WALC 2024
Applied AI

LSTM (Long Short-Term Model) at the Edge (& EI Python SDK)

Prof. Marcelo J. Rovai

rovai@unifei.edu.br

UNIFEI - Federal University of Itajuba, Brazil
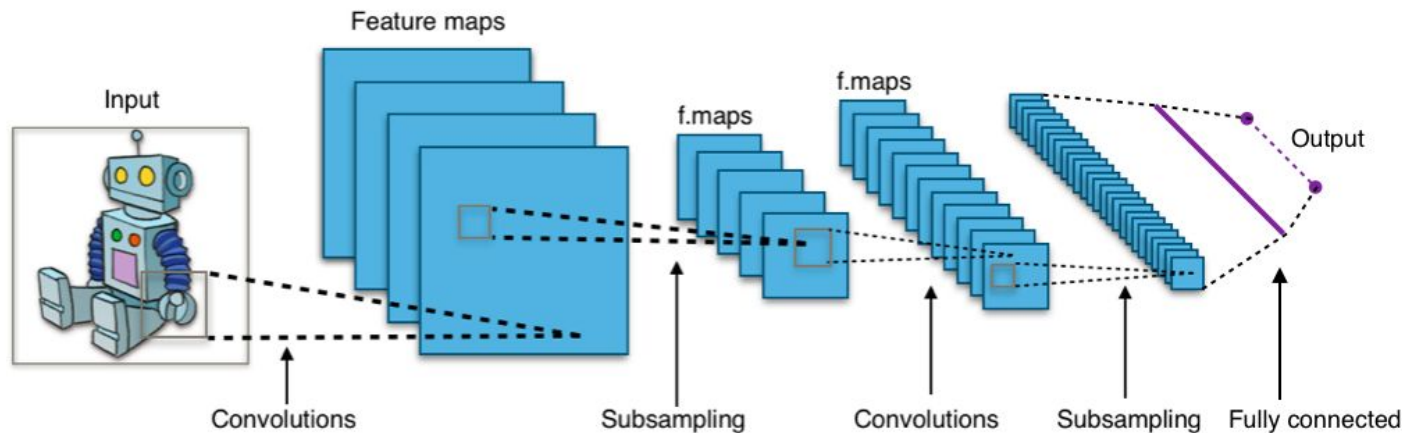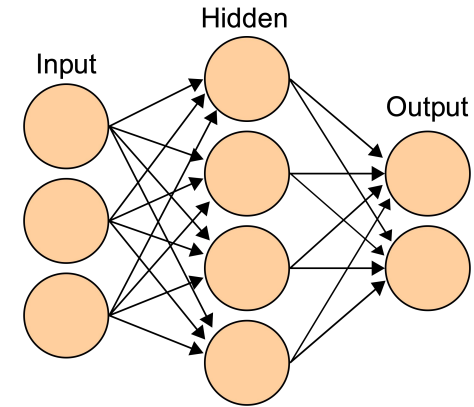TinyML4D Academic Network Co-Chair

TINYML4D

LSTM

Long Short-Term Model

*LSTM model is a type of recurrent neural network (RNN) that is well-suited for **sequence prediction** problems by effectively capturing long-term dependencies in data sequences.*

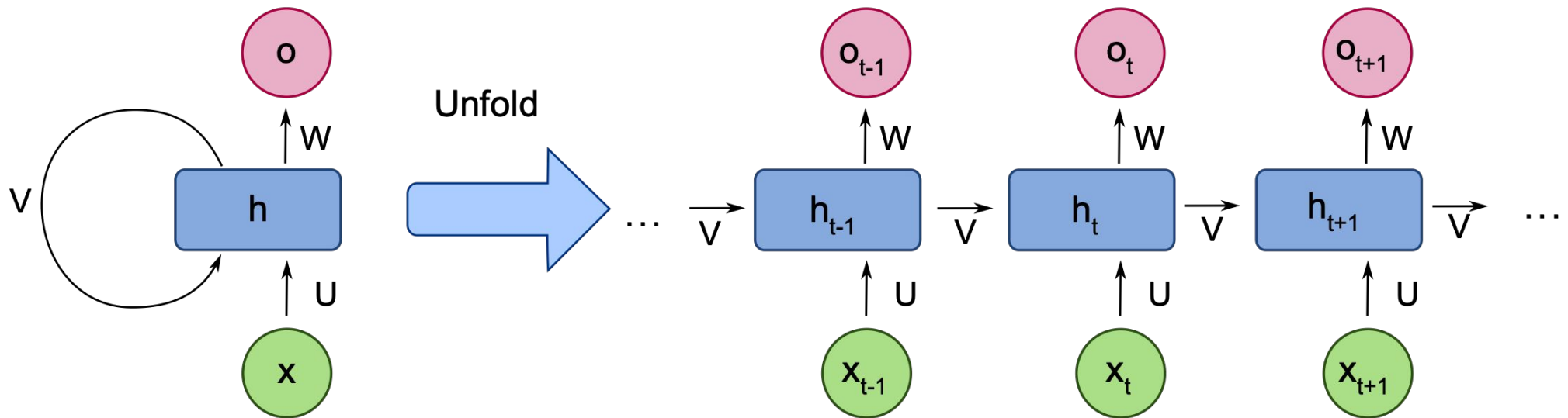# Deep Learning models (or artificial neural networks)

Fully Connected Neural Networks (FCNNs): Networks where **each neuron in one layer is connected to every neuron in the following layer**, useful for complex pattern recognition across diverse datasets.

Input Hidden Output

Feature maps

Input f.maps f.maps Output

Convolutions Subsampling Convolutions Subsampling Fully connected

Convolutional Neural Networks (CNNs): Specialized for **grid-like data such as images**, using convolutional layers to detect and learn spatial hierarchies of features.

# Deep Learning models (or artificial neural networks)

Recurrent Neural Networks (RNNs): Designed for **sequential data like time series or text**, these networks use their internal state (memory) to process sequences of inputs.

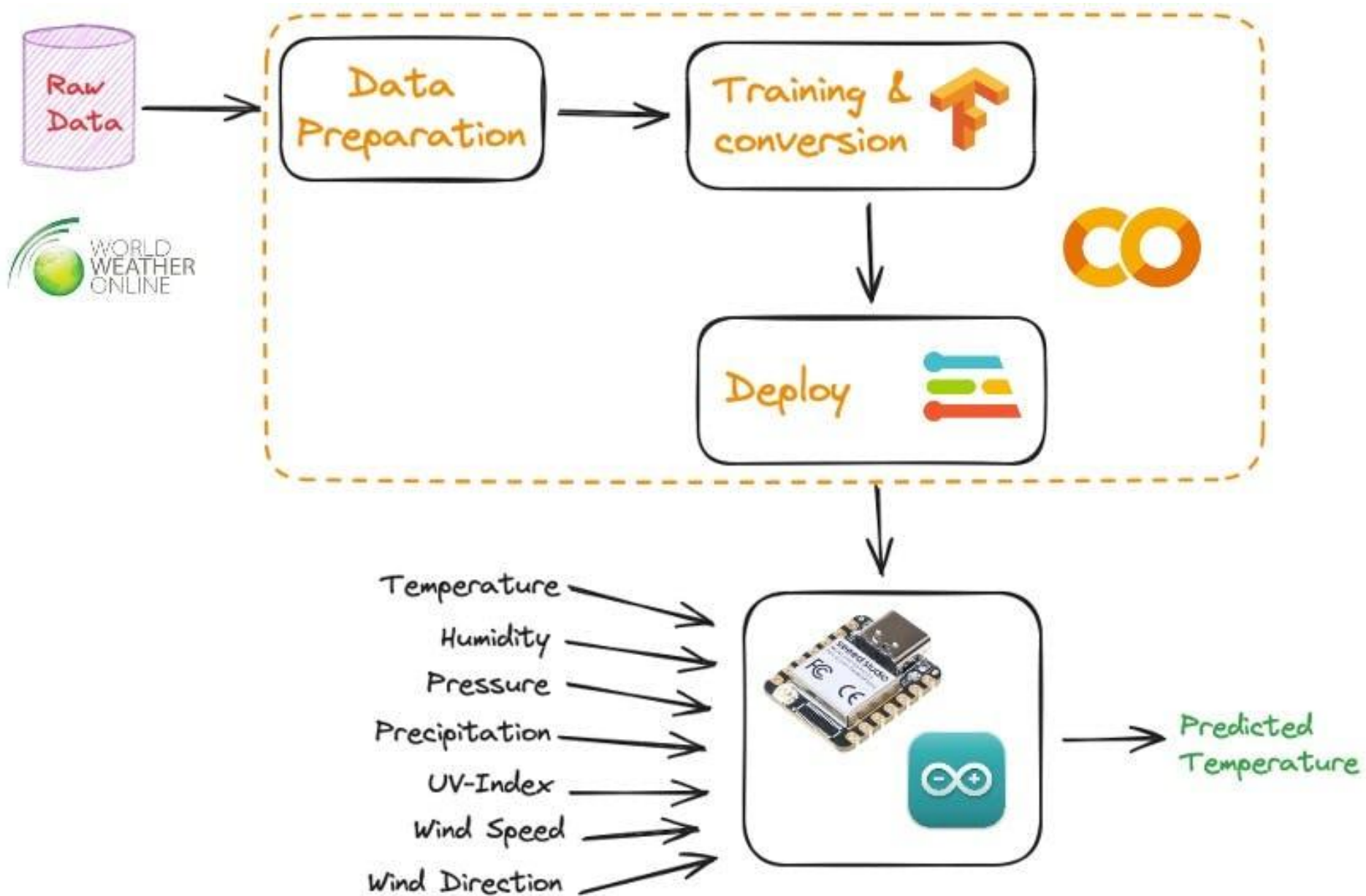# Deep Learning models (or artificial neural networks)

Long Short-Term Memory (LSTM): A type of RNN that can learn over long sequences without losing information, effectively managing long-term dependencies.
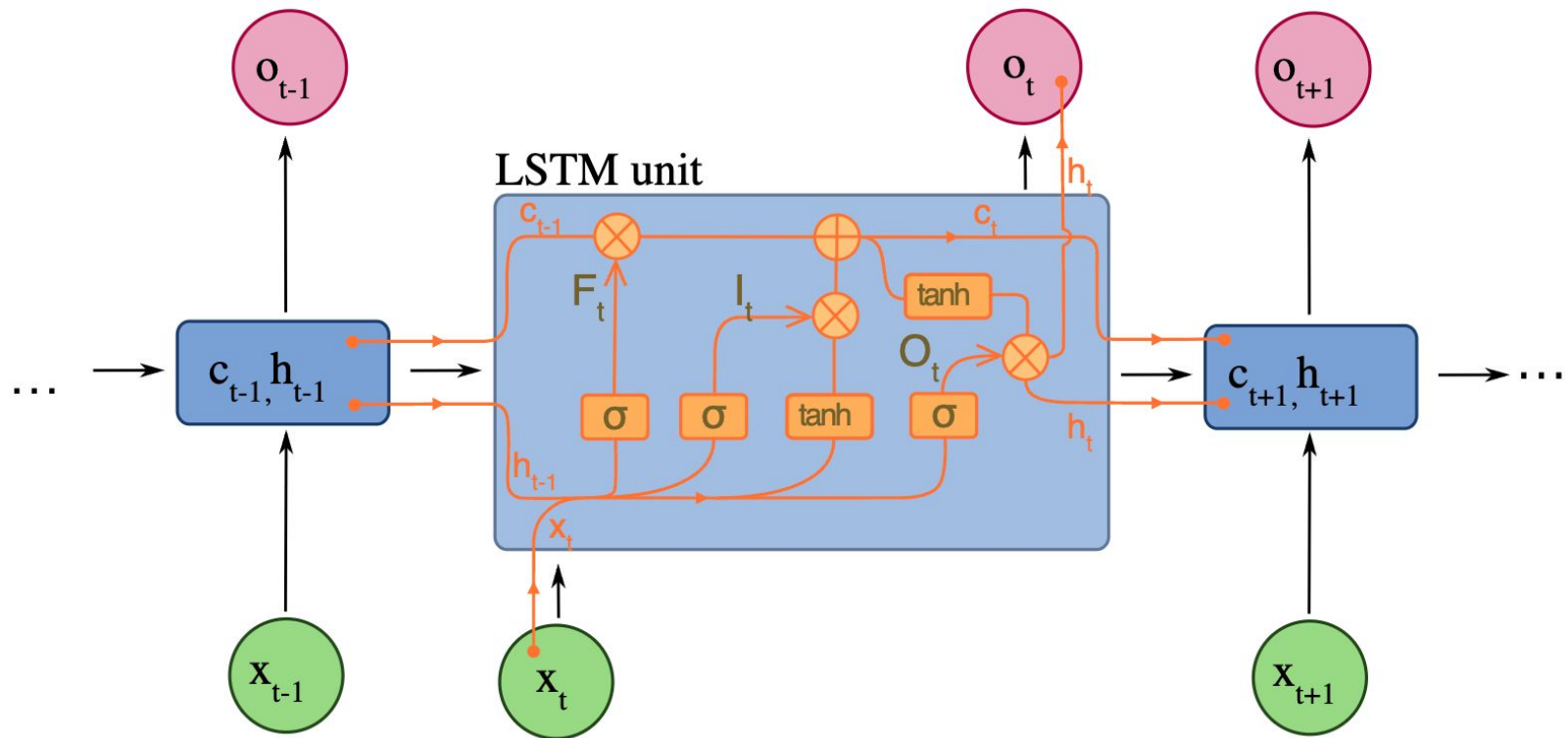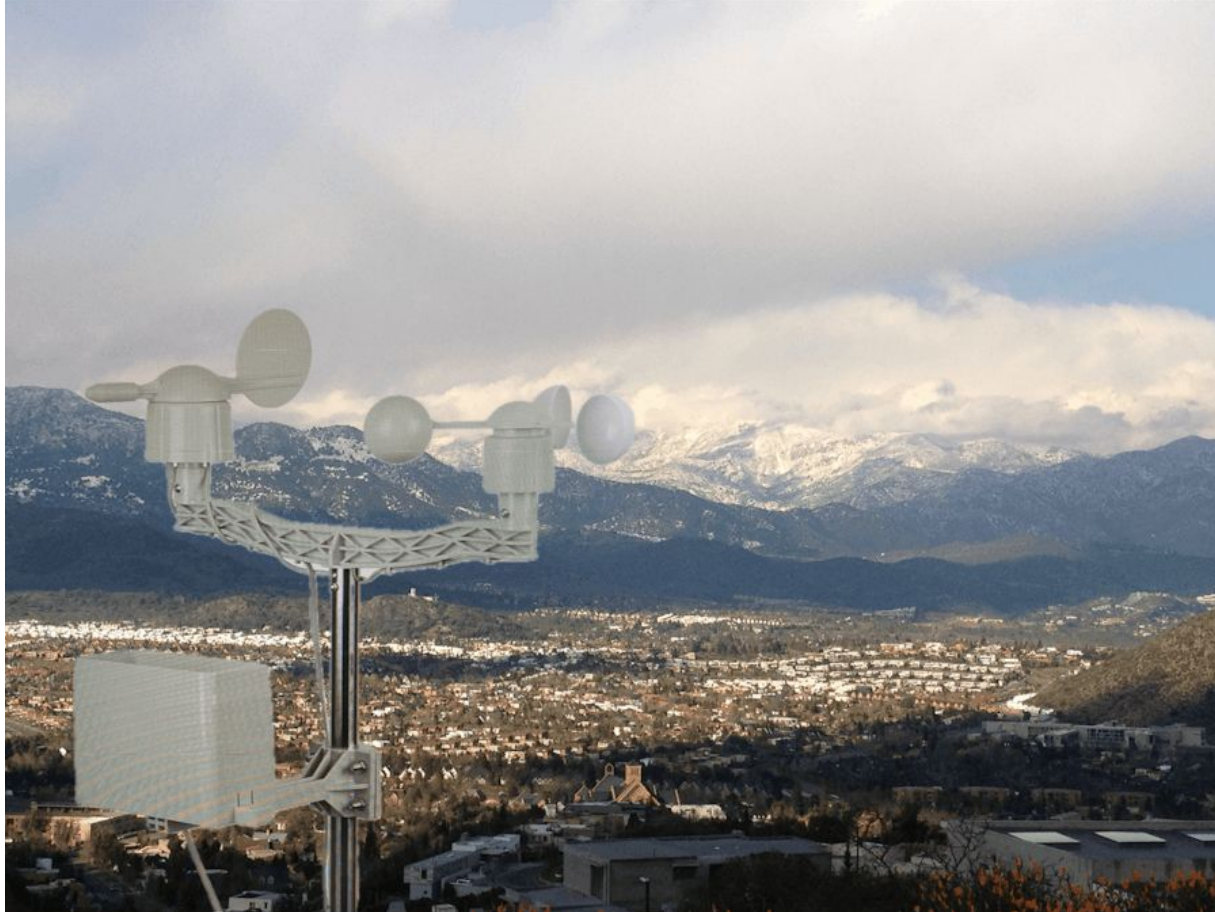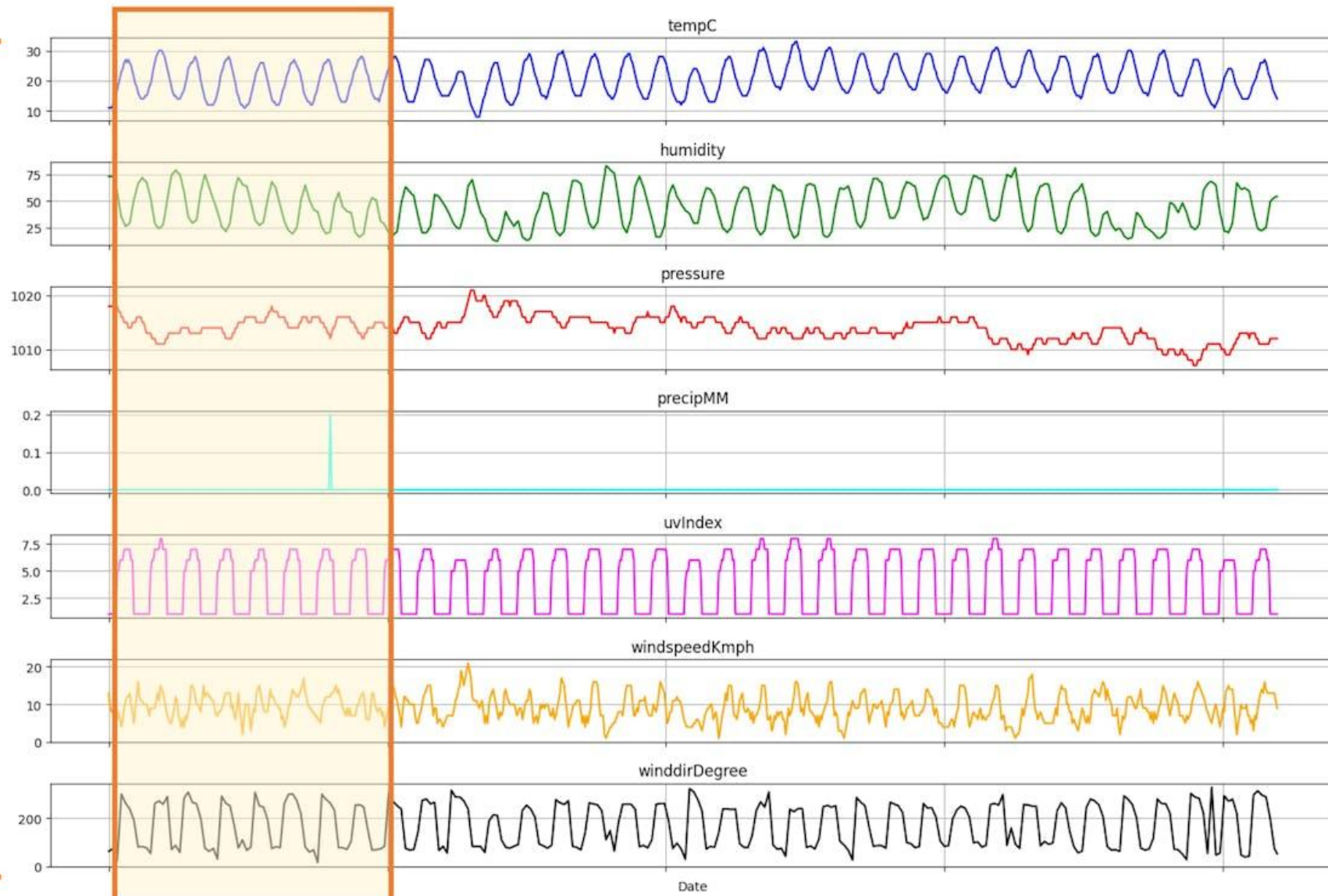
Temperature Prediction using an LSTM model

*The trained LSTM model will be converted with TFLite-Micro and **Edge Impulse Python SDK** and deployed on an XIAO ESP32S3.*
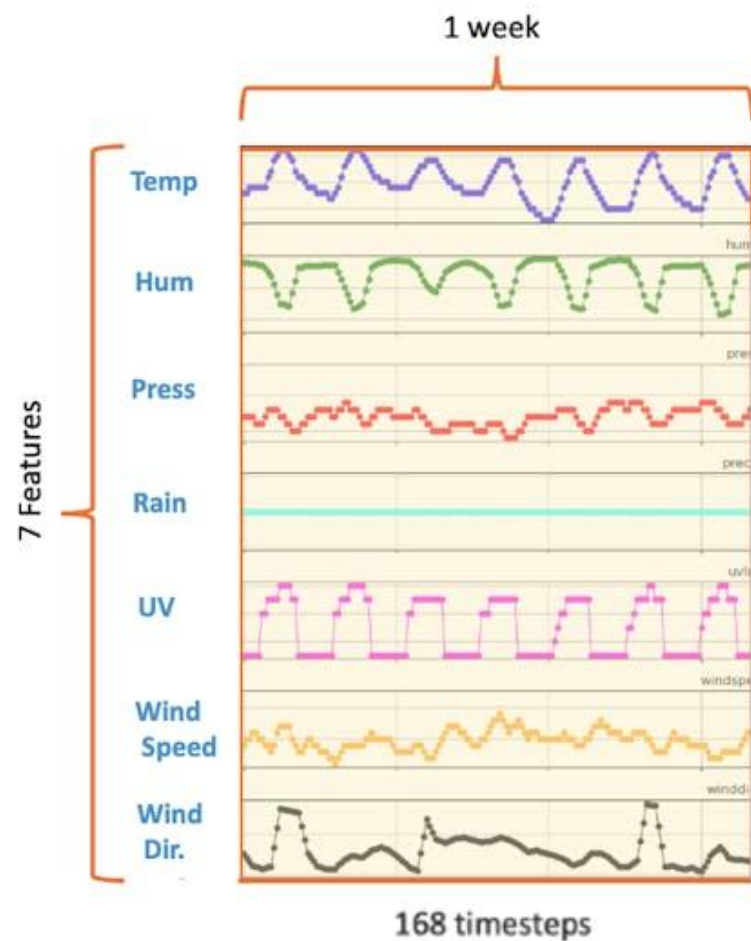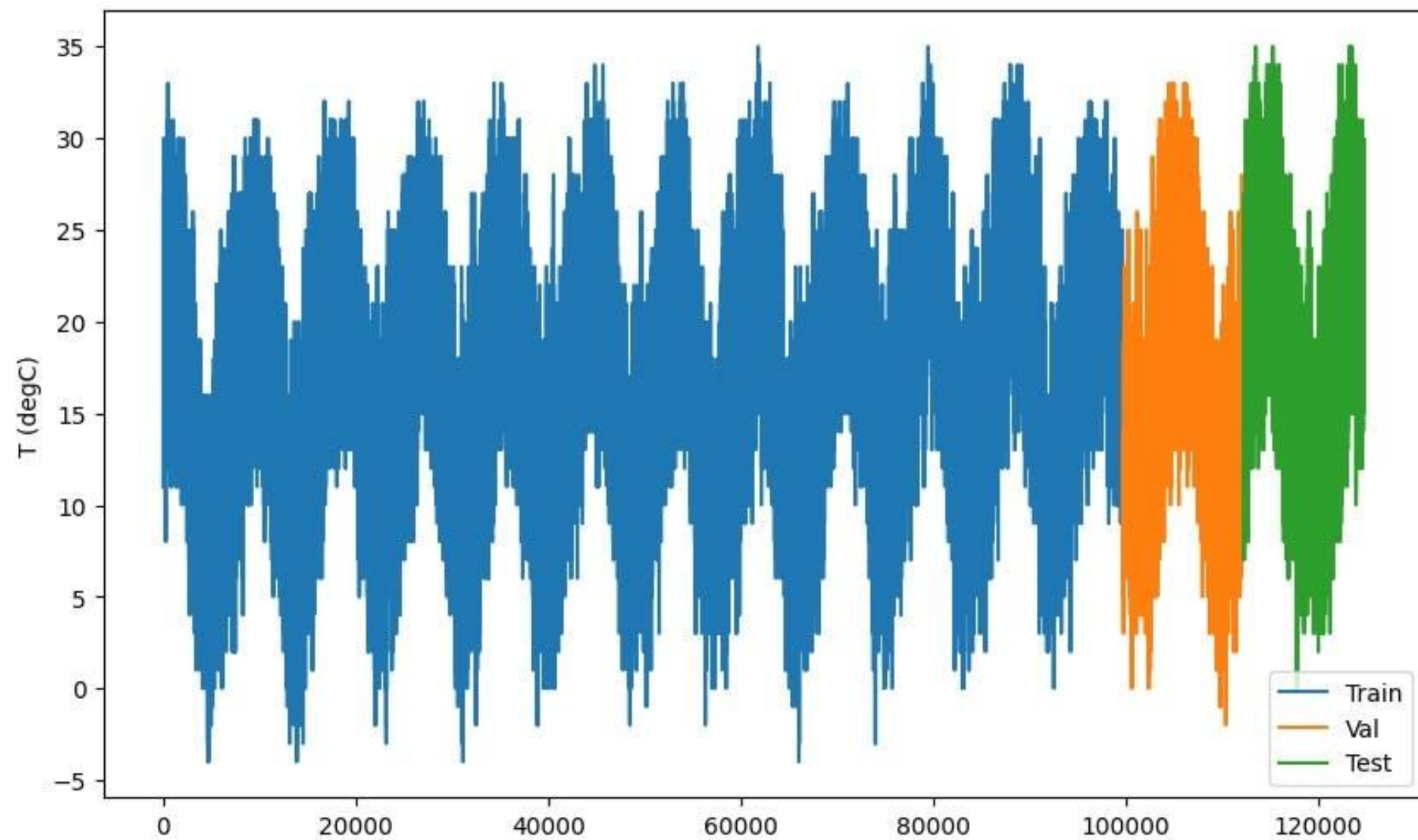
1 week

Temp

Hum

Press

Rain

UV

Wind Speed

Wind Dir.

7 Features

168 timesteps

LSTM Model

Prevision 6 hours in the future

Temperature

Temperature

```python
model = Sequential([
    LSTM(128,
         input_shape=(n_steps, X_train.shape[2])),
    Dense(1)
])
```

```python
model.compile(optimizer='adam', loss='mse')
```

```python
history = model.fit(
    X_train, y_train,
    validation_data=(X_val, y_val),
    epochs=20,
    batch_size=32,
    callbacks=[early_stopping]
)
```



```python
prediccion = model.predict(X_test)
```

```python
converter = tf.lite.TFLiteConverter.from_saved_model(MODEL_DIR)
tflite_model = converter.convert()

# Save the converted model to file
tflite_model_file = 'converted_model.tflite'
with open(tflite_model_file, 'wb') as f:
    f.write(tflite_model)
```
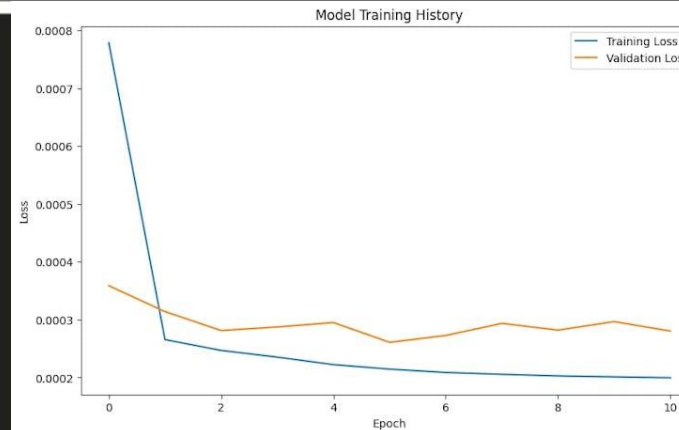
Time-Series Dataset

Feature Extraction

Model Training (TensorFlow)

Model Conversion (TFLite)

Edge Impulse Python SDK

profile()

```
target: cortex-m4f-80mhz
RAM: 39.1 kB
flash: 37.6 kB
latency: 145 ms
```
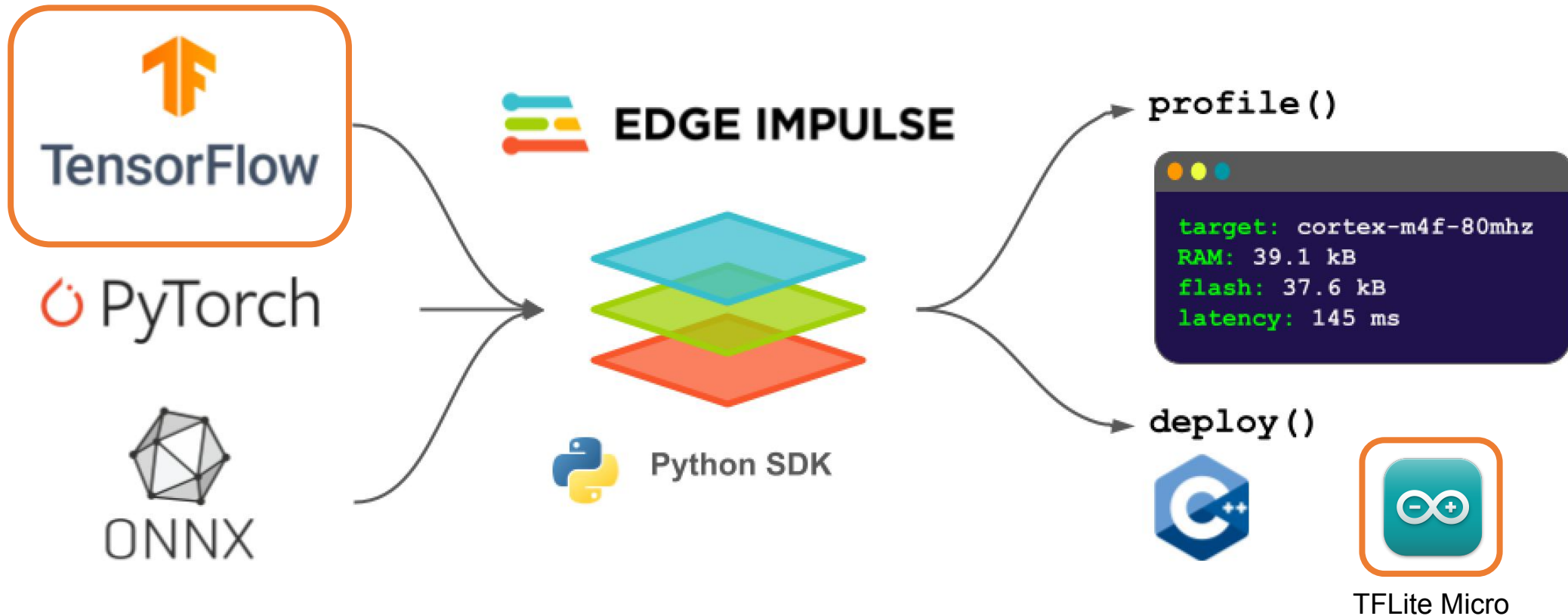
deploy()

TFLite Micro

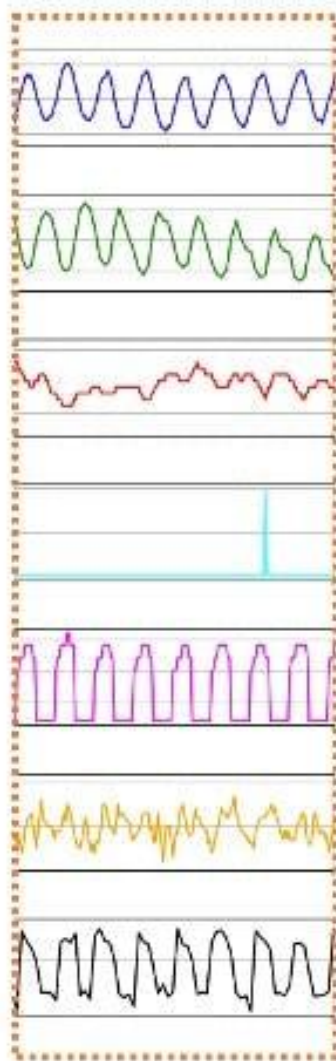converted_model.tflite ➡ EI Python SDK ➡ lstm_float32_model.zip

7 Features

168 timesteps

```
[0.38461538, 0.48979592, 0.4516129 , 0.        , 0.        ,
 0.04347826, 0.47777778, 0.38461538, 0.48979592, 0.4516129 ,
 0.        , 0.        , 0.        , 0.41666667, 0.41025641,
 0.5       , 0.4516129 , 0.        , 0.375     , 0.        ,
 0.35555556, 0.46153846, 0.43877551, 0.4516129 , 0.        ,
 0.375     , 0.04347826, 0.46666667, 0.51282051, 0.37755102,
 0.4516129 , 0.        , 0.5       , 0.13043478, 0.575     ,
 0.58974359, 0.31632653, 0.4516129 , 0.        , 0.5       ,
 0.17391304, 0.68611111, 0.61538462, 0.28571429, 0.4516129 ,
 0.        , 0.625     , 0.2173913 , 0.65277778, 0.66666667,
 0.26530612, 0.4516129 , 0.        , 0.625     , 0.30434783,
 0.61666667, 0.71794872, 0.24489796, 0.41935484, 0.        ,
 0.625     , 0.34782609, 0.58333333, 0.74358974, 0.2244898 ,
 0.41935484, 0.        , 0.625     , 0.47826087, 0.57777778,
 0.76923077, 0.21428571, 0.41935484, 0.        , 0.75      ,
 0.60869565, 0.575     , 0.74358974, 0.20408163, 0.38709677,
 0.        , 0.75      , 0.73913043, 0.57222222, 0.69230769,
 0.2244898 , 0.41935484, 0.        , 0.625     , 0.69565217,
```

1,176 Features

Edge Impulse Inferencing Demo
Edge Impulse standalone inferencing (Arduino)
run_classifier returned: 0
Timing: DSP 0 ms, inference 2024 ms, anomaly 0 ms
Predictions:
  value: 0.36065

static_buffer | Arduino IDE 2.3.2

XIAO_ESP32S3

static_buffer.ino

```
16
17   /* Includes ---------------------------------------------------------------- */
18   #include <LoBa_Temp_Prediction_-_LSTM_inferencing.h>
19
20   static const float features[] = {
21       0.38461538, 0.48979592, 0.4516129 , 0.        , 0.        ,
22           0.04347826, 0.47777778, 0.38461538, 0.48979592, 0.4516129 ,
23           0.        , 0.        , 0.        , 0.41666667, 0.41025641,
24           0.5       , 0.4516129 , 0.        , 0.375     , 0.        ,
25           0.35555556, 0.46153846, 0.43877551, 0.4516129 , 0.        ,
26           0.375     , 0.04347826, 0.46666667, 0.51282051, 0.37755102,
27           0.4516129 , 0.        , 0.5       , 0.13043478, 0.575     ,
28           0.58974359, 0.31632653, 0.4516129 , 0.        , 0.5       ,
29           0.17391304, 0.68611111, 0.61538462, 0.28571429, 0.4516129 ,
30           0.        , 0.625     , 0.2173913 , 0.65277778, 0.66666667
```

Output   Serial Monitor ✕

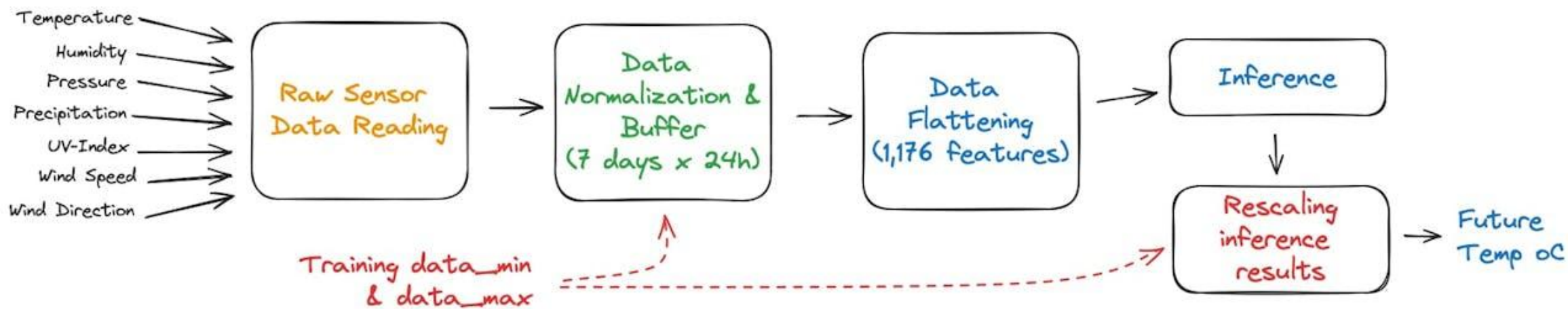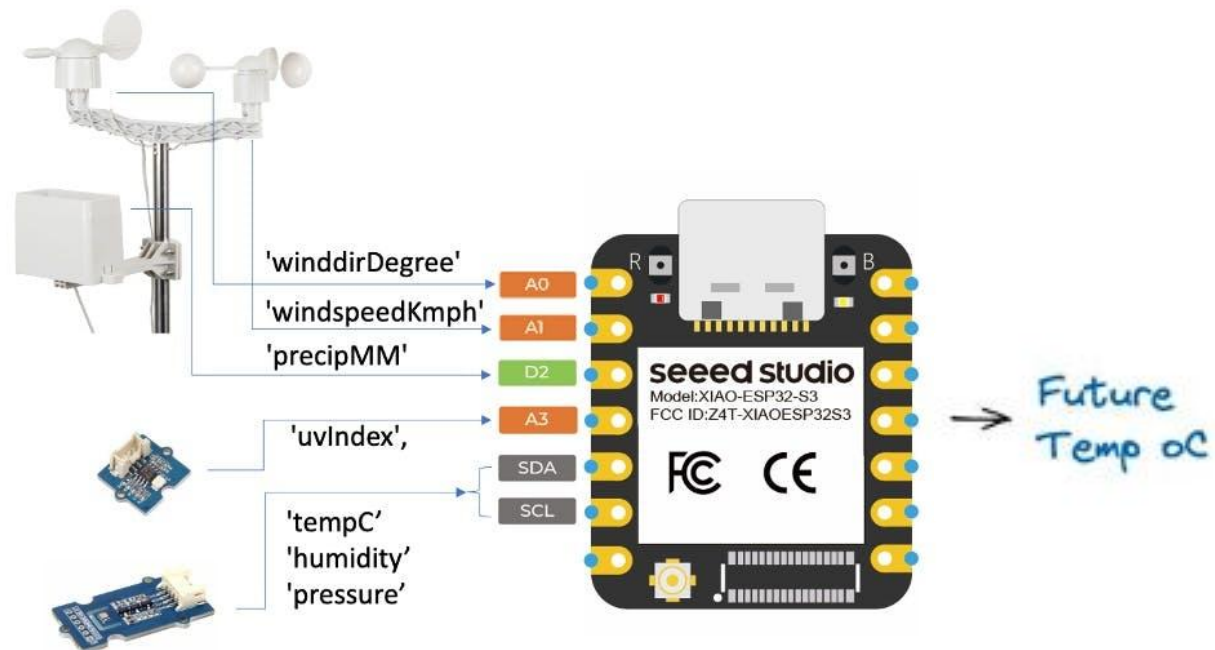Message (Enter to send message to 'XIAO_ESP32S3' on '/dev/cu.usbmodem101')   Both NL & CR   115200 baud

```
Edge Impulse Inferencing Demo
Edge Impulse standalone inferencing (Arduino)
run_classifier returned: 0
Timing: DSP 0 ms, inference 2024 ms, anomaly 0 ms
Predictions:
  value: 0.36065
```

indexing: 14/49   Ln 256, Col 18   XIAO_ESP32S3 on /dev/cu.usbmodem101 ⟳ 2 ▤

'winddirDegree'
'windspeedKmph'
'precipMM'
'uvIndex',
'tempC'
'humidity'
'pressure'

Future Temp oC

Temperature
Humidity
Pressure
Precipitation
UV-Index
Wind Speed
Wind Direction

Raw Sensor Data Reading

Data Normalization & Buffer (7 days x 24h)

Data Flattening (1,176 features)

Inference

Rescaling inference results

Future Temp oC

Training data_min & data_max