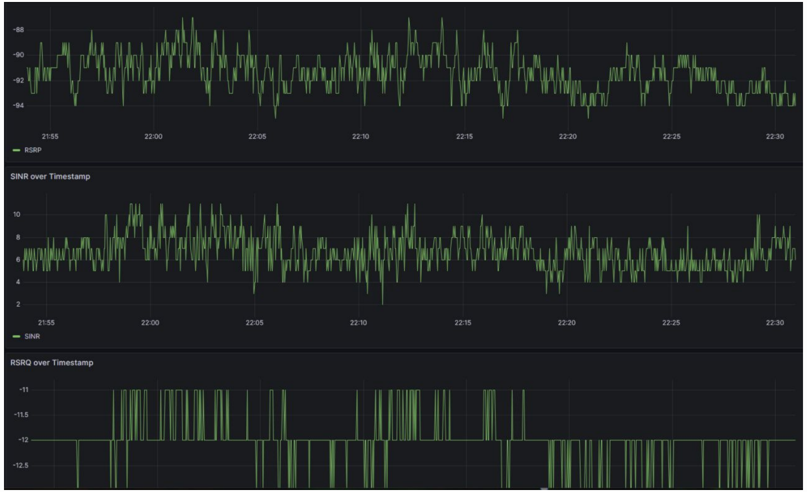
☑ DateTime ☐ Humidity ☐ KineticEnergy
☐ MORVisibility ☐ ParticlesDetected ☐ Pressure
☐ RadarReflectivity ☐ RainAbsolute
☐ RainAccumulated ☐ RainIntensity ☐ RainRate
☐ SnowDepthIntensity ☐ Temperature
☐ WindDirection ☐ WindSpeed

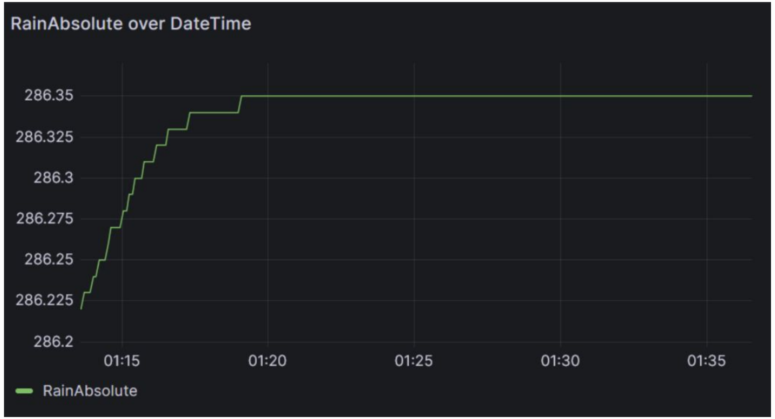
COTS Fields

☑ Timestamp ☐ ARFCN ☐ RSRP ☐ SINR
☐ RSRQ

Graphics Visualization

- Grafana visualization tool
    - Produces charts and graphs for data sources
- Using Grafana to visualize wireless and corresponding weather data
- Wireless data and weather data graphs will be individual graphs
- Each corresponding weather parameter field selected will have its own graph





Testing

Unit Testing

- Wrote several unit tests covering both our API predictions and integrated systems
- Tools we used for testing:
  - PyTest: specifically used for mocking a number of calls
  - Monkey Patch: helped bypass many API calls during tests
- Most of our functions directly interact with either APIs, direct data collection from devices (i.e. wireless data collection), and I/O writing and reading of data from files
- The need to mock a large number of these features when testing does make our tests more “brittle”, but was necessary for testing purposes.
- For the most part, our unit tests are parameterized tests which test different inputs and expected outputs of several functions allowing for significant coverage of different paths

System Testing

- Because our system runs real time and on multiple threads, full system testing can be extremely difficult
- Our goal was to have sufficient coverage, both through unit tests and integration tests focusing on either the entire prediction to event creation pipeline, or the data collection pipeline
- Additionally, we created dedicated tests for the COTS API
- Due to the unknowable nature of real time data collection, we relied on metamorphic testing to test its functionality
- All of these tests intandem we believe provide sufficient coverage of our system as a whole
- We have also made sure to do our own manual testing and logging to further guarantee correctness within our system

Acceptance Testing

- Goals of our Acceptance test
  - Percentage of weather event data points collected
  - Percentage of weather event lead in/out data points collected
  - Percentage of non-weather event data points collected
- Without a historical repository of ARA weather data, this is difficult to do
- Current system running has not missed any rain events while it has been up

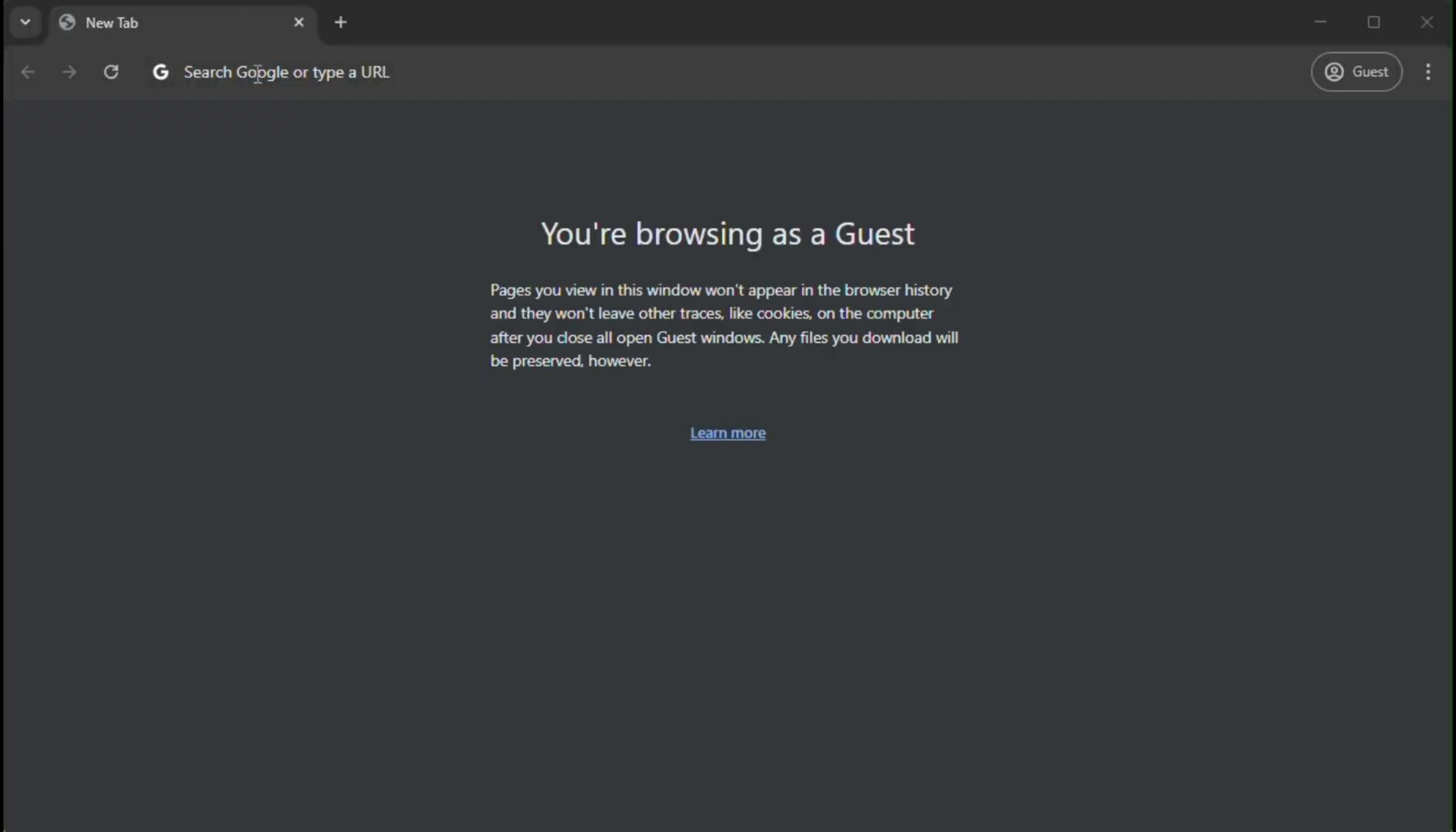
User Testing

- Testing for our UI and Grafana data visualization

- Individual UI Grafana testing
  - Simulated datasets to verify Grafana visualization
    - Ensured Web Application success independent of collection and formatting script

- Whole system UI testing
  - Real datasets from collection and formatting systems
    - Confirmed successful collection and formatting of entire system
    - Ensured UI could accurately display data collected, formatted, and stored by system

Demo



You're browsing as a Guest

Pages you view in this window won't appear in the browser history and they won't leave other traces, like cookies, on the computer after you close all open Guest windows. Any files you download will be preserved, however.

Learn more

Conclusion

Project Outcomes

- Designed, developed, and deployed a system utilizing external APIs to predict weather events which trigger weather and wireless data collection
- Developed and integrated scripts for wireless and weather data collection
- Utilized logic to implement lead-in and lead-out times for data collection
- Implemented external applications like Flask, MariaDB, and Grafana
- ARA researches able to collect and visualize wireless and weather data
    - Visualization UI accessed through ARA server
    - Data collection scripts running continuously on ARA server
- 9 datasets currently collected

Next Steps

- Expand UI to showcase more data points
- Improve accuracy of our prediction algorithm
- Refine README generation script
- Integrate current scripts with existing ARA experiments framework
- Incorporate additional wireless and weather nodes into our system
- Continue weather collection to build up ARA historical weather data points

Questions?