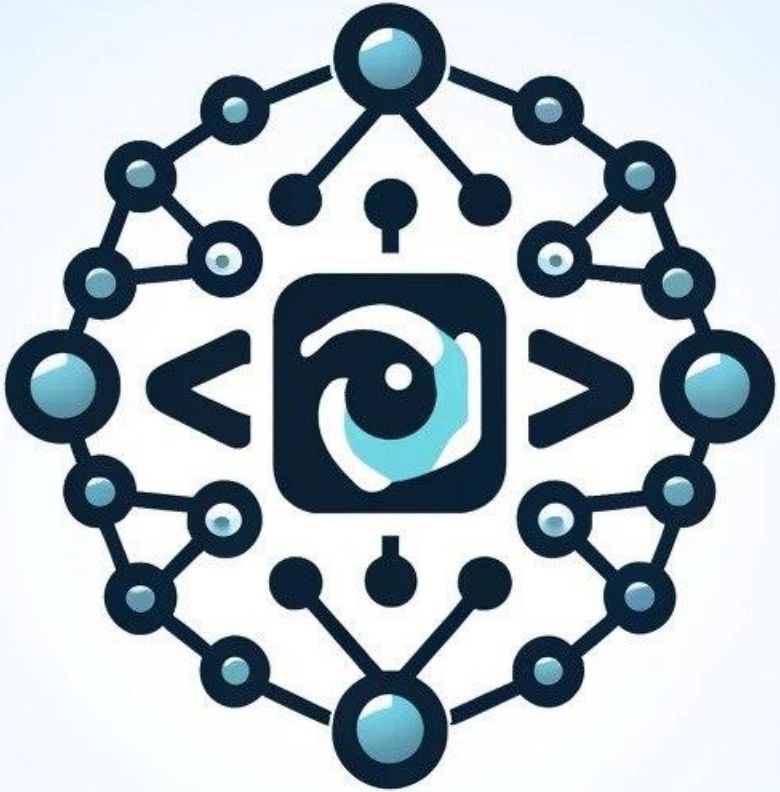
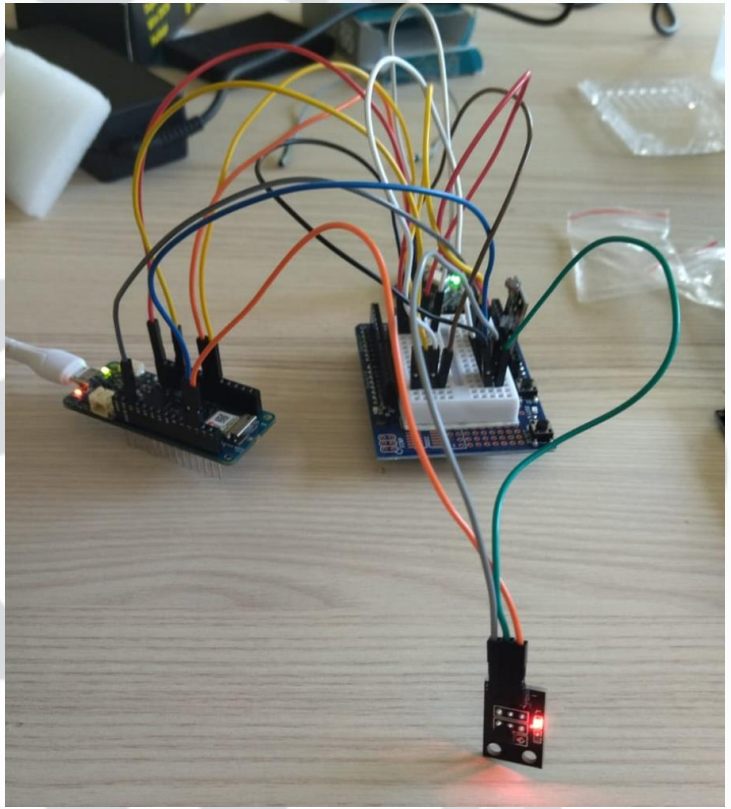


Sensing, Adapting, and Predicting

AdaptoSense

Project idea

AdaptoSense is an ANN-based system that predicts missing sensor data (temperature, humidity, CO2, light) in a room using internal and external weather data and trained on data collected during a successful implementation week.



Auditorium - the project location

- Located on the ground floor
- Spacious lecture room with two external wall, one glasswall with doors, one internal wall with huge, wall-to-wall windows

  → This unique layout influences the building energy processes

Software Used

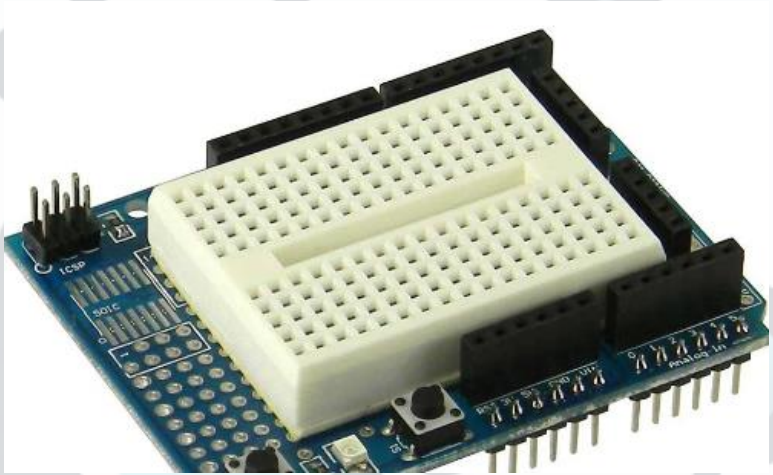
ARDUINO SCETCH IDE

PYTHON

EXCEL CSV FILE

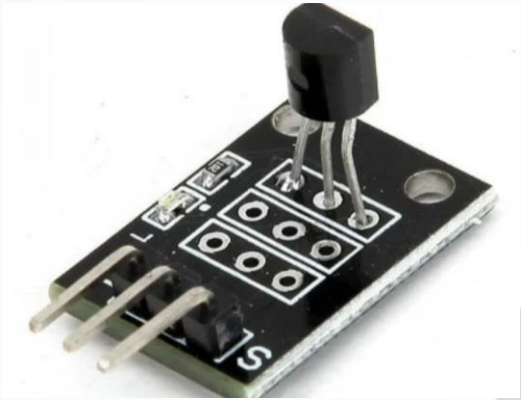
Arduino board and sensors

ARDUINO MKR WI-FI 1010



PROTOTYPE SHIELD

Arduino board and sensors



- **DIGITAL TEMPERATURE SENSOR**

Measure ambient temperature of the room.

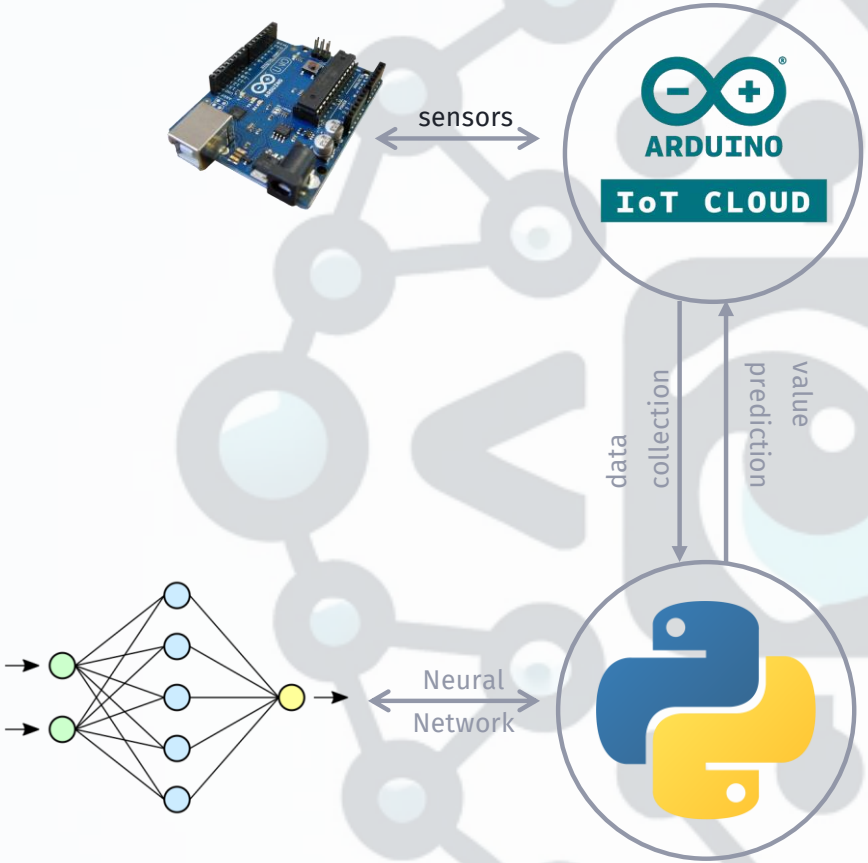
- **PHOTORESISTOR SENSOR**

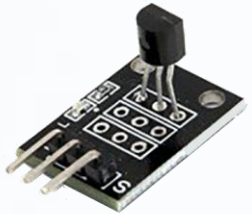
Measure ambient brightness.

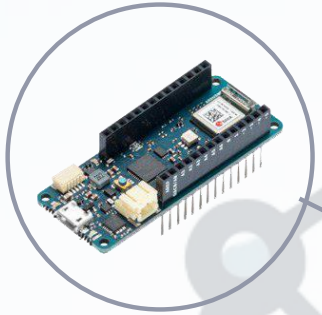
- **CO2 AND HUMIDITY SENSOR**

Measure the concentration of carbon dioxide in the air and the humidity of the air.

Process Flow

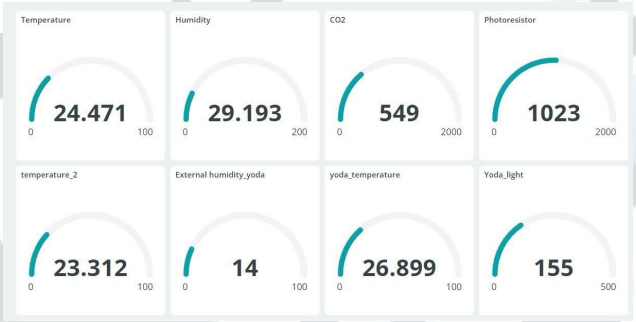




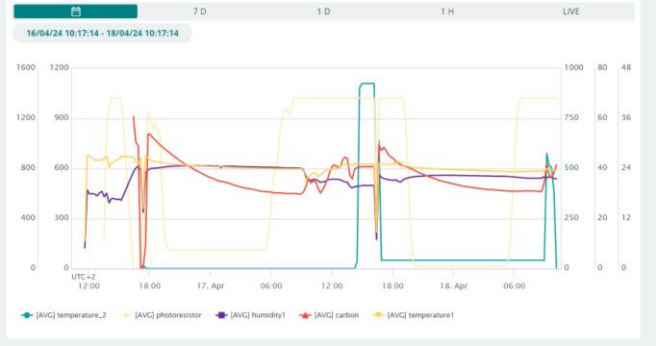


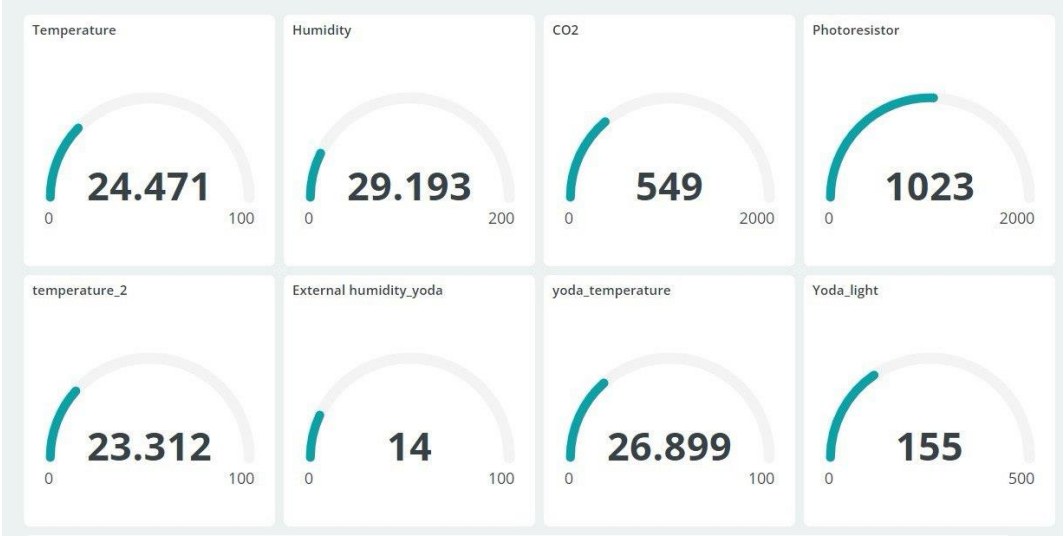
ARDUINO
IoT CLOUD

INTERFACE



Arduino Cloud Interface





Advantages of prediction of sensor values

CONTINIOUS OPERATION

In many industrial processes and environment, the uninterrupted operation of equipment is a key point. Systems can continue working based on predicted temperature and humidity data until the sensoring is normalized again.

DATA INTEGRITY

Gaps in data lead to uncertainties and inaccuracies in many processes, like analysis and decision-making. By applying the sensor-driven prediction, data collection is continious, ensuring the integrity of the overall data.

COST EFFICIENCY

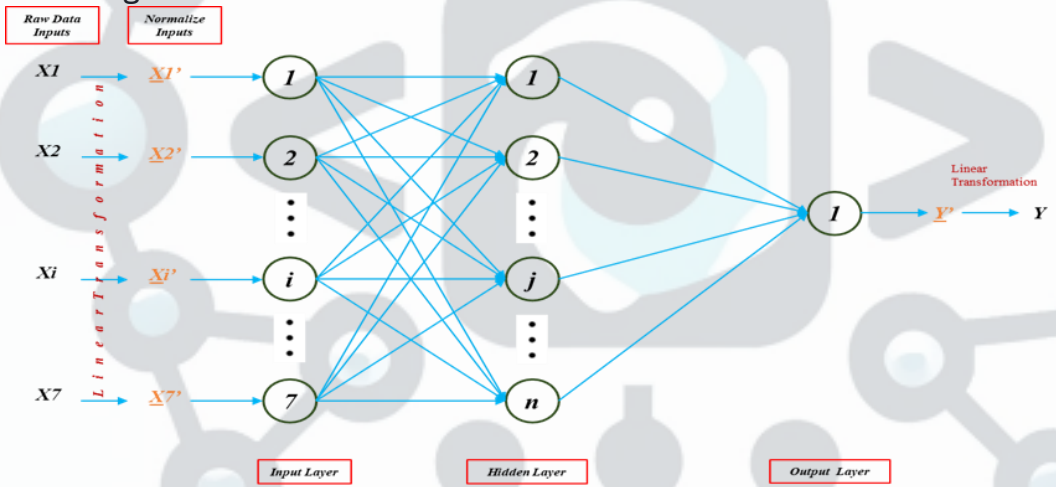
Instead of immidiately replacing and repairing meters a system can operate based on predited values until a new metering device is installed.

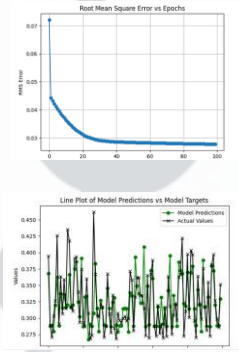
AdaptoSense: NN Architecture

The ANN model consist of

1. **An Input Layer** (7 Nodes): can vary as per requirement
2. **A Hidden Layer** (Nodes =70): Can be vary as per requirement
3. **An Output Layer** (Nodes = 1): Can be vary as per requirement
1. **No. of epoch:** 100 : can be changed

PREPROCESS DATA

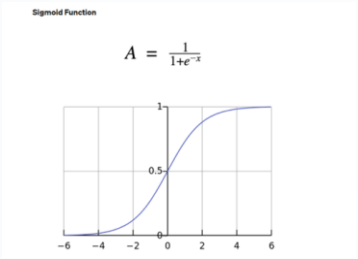




AdaptoSense: NN Architecture

- Any missing parameters can be represented as a mathematical function dependent on other working seven variables:
- $y=f(x)$
- Here, x is defined as:
- $x=[x1,x2...,x7]T$
- Error cost function = (target - predict)^2
- forward and back propagation algorithm with gradient descent methodology

sigmoid activation



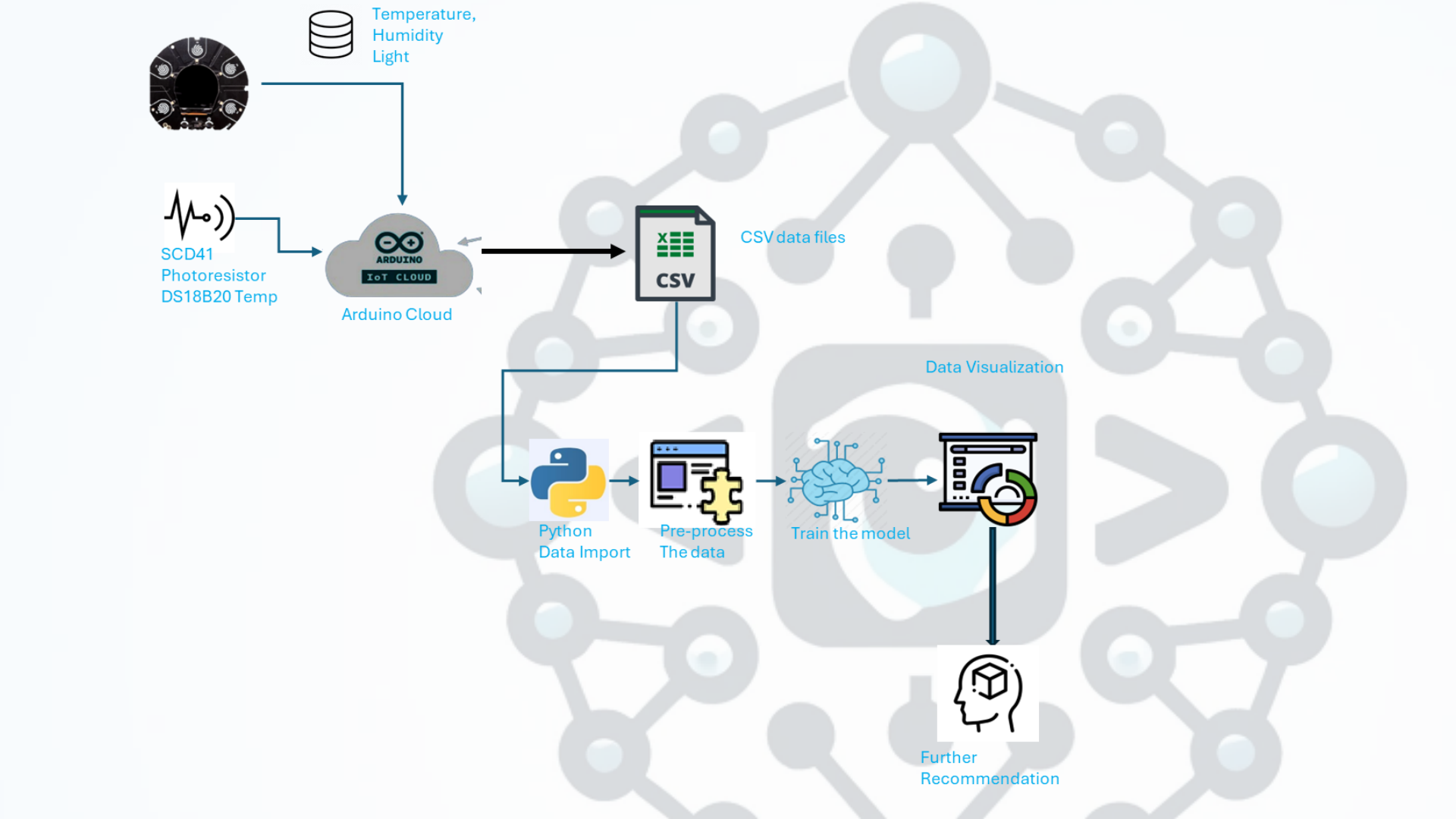
Performance Check

- Root Mean Square Error
- Graph Between Actual value and predicted value
- Equity Line

$$C.E. = 1 - E^2_{r.r.m.s.}$$

$$E_{r.r.m.s.} = \sqrt{\sum (y^{(p)} - y^{(a)})^2 / \sum (y^{(a)})^2}$$

AdaptoSense: Case Studies


Temperature,
Humidity
Light
SCD41
Photoresistor
DS18B20 Temp
ARDUINO
IoT CLOUD
Arduino Cloud
CSV
CSV data files
Data Visualization
Python
Data Import
Pre-process
The data
Train the model
Further
Recommendation

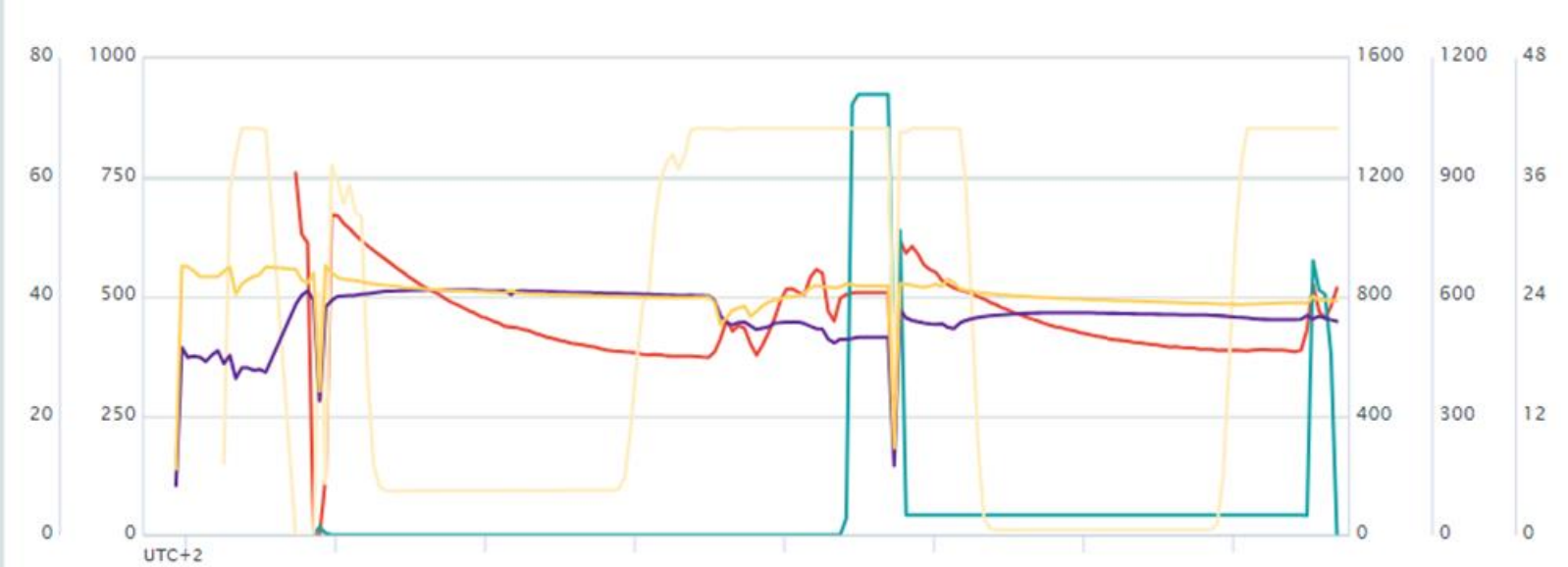
CO2 SENSOR

Photoresister SENSOR

Humidity SENSOR

Temperature SENSOR

Opla SENSOR



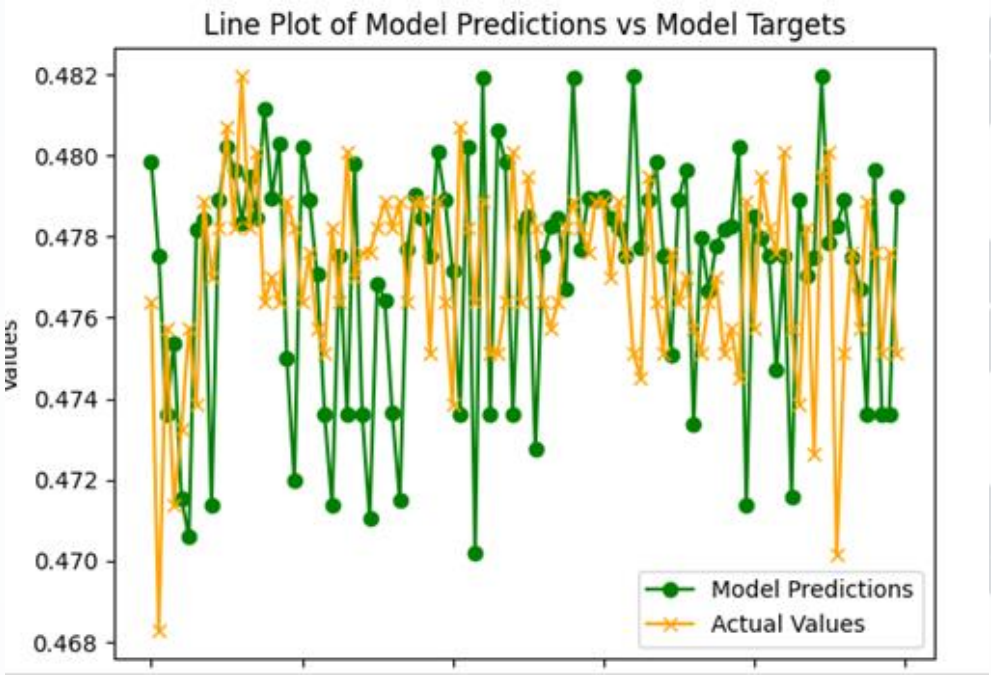
~~CO2 SENSOR~~

Photoresister SENSOR

Humidity SENSOR

Temperature SENSOR

Opla SENSOR



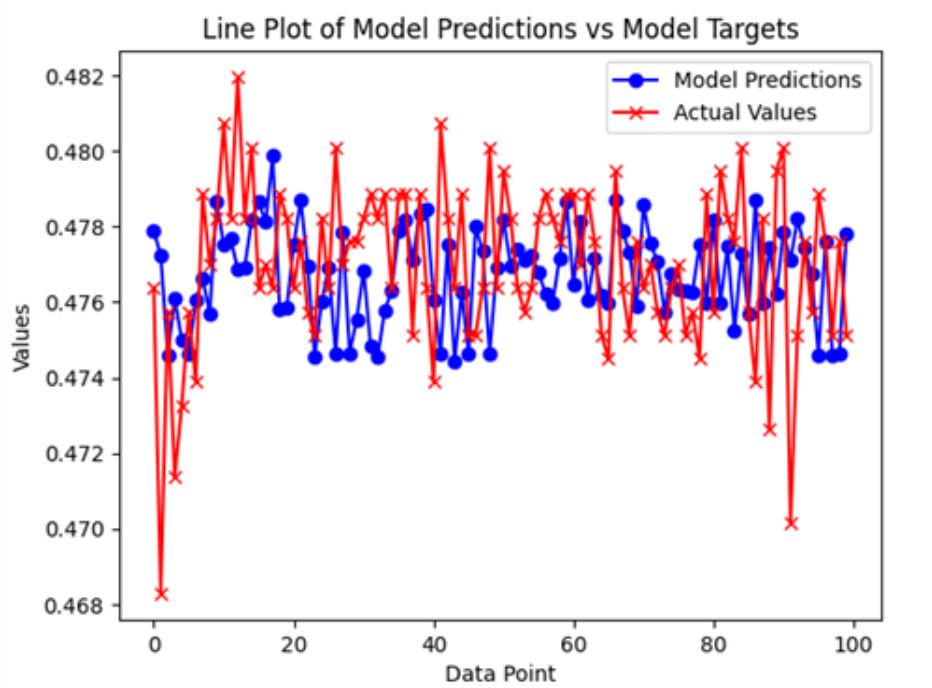
CO2 SENSOR

~~Photoresi~~ SENSOR

Humidity SENSOR

Tempera ture SENSOR

Opla SENSOR



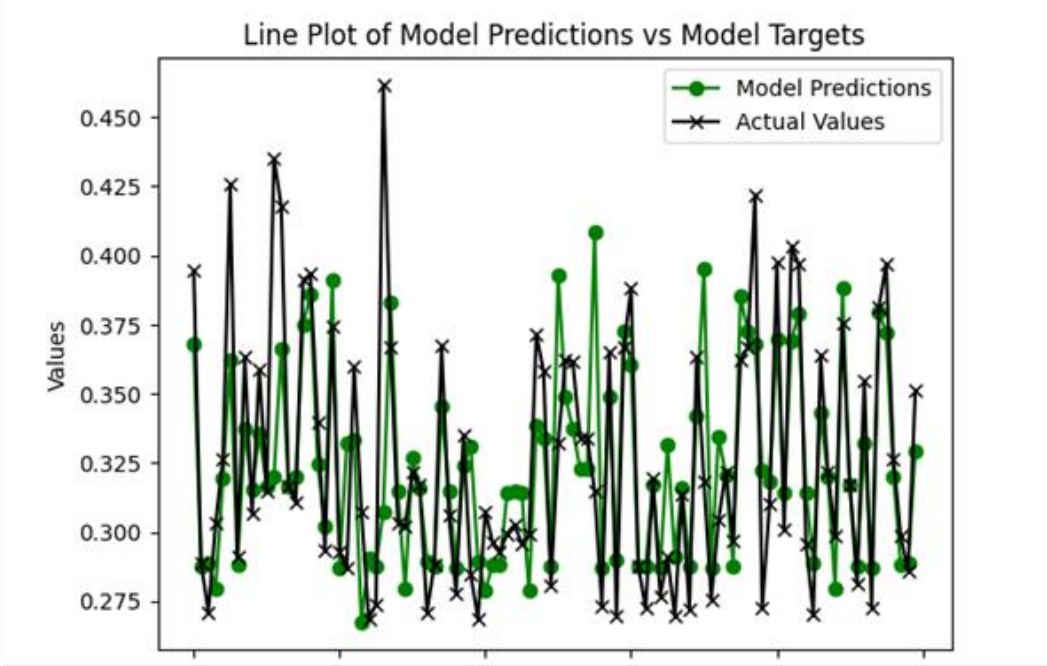
CO2 SENSOR

Photoresister SENSOR

~~Humidity SENSOR~~

Temperature SENSOR

Opla SENSOR



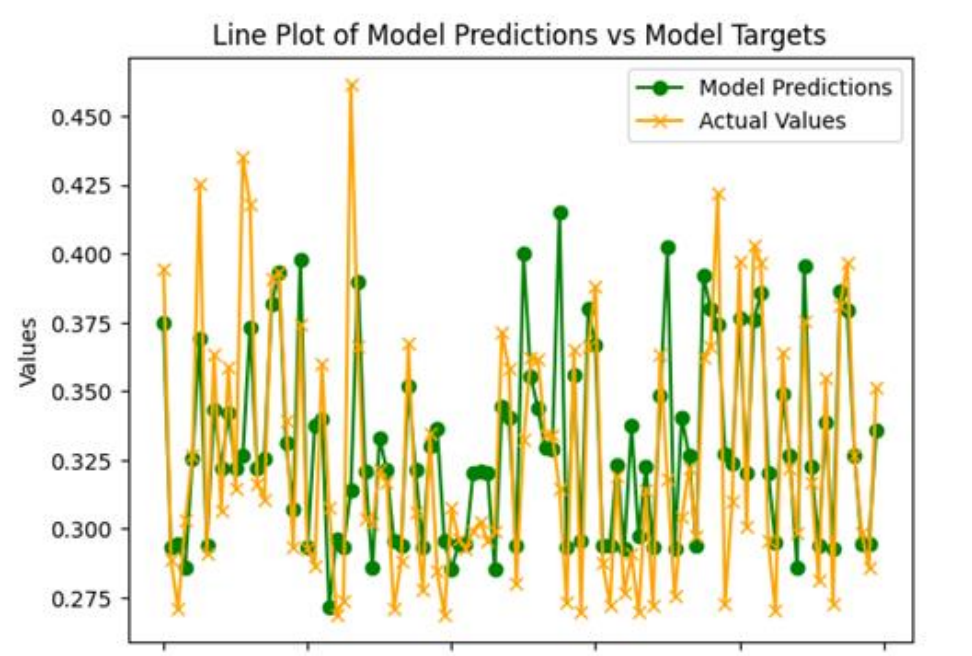
CO2 SENSOR

Photoresister SENSOR

Humidity SENSOR

Temperature SENSOR

Opla SENSOR



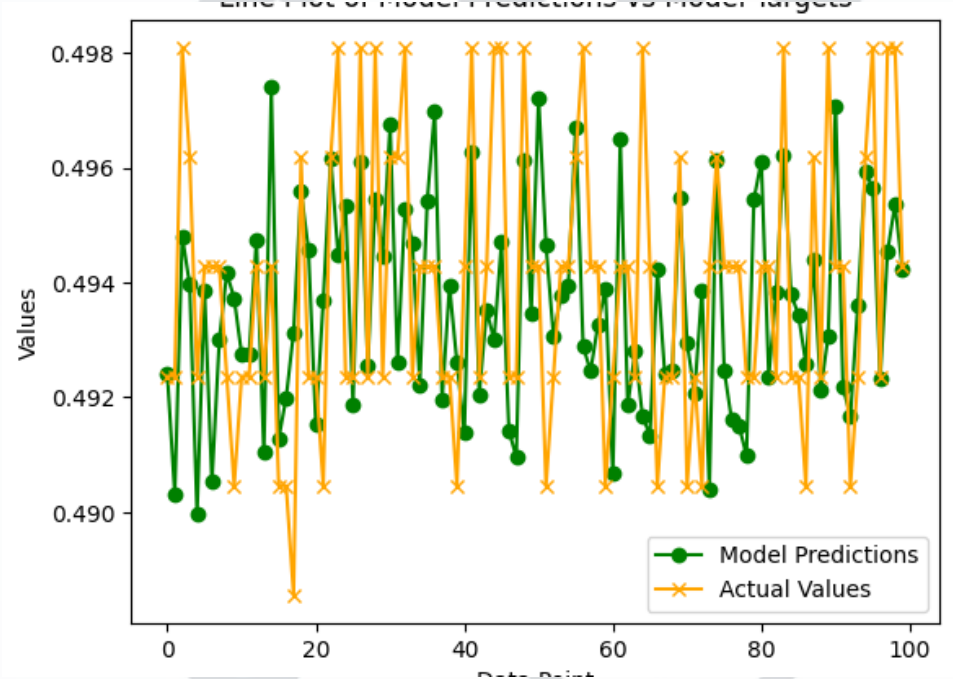
CO2 SENSOR

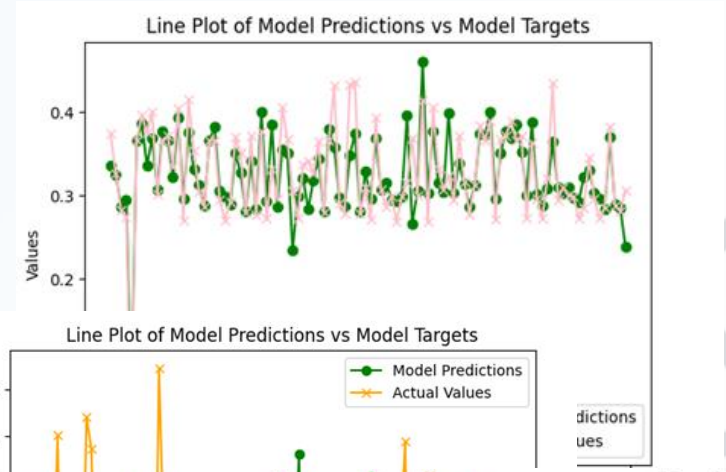
Photoresister SENSOR

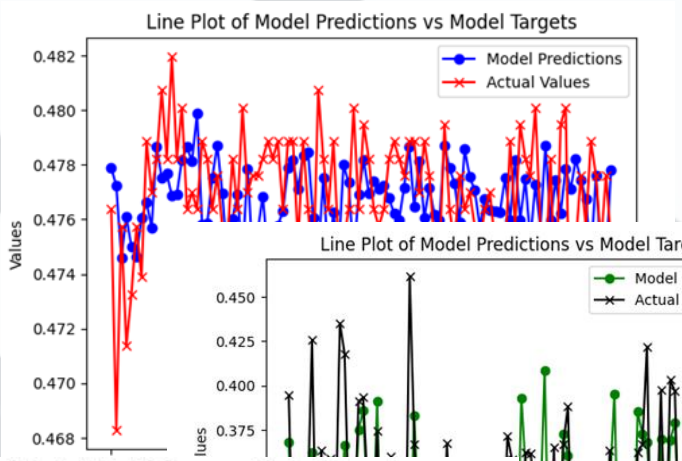
Humidity SENSOR

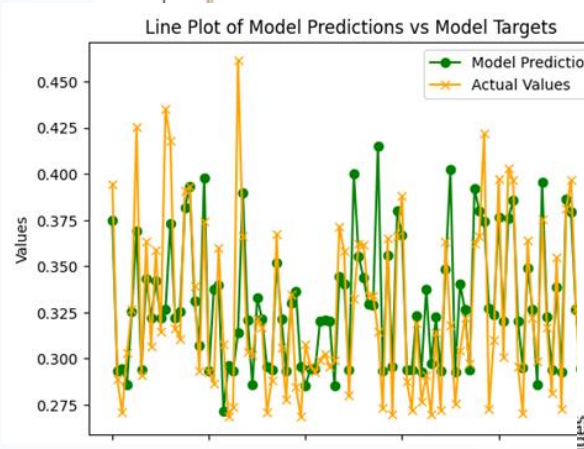
Temperature SENSOR

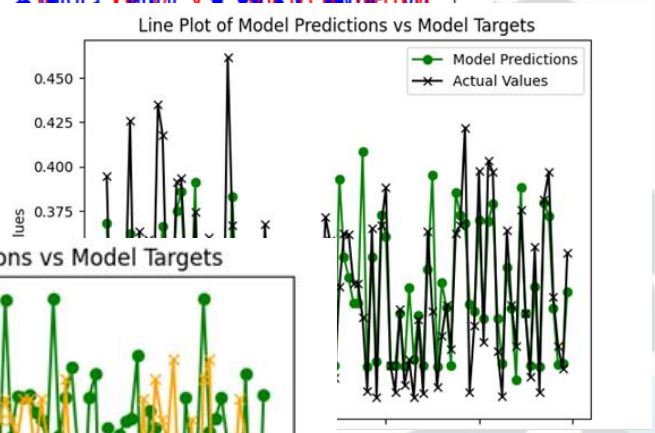
Opla SENSOR Temperature

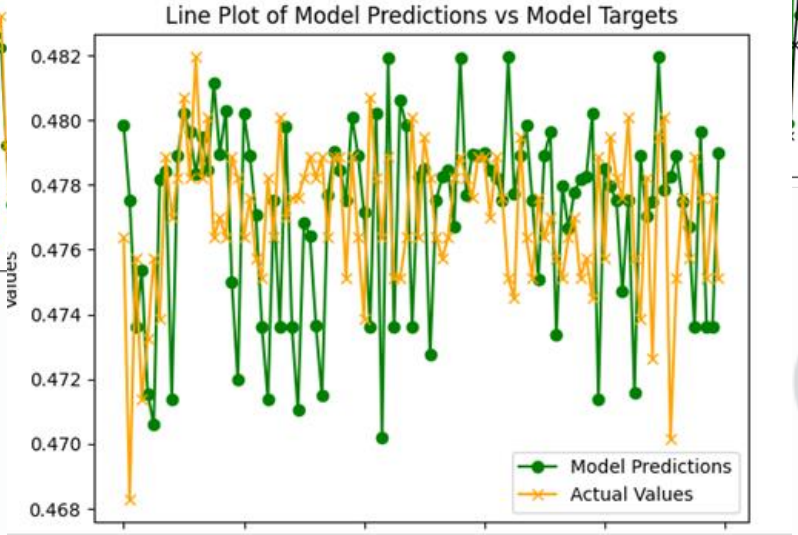



Line Plot of Model Predictions vs Model Targets
Values
0.4
0.3
0.2


Line Plot of Model Predictions vs Model Targets
Model Predictions
Actual Values
Values
0.482
0.480
0.478
0.476
0.474
0.472
0.470
0.468

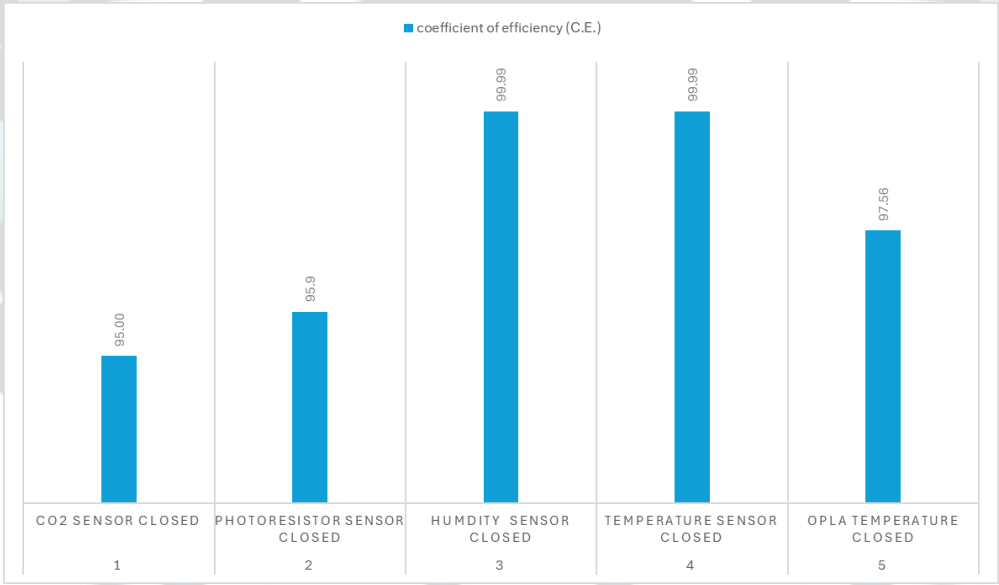

Line Plot of Model Predictions vs Model Targets
Model Predictions
Actual Values
Values
0.450
0.425
0.400
0.375
0.350
0.325
0.300
0.275


Line Plot of Model Predictions vs Model Targets
Model Predictions
Actual Values
0.450
0.425
0.400
0.375


Line Plot of Model Predictions vs Model Targets
Values
0.482
0.480
0.478
0.476
0.474
0.472
0.470
0.468
Model Predictions
Actual Values

Model Performance for All Scenario

Scenario	Description	coefficient of efficiency (C.E.)
1	CO2 Sensor Closed	95.00
2	Photoresistor Sensor closed	95.9
3	Humdity Sensor Closed	99.99
4	Temperature sensor closed	99.99
5	Opla temperature closed	97.56



Conclusions

ADEQUATE SOLUTION

- The outcome of the project was satisfying
- Neural network learned to make prediction with 99% correctness over time
- Sensor measurements were validated with other independent tools

WELL CHOSEN ENVIRONMENT

- The collaboration of Arduino Cloud and Pyhton code written Google Colab was successful
- Arduino Cloud handled connected sensors and board well
- Data measurement was uninterrupted
- Neural Network trained itself with greater accuracy over time

→ Data collection with Arduino sensors and their representation and storage in the cloud was successful

→ Data prediction based on previous values of the sensor was successful

Thank you for your attention!