Inference At the Edge:
A Case Study at the Amazon Spheres

Miro Enev, PhD
Sr. Solution Architect
NVIDIA

WenMing Ye
Specialist Solution Architect
Amazon

Introduction

Agenda

Introduction: AI @ Amazon Spheres

Video: Welcome to the Amazon Spheres [living wall video]

Approach:

Anomaly Detection using DL on Time-Series Sensor Streams

Architecture:

Training (Amazon SageMaker)

Inference (NVIDIA Jetson Xavier, Amazon SageMaker Neo)

Results:

Improved alerting

Future Work:

Computer vision based plant stress

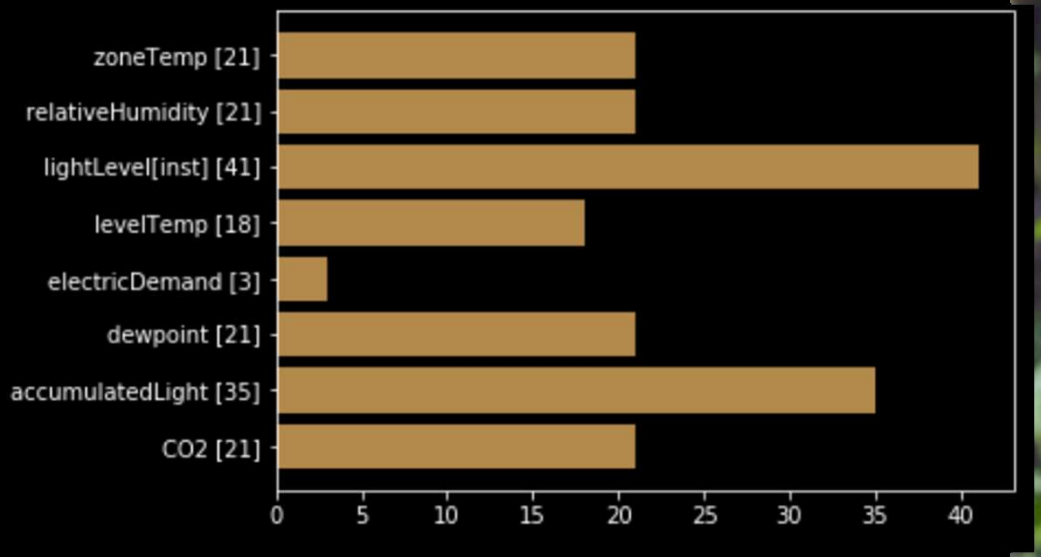
Our Goal = Help the Caretakers

Claire

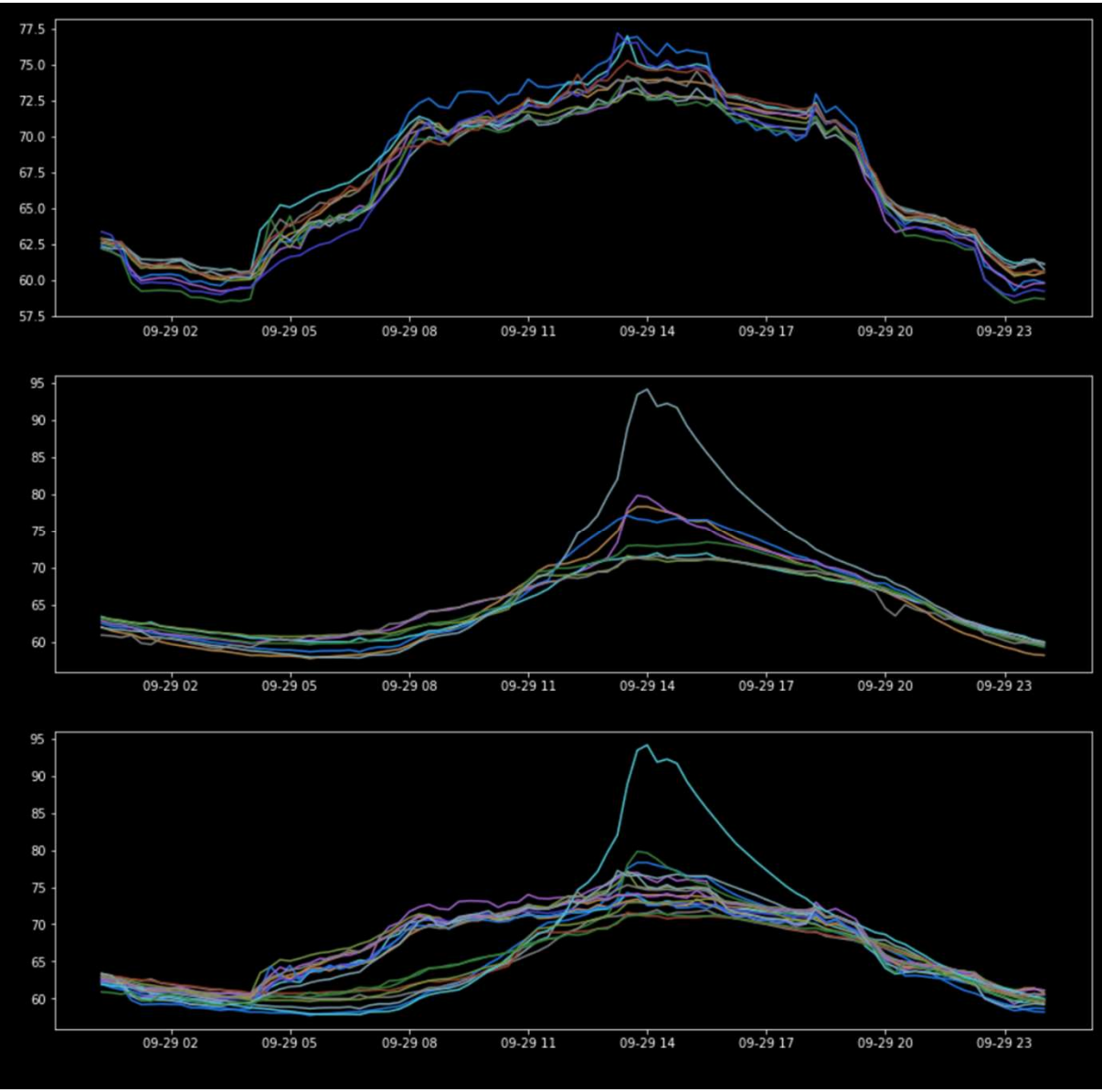
Ben

"We take care of **40,000 plants** from over **700 species**!"

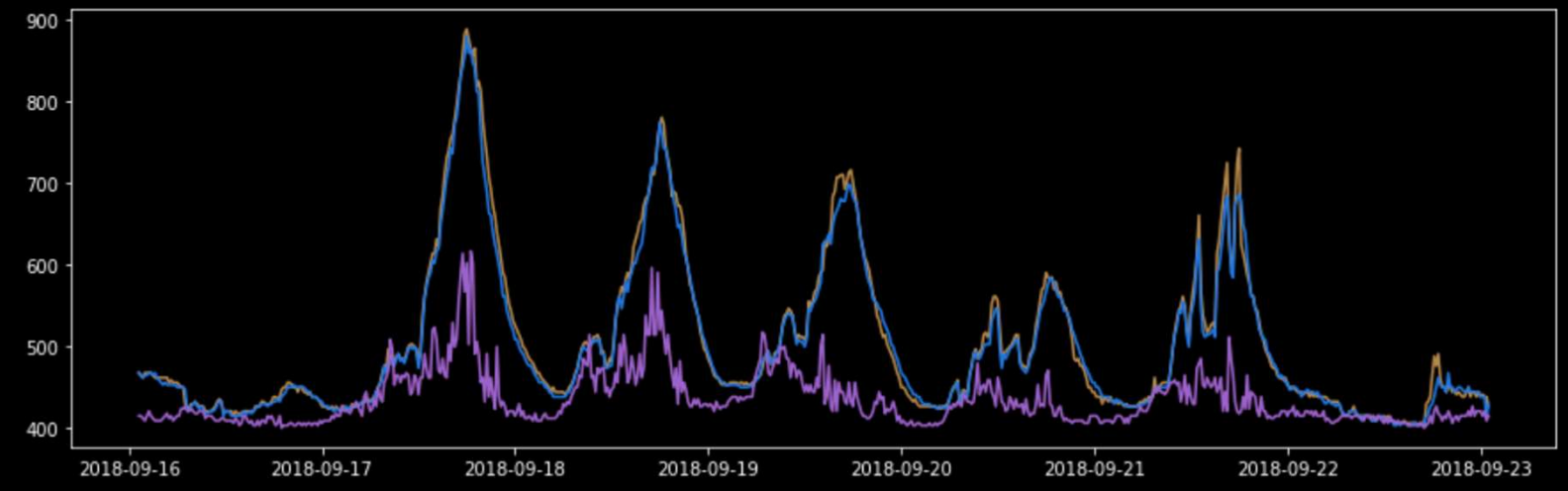
Sensor Types

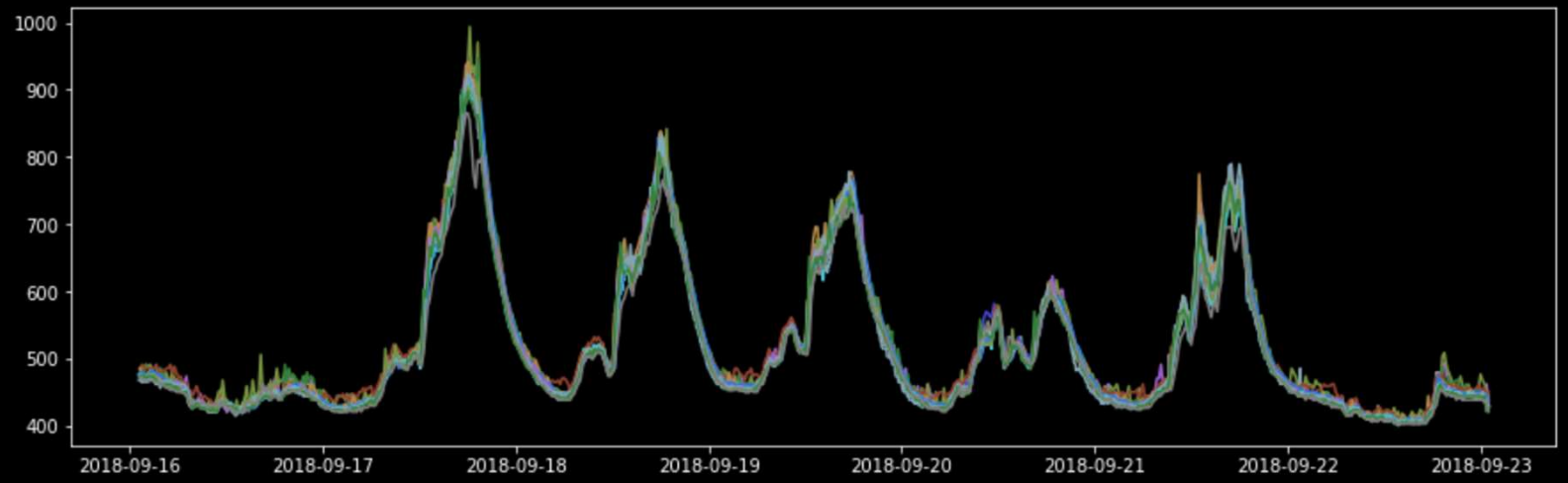


Temperature

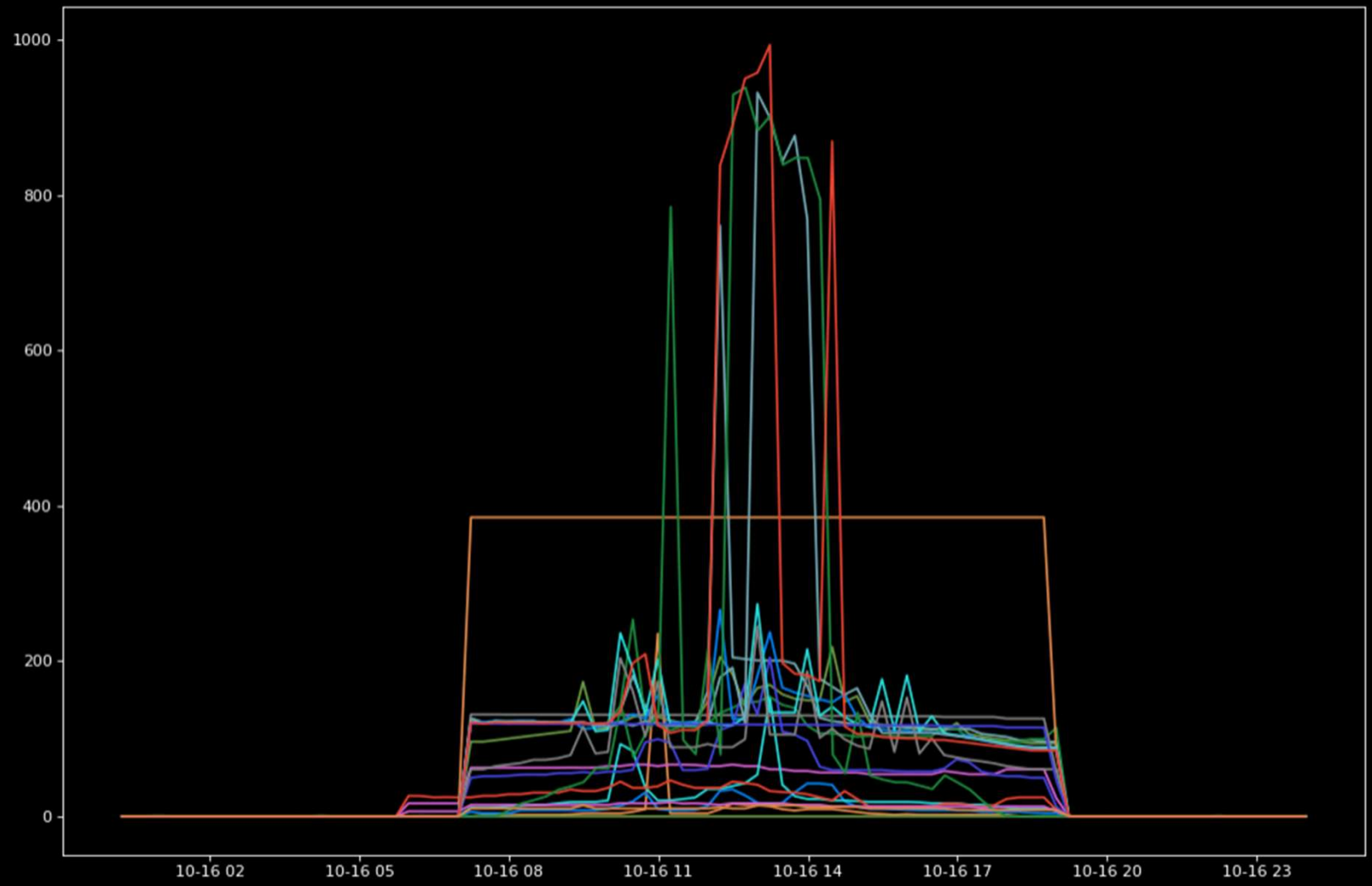


Co2

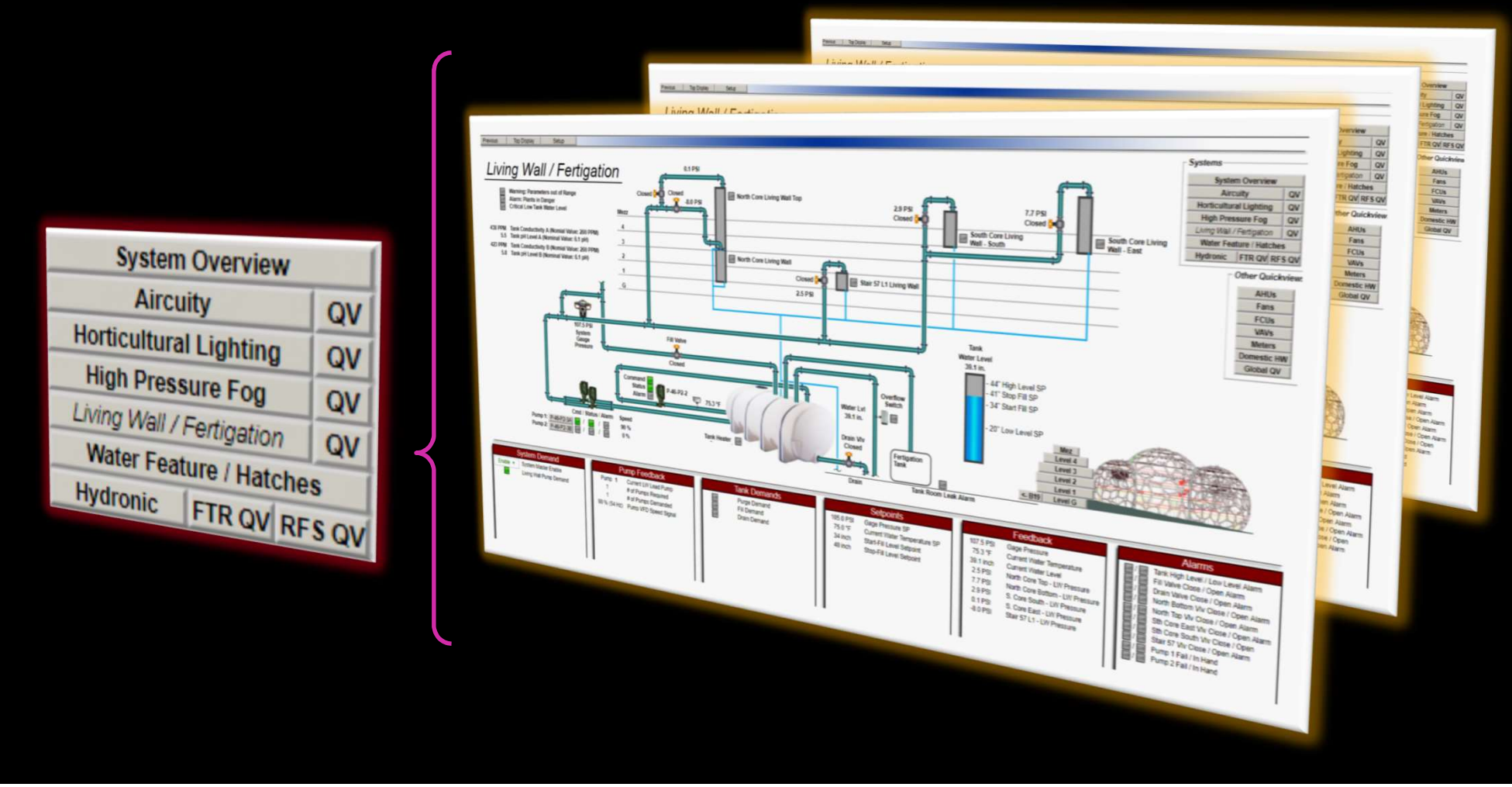

900
800
700
600
500
400
2018-09-16
2018-09-17
2018-09-18
2018-09-19
2018-09-20
2018-09-21
2018-09-22
2018-09-23


1000
900
800
700
600
500
400
2018-09-16
2018-09-17
2018-09-18
2018-09-19
2018-09-20
2018-09-21
2018-09-22
2018-09-23

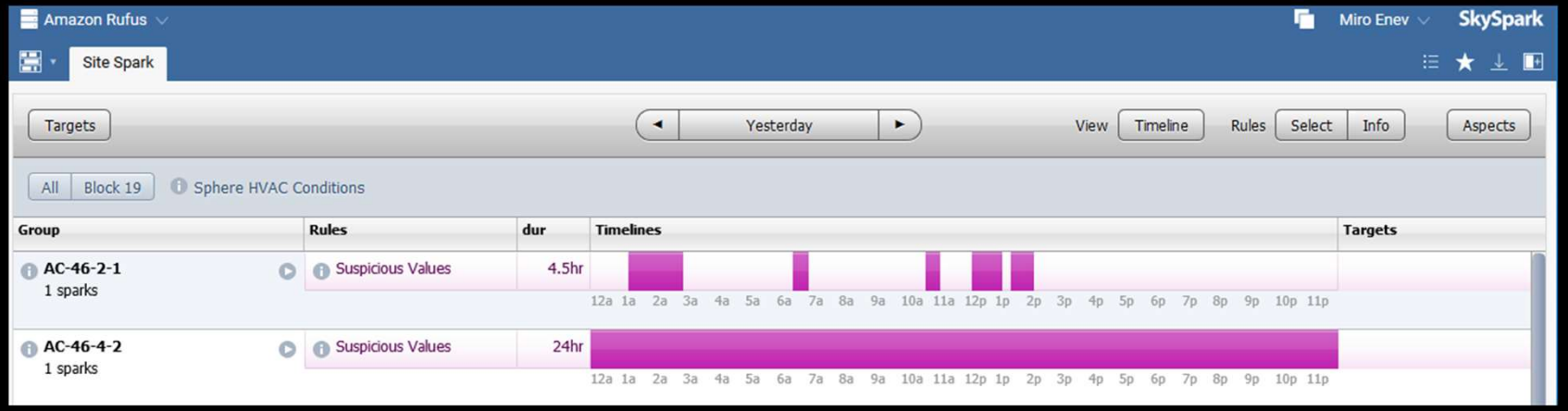
Inst Light Levels



Challenge 1: Lots of Systems to Manage



Challenge 2: Too Many Suspicious Values



flt_suspiciousValue

```
flt_suspiciousValue(thePoint, sparkDate, duration: 30min, negativeThreshold: 0, airTempHighLimit:
140, waterTempHighLimit: 180, steamTempHighLimit: 250)
```

This point has data values that are less than zero, suspicious temperature values if the point is a temperature sensor or null data values.

Parameters (Default):

- `duration (30min)`: The duration of the spark condition before firing a spark.
- `negativeThreshold (0)`: The buffer of allowed negative numbers. Any negative numbers above this value are not analyzed. Used to disregard sensors that show slightly negative values when off.
- `airTempHighLimit (140)`: The maximum allowed air temperature. Any air temp value above this limit will produce a spark.
- `waterTempHighLimit (180)`: The maximum allowed water temperature. Any water temp value above this limit will produce a spark.
- `steamTempHighLimit (250)`: The maximum allowed steam temperature. Any steam temp value above this limit will produce a spark.

Pseudocode: Find periods of time a number point with history data, that is not calculated and not outside air temperature is null and/or less than 0 for 30 minutes. In addition, find periods of time when a temperature sensor returns a value higher than 140 for “air” tags, 180 for “water” tags, or 250 for “steam” tags for 30 minutes.

When Issues Occur, They Go Unnoticed

Example 1: During a product launch (Alexa microwave integration), event organizers requested that the temperature be lowered for media and the air velocity reduced for better acoustics.

Problem: Incorrect temp. and air velocity for 4th floor plants for a week

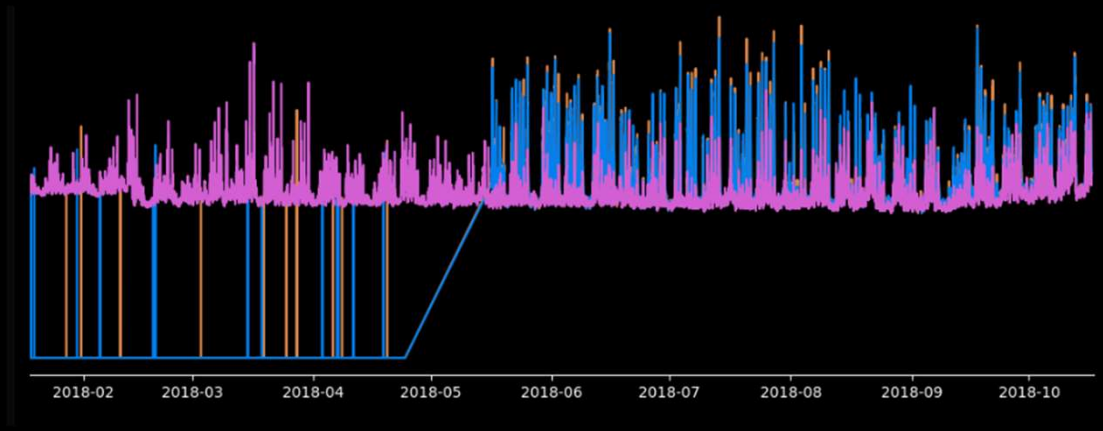
Example 2: Building automation staff suspended the irrigation for the living wall to update/repairs several sensors.

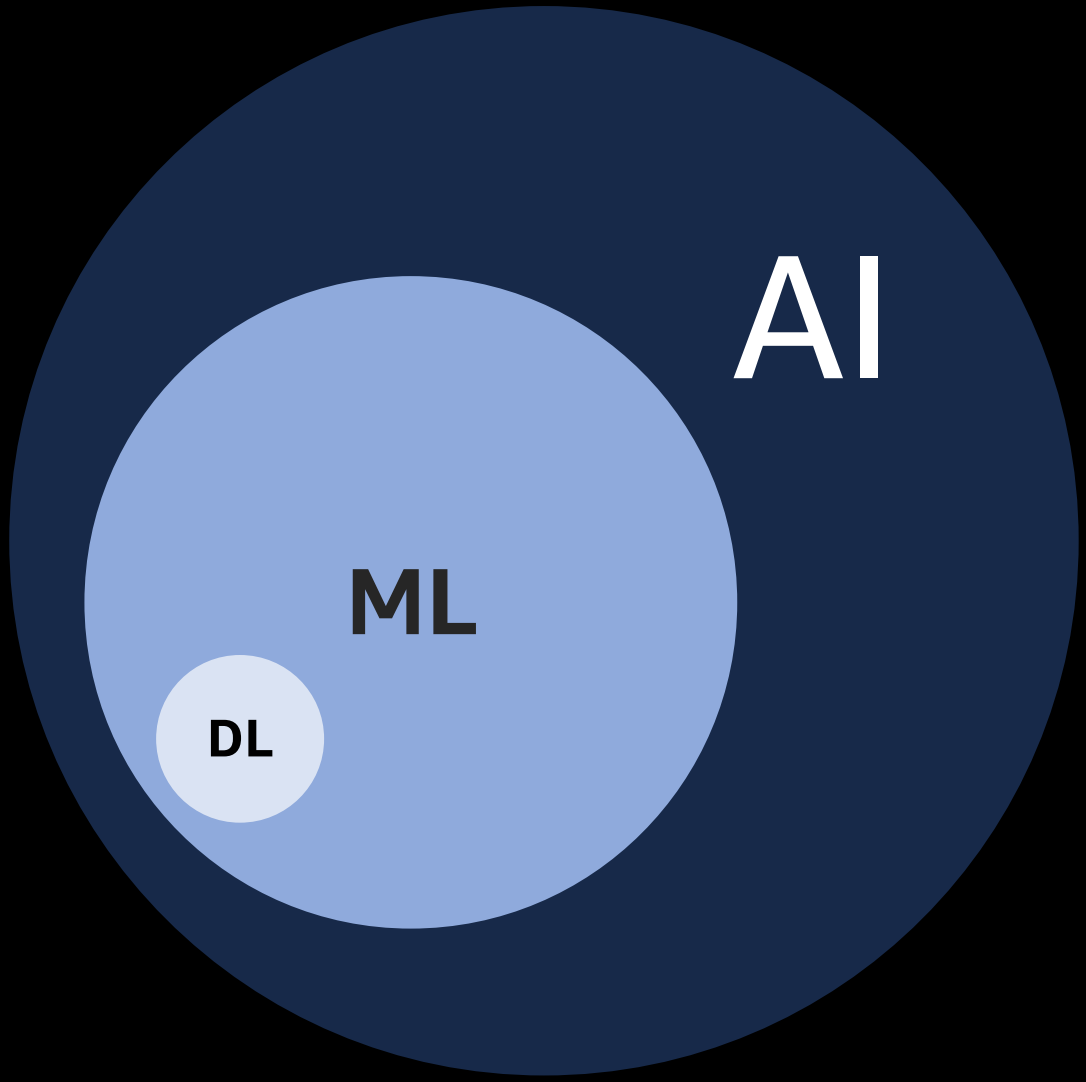
Problem: 24 hours without water for living wall
[low irrigation pressure warning was ignored]

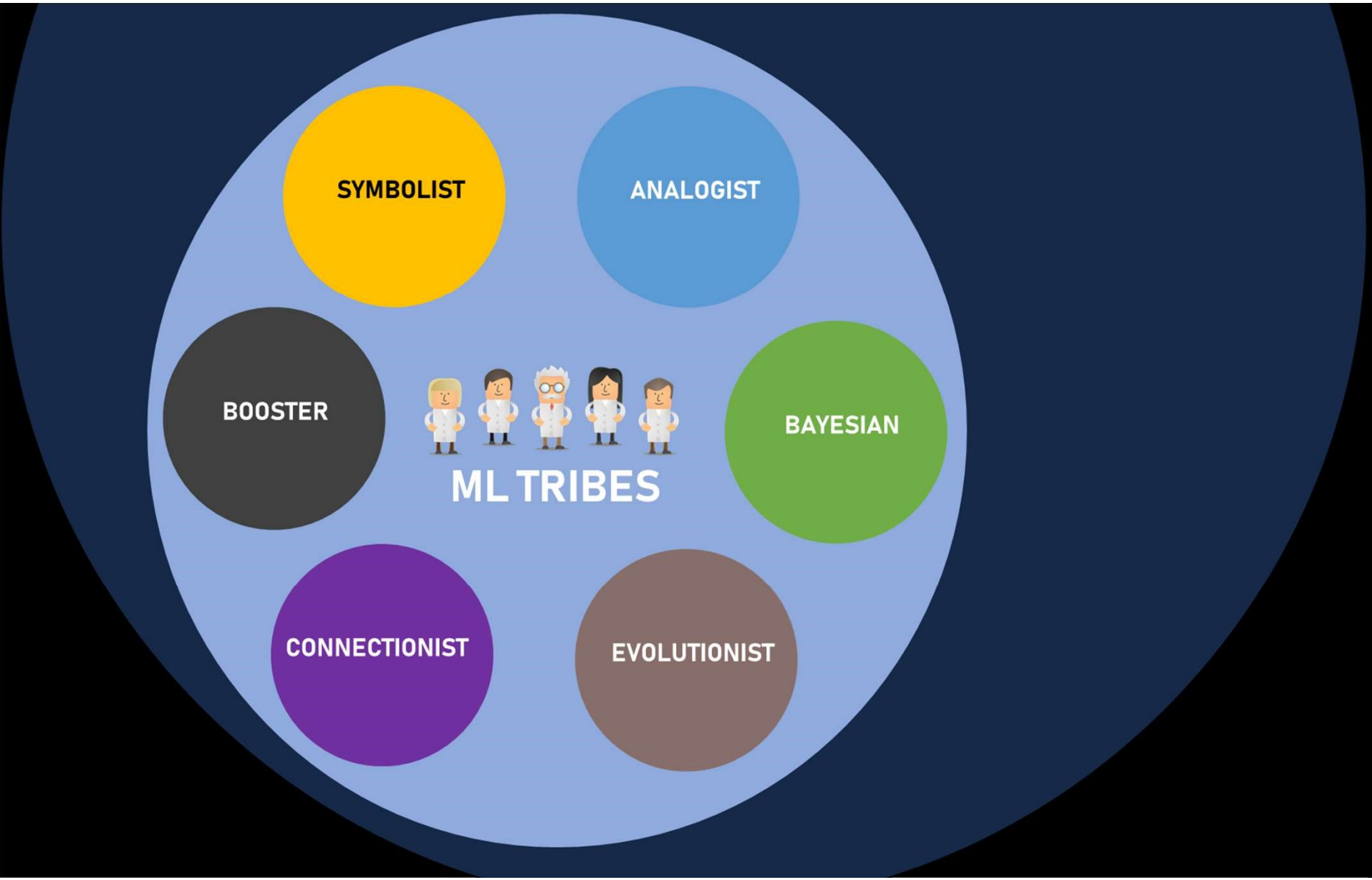
Approach

AI to Assist the caretakers

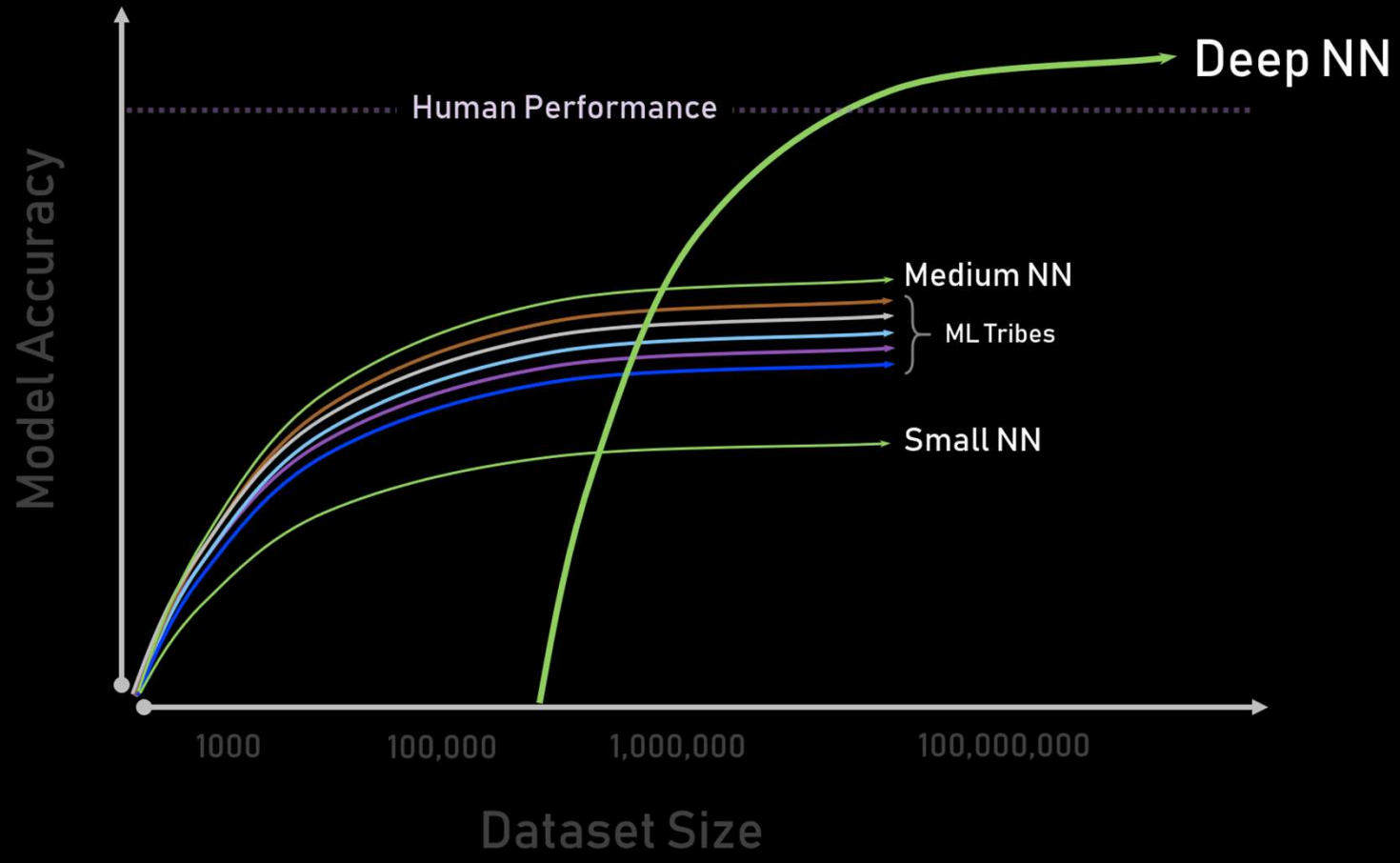
- Accurate Alerts [low false alarm rate]
- Real-time & Low Cost
- Enable Current/Future Science
- Scalability & Availability of Technology




AI
ML
DL

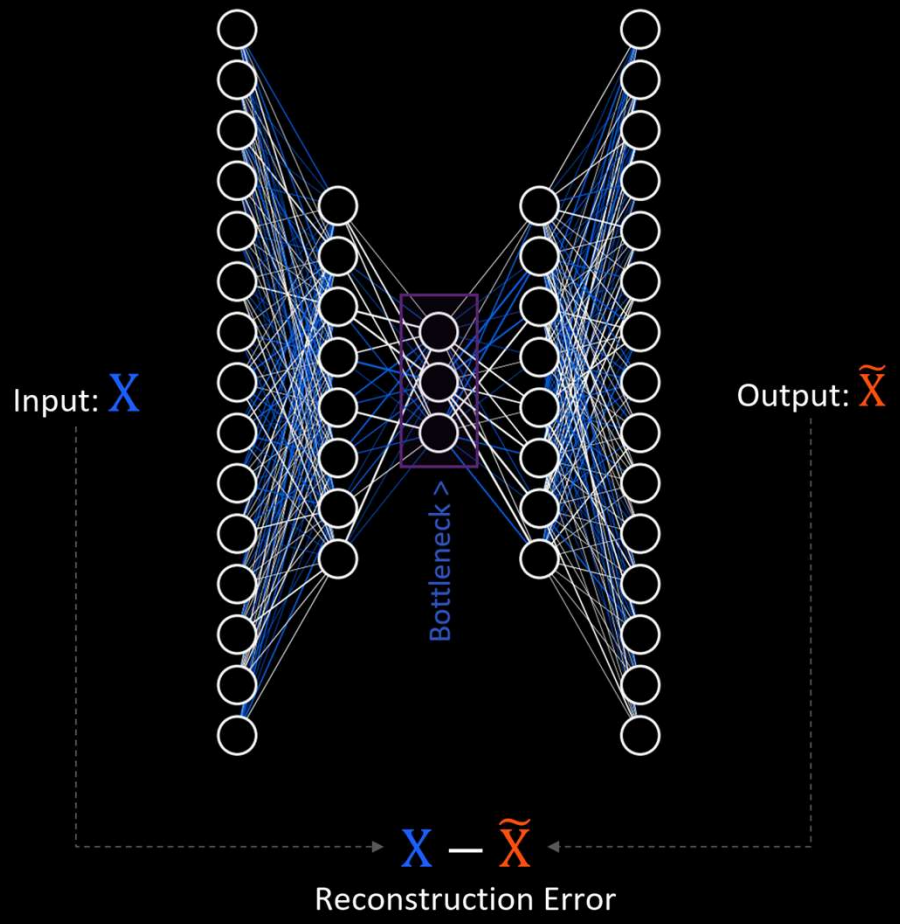

SYMBOLIST
ANALOGIST
BOOSTER
BAYESIAN
ML TRIBES
CONNECTIONIST
EVOLUTIONIST

Why DL


Deep NN
Human Performance
Medium NN
ML Tribes
Small NN
Model Accuracy
1000
100,000
1,000,000
100,000,000
Dataset Size

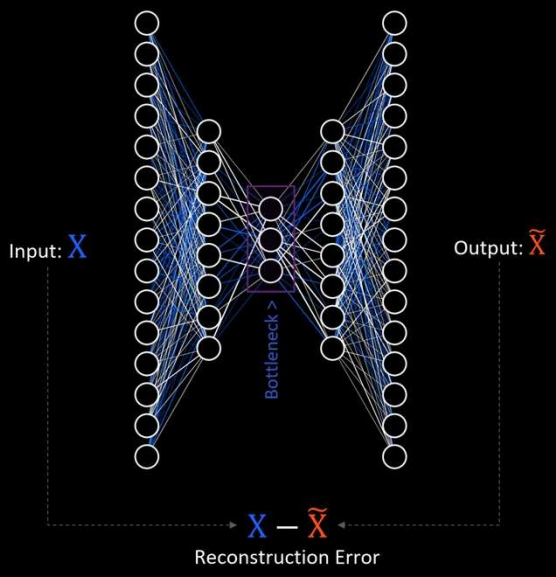
Deep Learning @ Spheres

[AutoEncoder Network]



Deep Learning @ Spheres

[AutoEncoder Network]



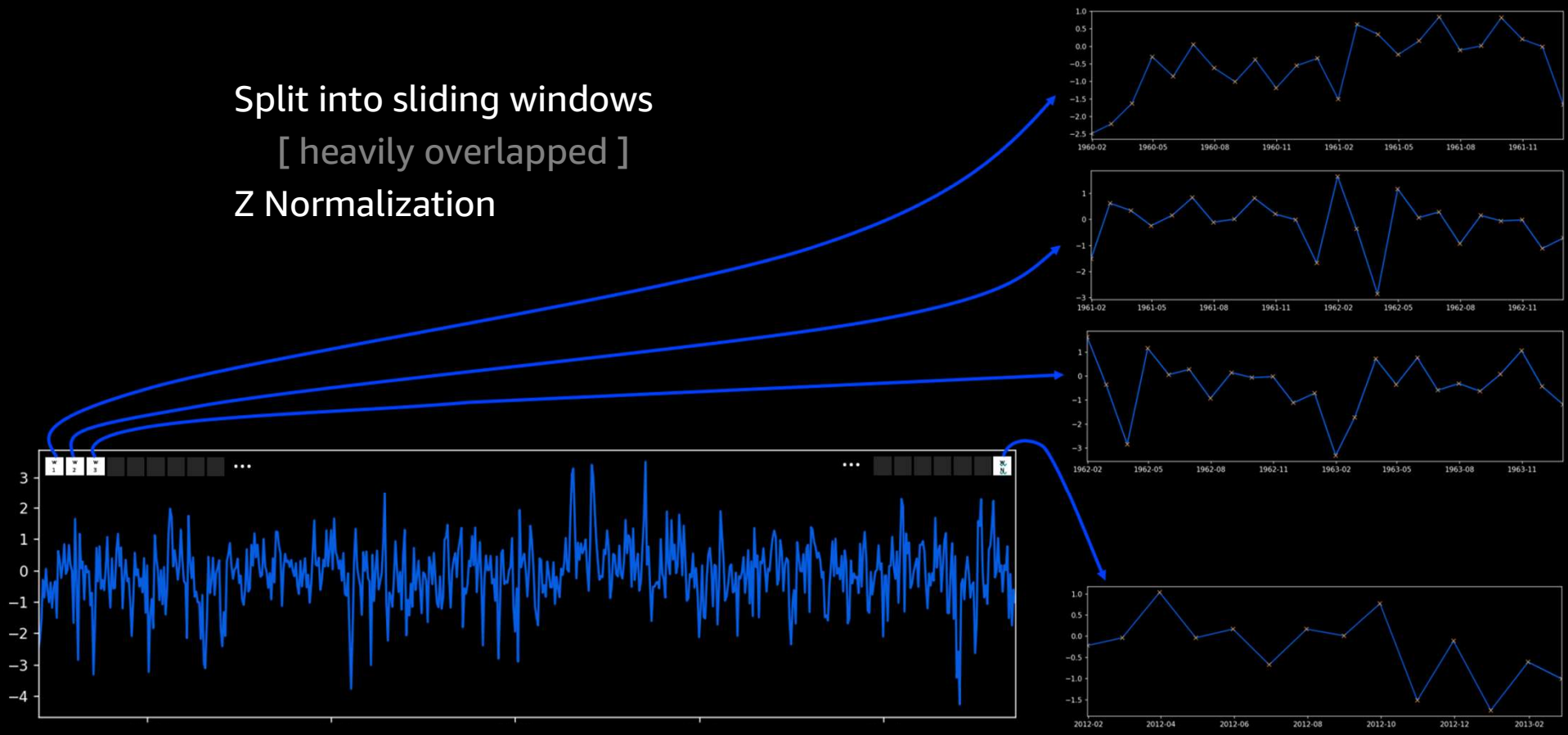

```
(encoderDecoder): Sequential(
  (layer-1-encoder): Linear(in_features=2016, out_features=4032, bias=True)
  (actFn-1-encoder): Sigmoid()
  (dropout-1-encoder): Dropout(p=0.1)
  (layer-2-encoder): Linear(in_features=4032, out_features=2016, bias=True)
  (actFn-2-encoder): Sigmoid()
  (dropout-2-encoder): Dropout(p=0.1)
  (layer-3-encoder): Linear(in_features=2016, out_features=50, bias=True)
  (actFn-3-encoder): Sigmoid()
  (layer-4-decoder): Linear(in_features=50, out_features=2016, bias=True)
  (actFn-4-decoder): Sigmoid()
  (layer-5-decoder): Linear(in_features=2016, out_features=4032, bias=True)
  (actFn-5-decoder): Sigmoid()
  (dropout-5-decoder): Dropout(p=0.1)
  (layer-6-decoder): Linear(in_features=4032, out_features=2016, bias=True)
)
```

Preparing Data for Model Training

Split into sliding windows

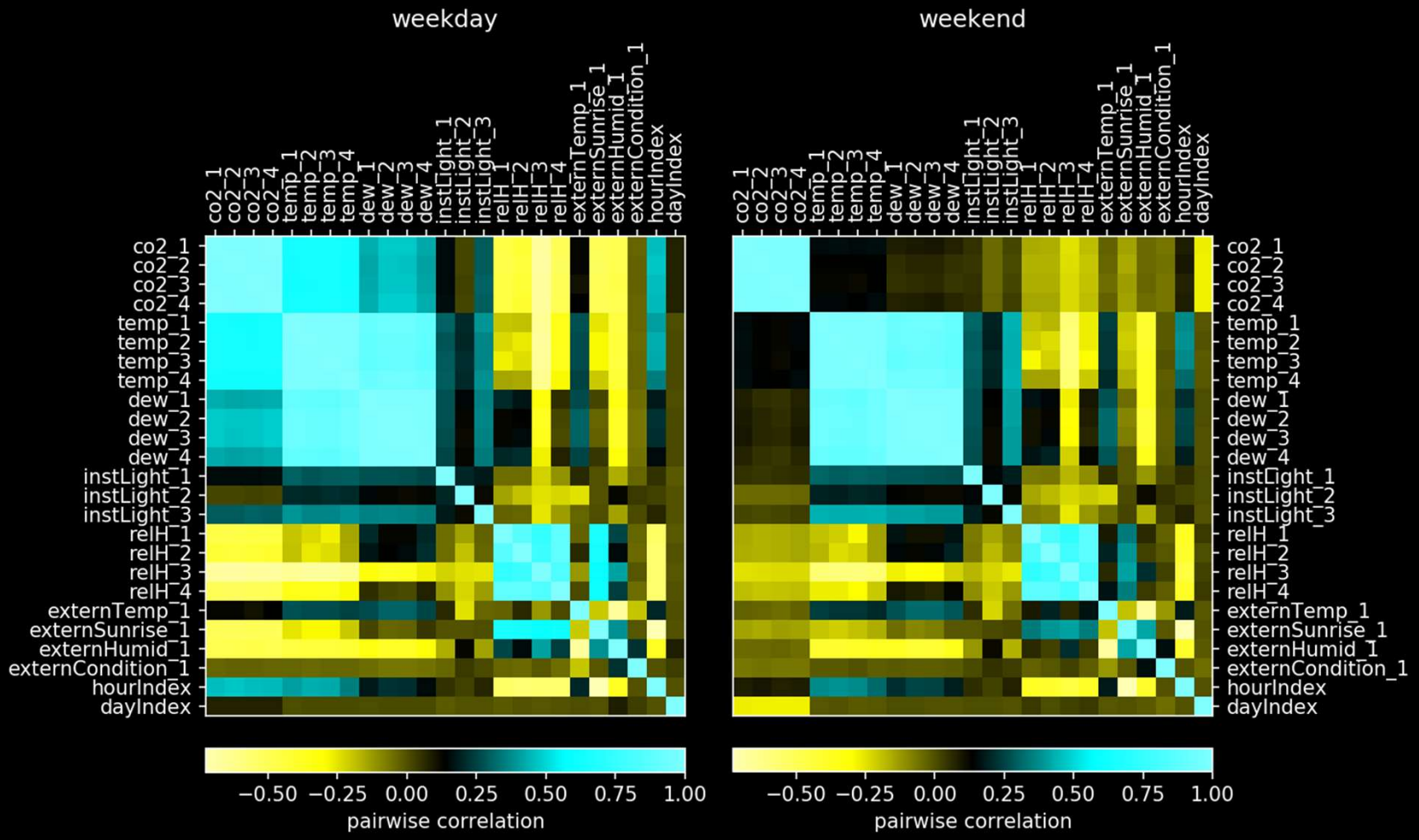
[heavily overlapped]

Z Normalization



Correlated Sensors

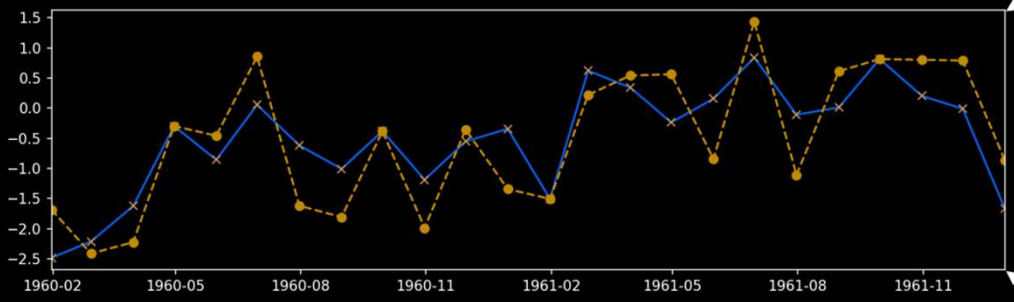
[Weekday & Weekend Behaviors]



Detecting Anomalies

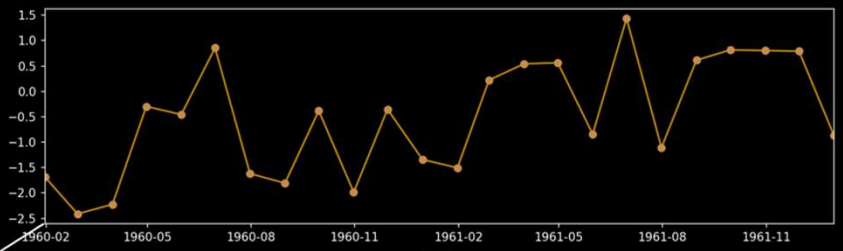
Reconstruction error (RE) as a proxy to outliers

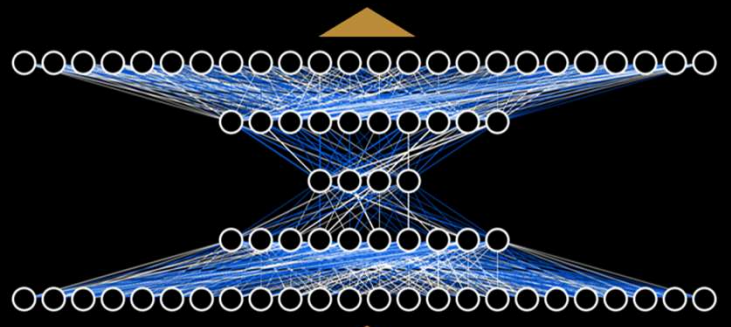
Whenever RE is high, get an alert

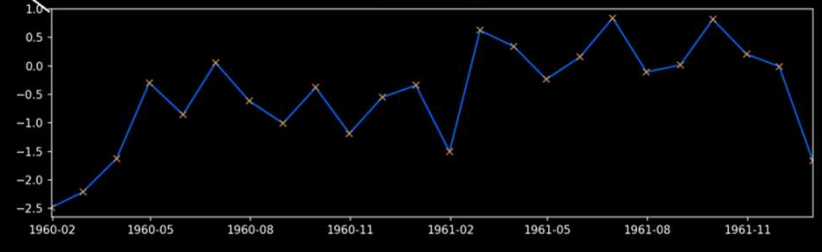


Reconstruction vs Input

Output (Reconstruction)



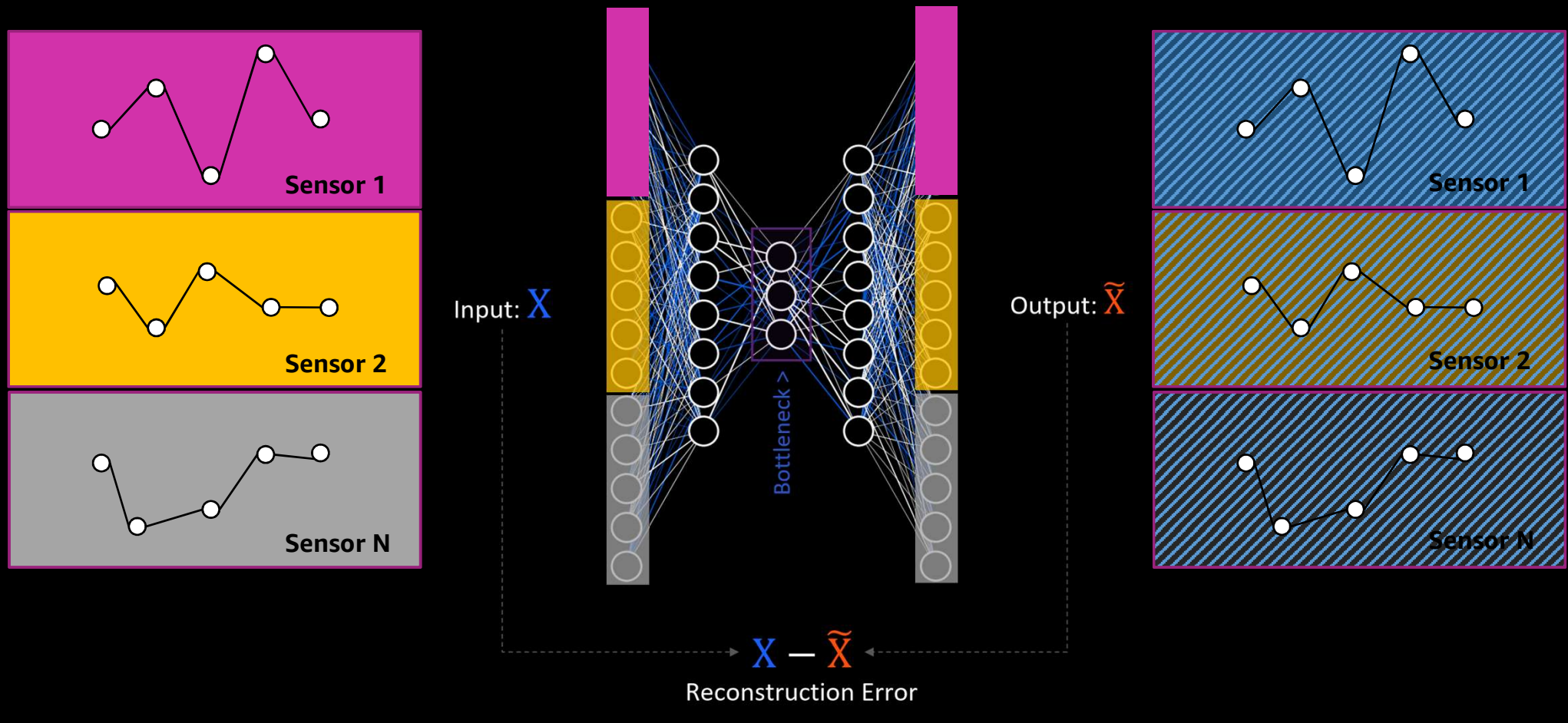




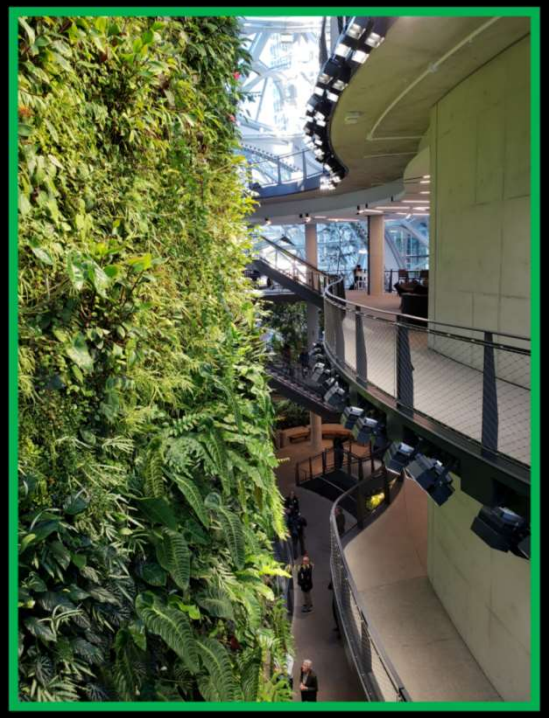
Input

Multi Sensor Models

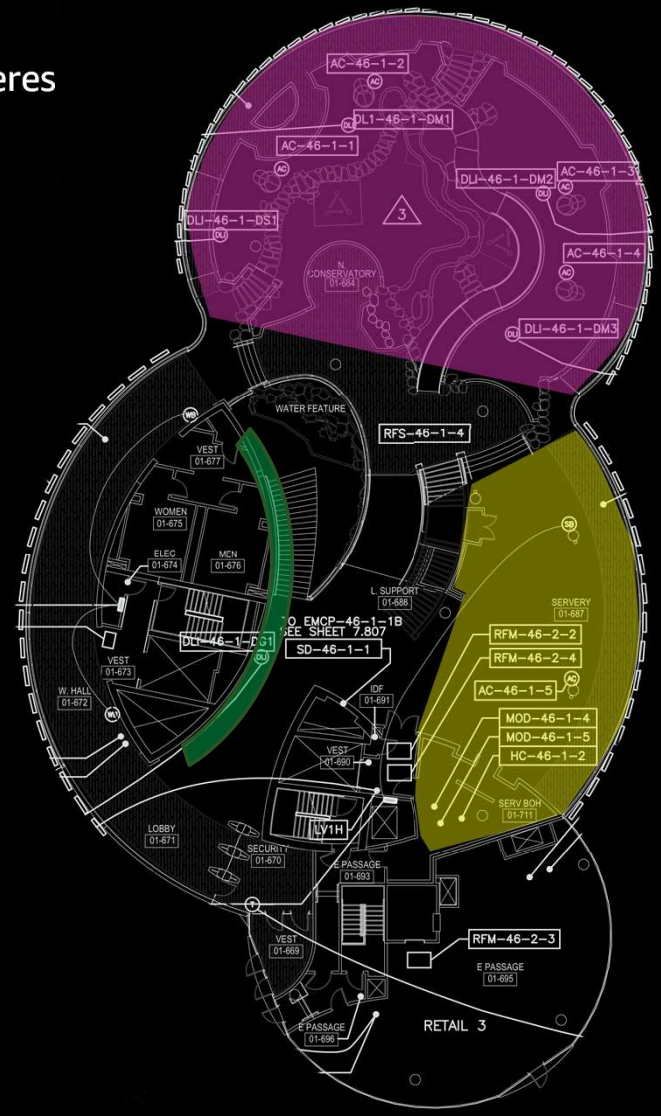
[AutoEncoder Network]



Inside the Spheres
[1st floor]



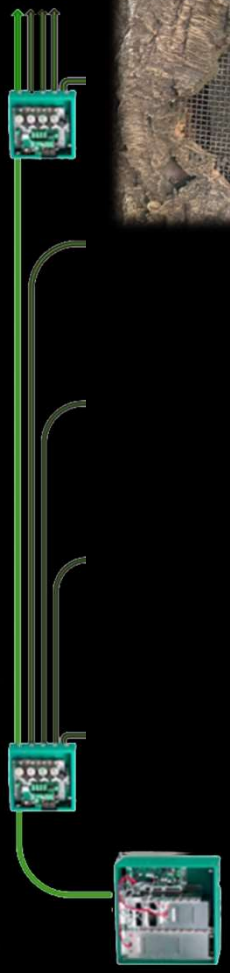
Living Wall

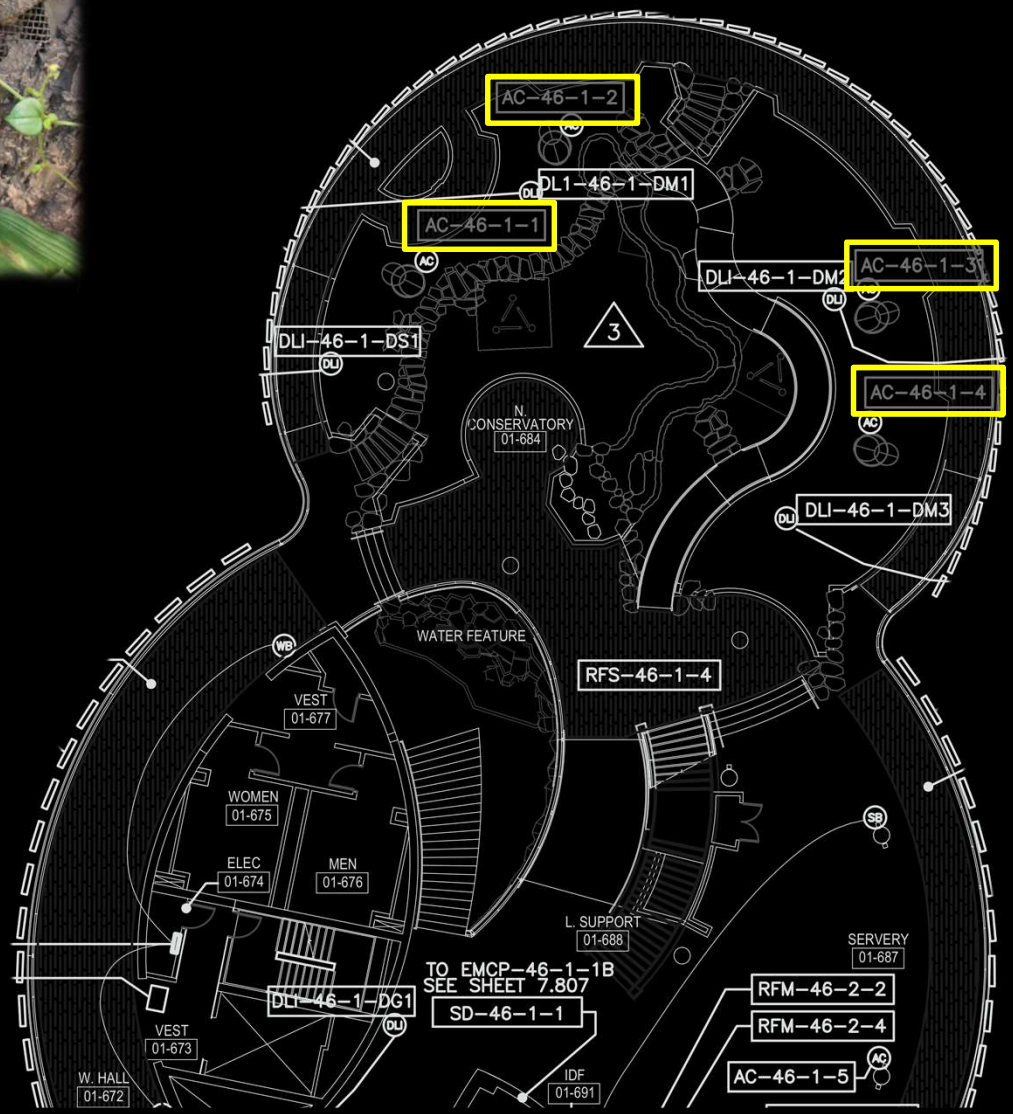


North Conservatory

Cafe

AirCuity
Sensors





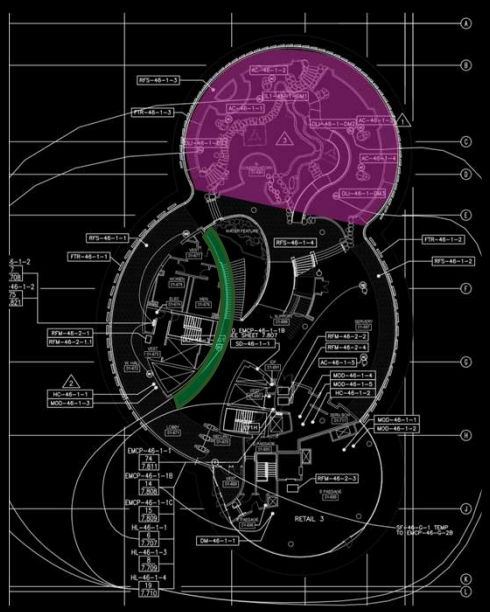
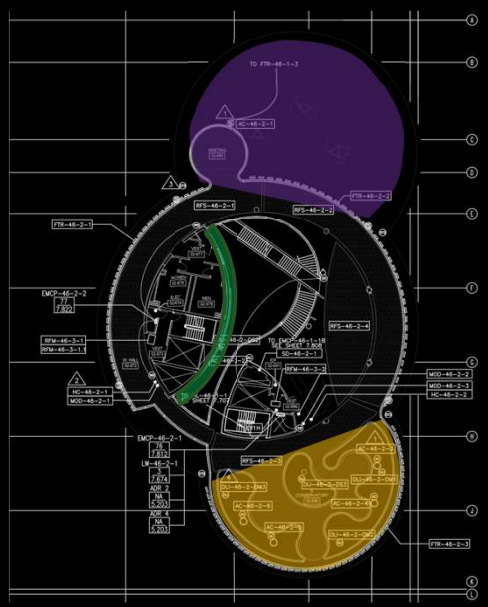
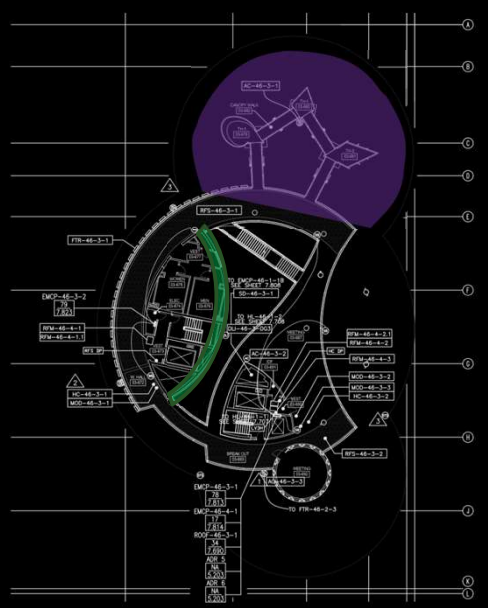
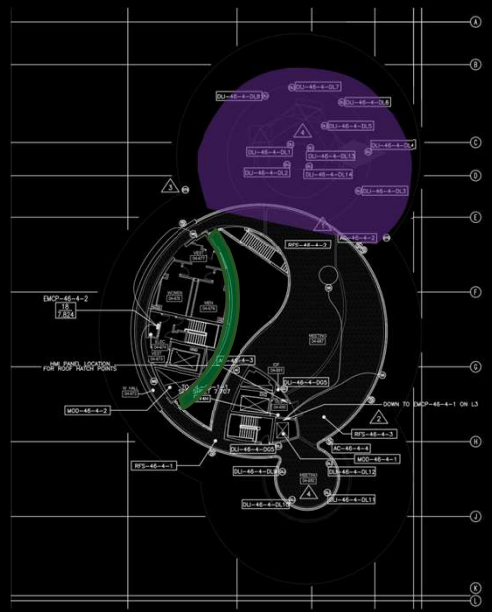
Sensor Zones

Living Wall [4 floors]
t,rh,d,co2 [X, AC-46-2-2, AC-46-3-2, AC-46-4-3]
light level [DLI-46-1-DG1, DLI-46-2-DG2, DLI-46-3-DG3, DLI-46-4-DG5]

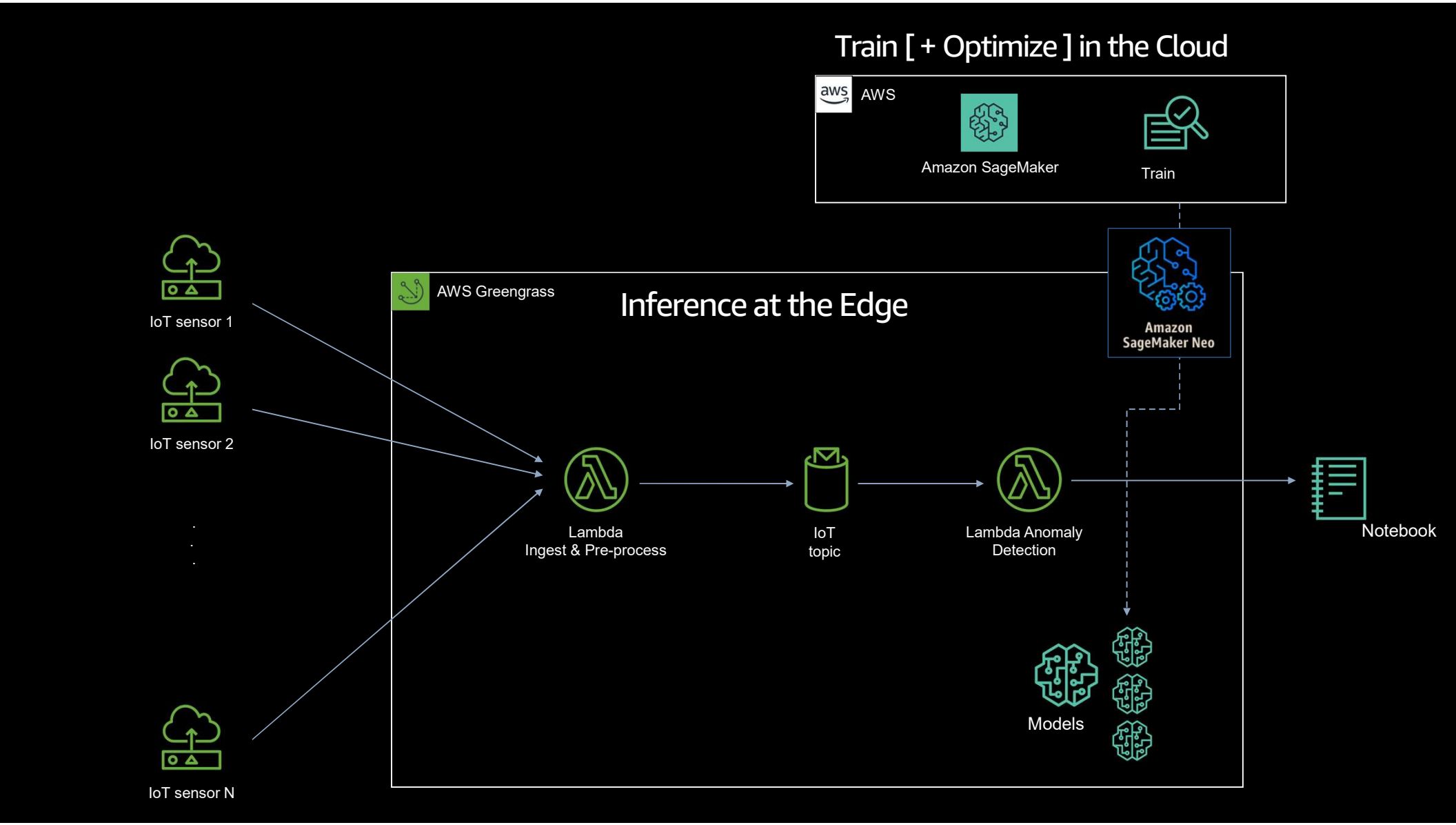
North Conservatory [1st floor]
t,rh,d,co2 [AC-46-1-1, AC-46-1-2, AC-46-1-3, AC-46-1-4]
light level [DLI-46-1-DS1, DLI-46-1-DM2, DLI-46-1-DM3]

South Conservatory [2nd floor]
t,rh,d,co2 [AC-46-2-3, AC-46-2-4, AC-46-2-5, AC-46-2-6]
light level [DLI-46-2-DM1, DLI-46-2-DM2, DLI-46-2-DM3]

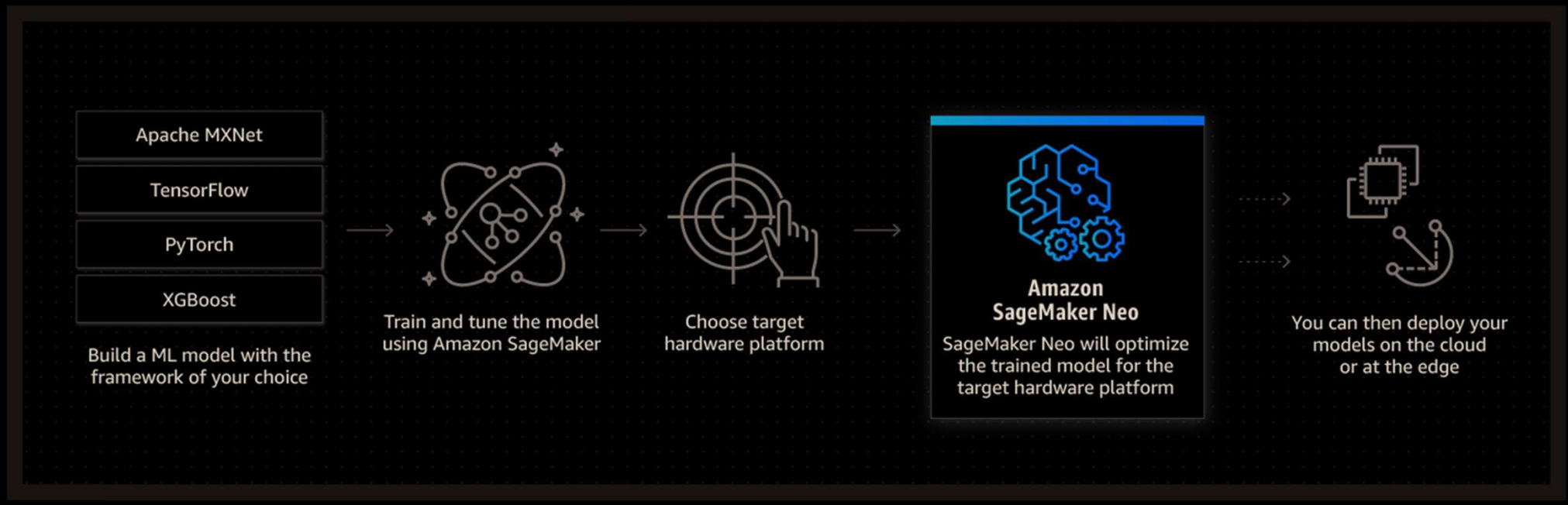
Canopy [3 floors above N. Conservatory]
t,rh,d,co2 [AC-46-2-1, AC-46-3-1, AC-46-4-2]
light level [DLI-46-4-DL1, DLI-46-4-DL2, DLI-46-4-DL3, DLI-46-4-DL4, DLI-46-4-DL5, DLI-46-4-DL6, DLI-46-4-DL7, DLI-46-4-DL8, DLI-46-4-DL13, DLI-46-4-DL14]

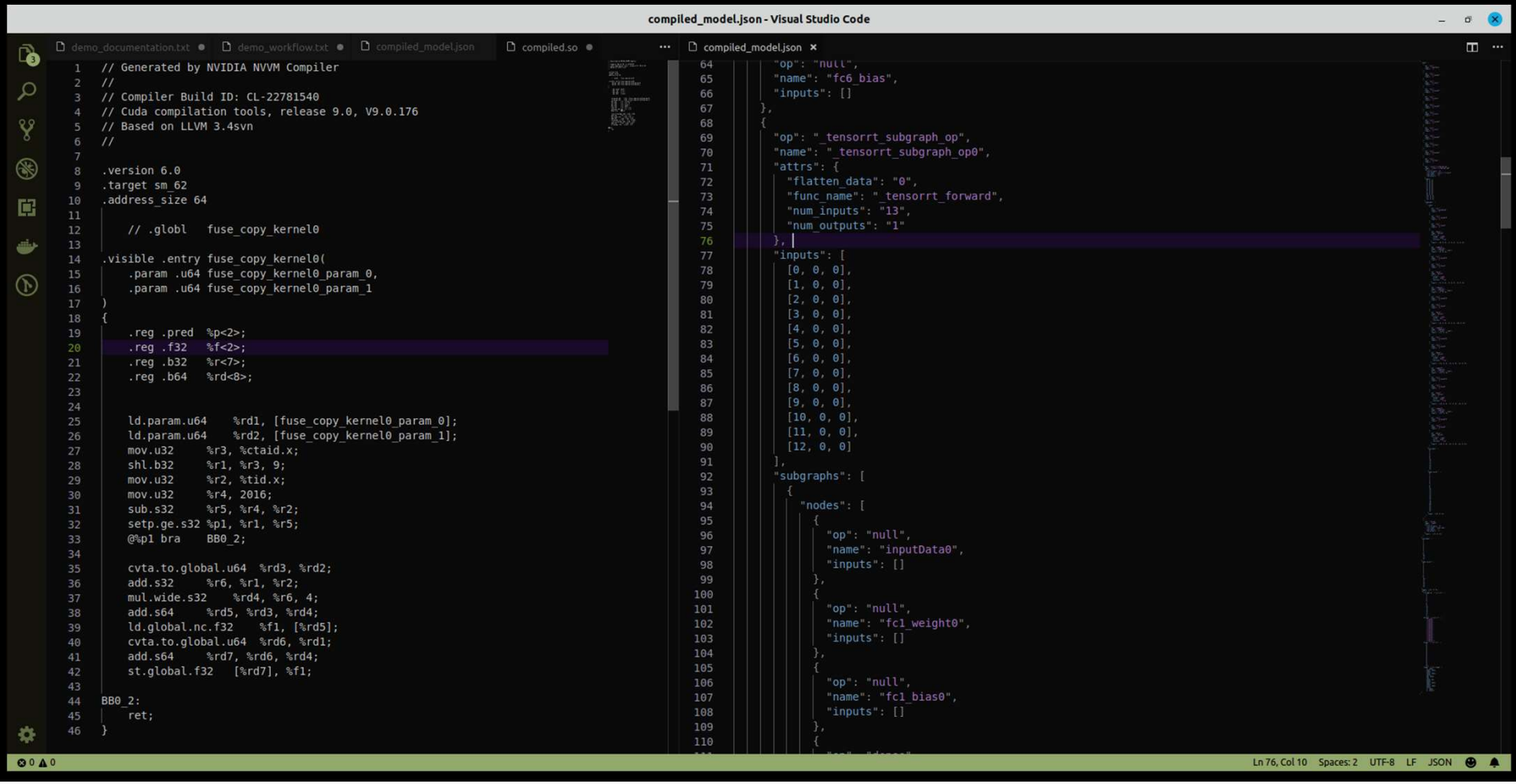
Architecture


Train [+ Optimize] in the Cloud
AWS
Amazon SageMaker
Train
Amazon SageMaker Neo
AWS Greengrass
Inference at the Edge
IoT sensor 1
IoT sensor 2
IoT sensor N
Lambda
Ingest & Pre-process
IoT
topic
Lambda Anomaly
Detection
Notebook
Models

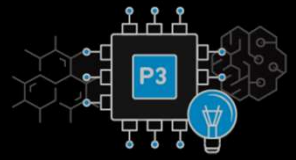
Amazon SageMaker Neo



Amazon SageMaker Neo



Training Architecture @ p3.4xlarge



Query → Process → Train → Eval → Store

Zone_1
Sensors
Time Range
S3.cache FileName
⋮
Zone_2
Zone_3
Zone_4

Process
Resample [15 m]
Group Scale Stats.
Weekday Extract
Trim/Extend Windows
Add Time Ref.
Numpy.fp32

Model_1: co2
Input Sensors
Recon. Target Sensors
Encoder Dims
Decoder Dims
Optional Hyper Params
Convert to ONNX
⋮

Model_2: temp	Model_4: relH
Model_3: dew	Model_5: Inst Light

Eval
Sliding Window Reconstruction
Mean + Std. Dev

z1_m1_latest.zip
ONNX & pyTorch
Time Range
Scale Stats.
Encoder/Decoder Dims
Input Sensors
Recon. Target Sensors

z1_m2_latest.zip
z1_m5_latest.zip

Inference Architecture @ Jetson Xavier

Zone_N
Sensors
Time Range
S3.cache FileName

Inference Params [s3]
Scale Stats.
Input Sensor Sets
Recon. Target Sensors

Process
Resample [15 m]
Apply Scale Stats.
Weekday Extract
Trim/Extend Windows
Add Time Ref.
Numpy.fp32

Eval
Sliding Window Reconstruction
Compare to Mean + Std. Dev Stats

Reconstruction Error Alerts	
1 : 3 * stDev	Green
3 : 6 * stDev	Yellow
> 6 * stDev	Red

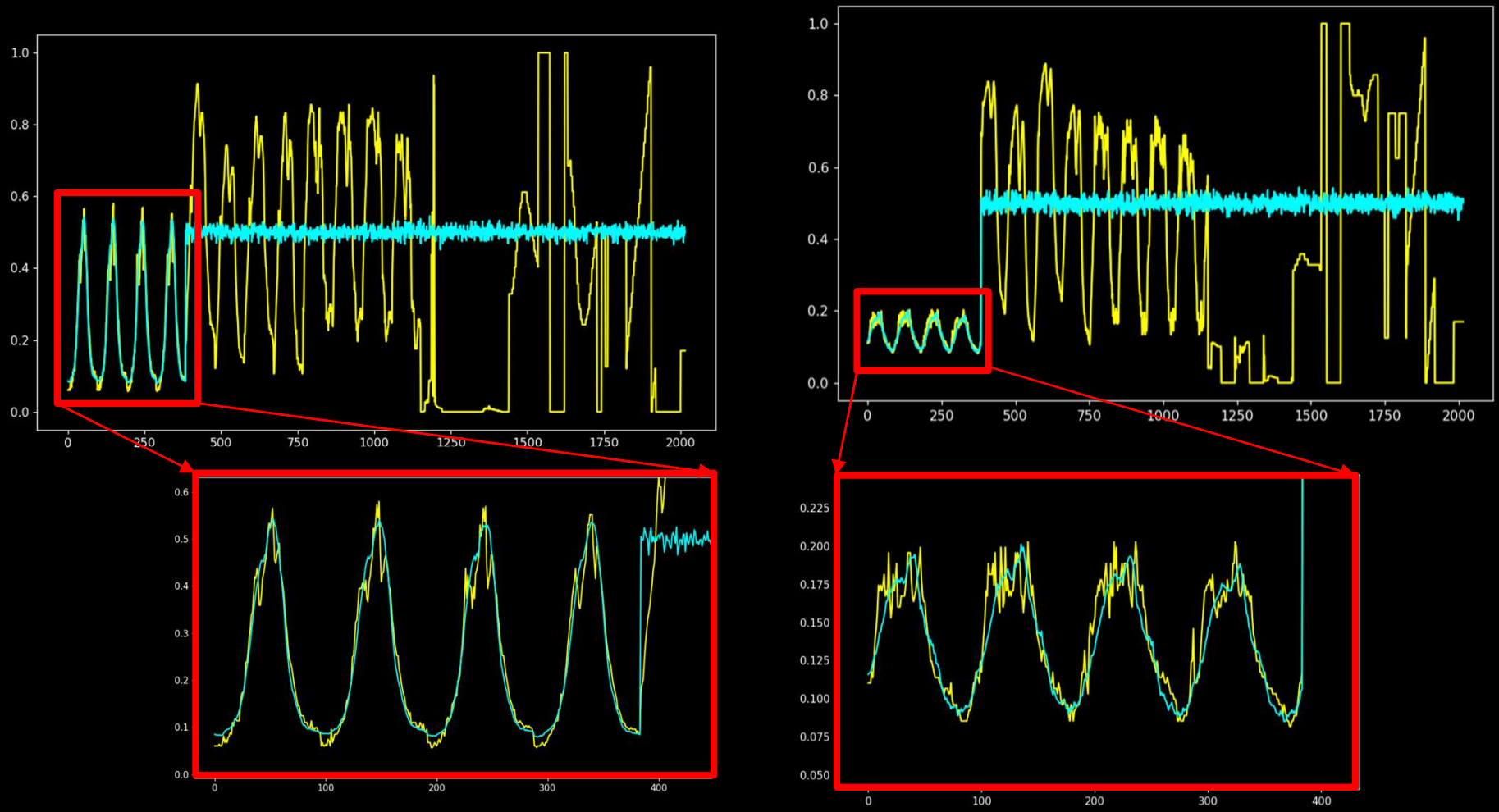
Assign Label
Anomaly (Y/N)
Anomaly Category

Store
+ Data Timestamps
+ Reconstruction Error in Window

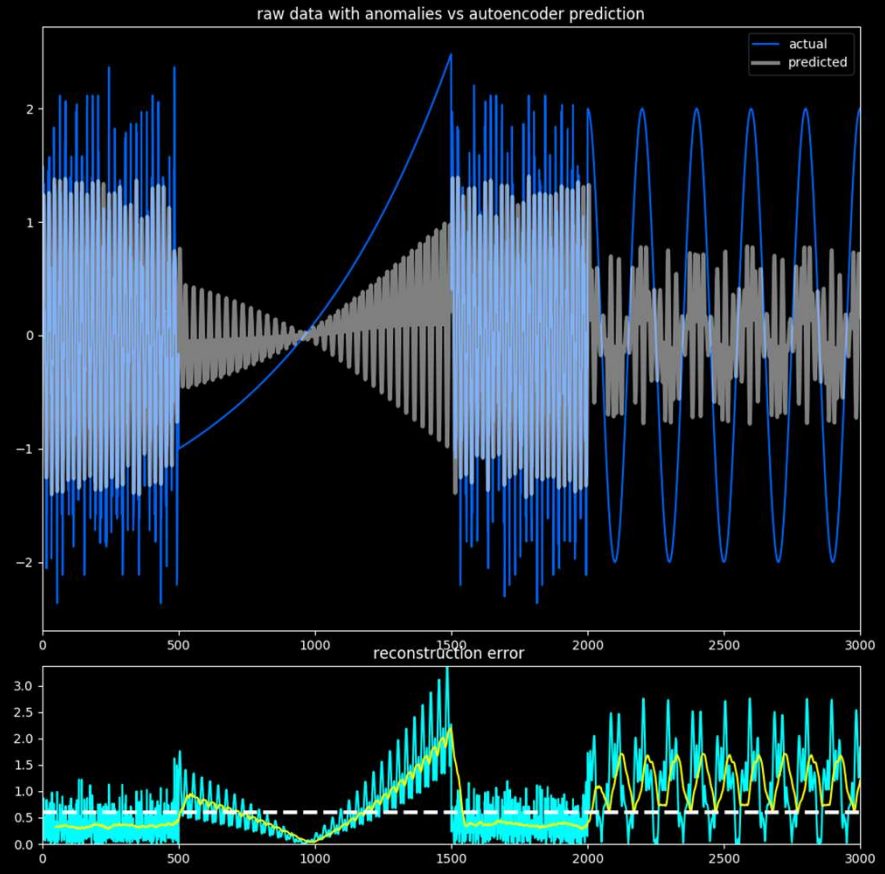
Notebook Demo

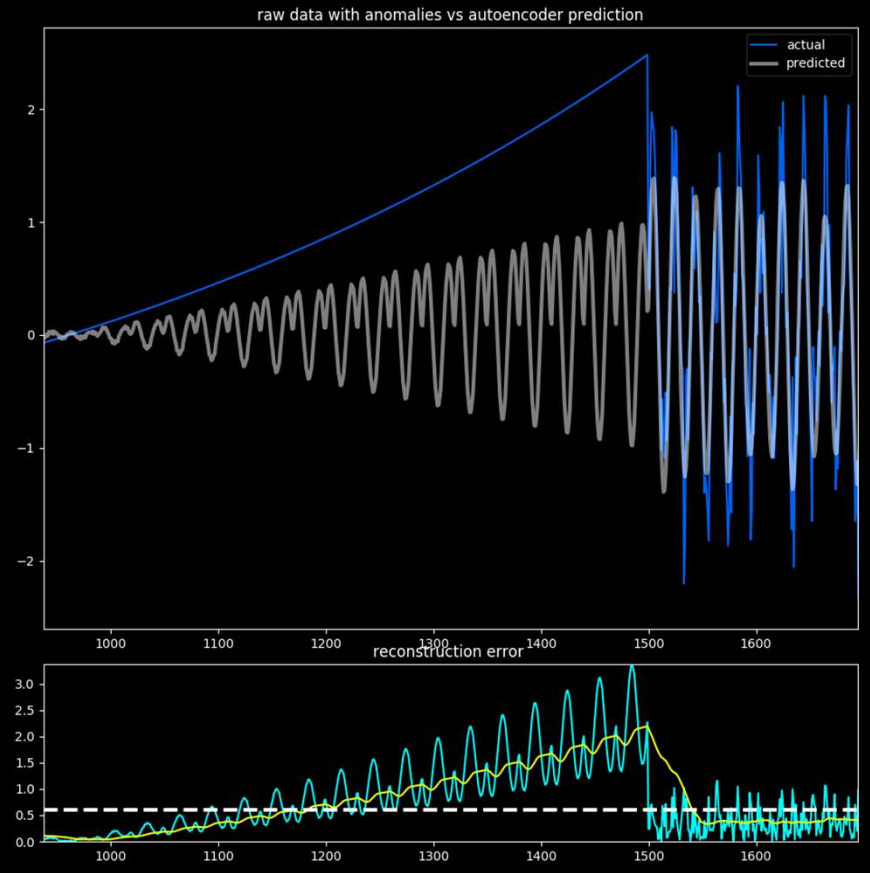
Results

Sample Reconstructions

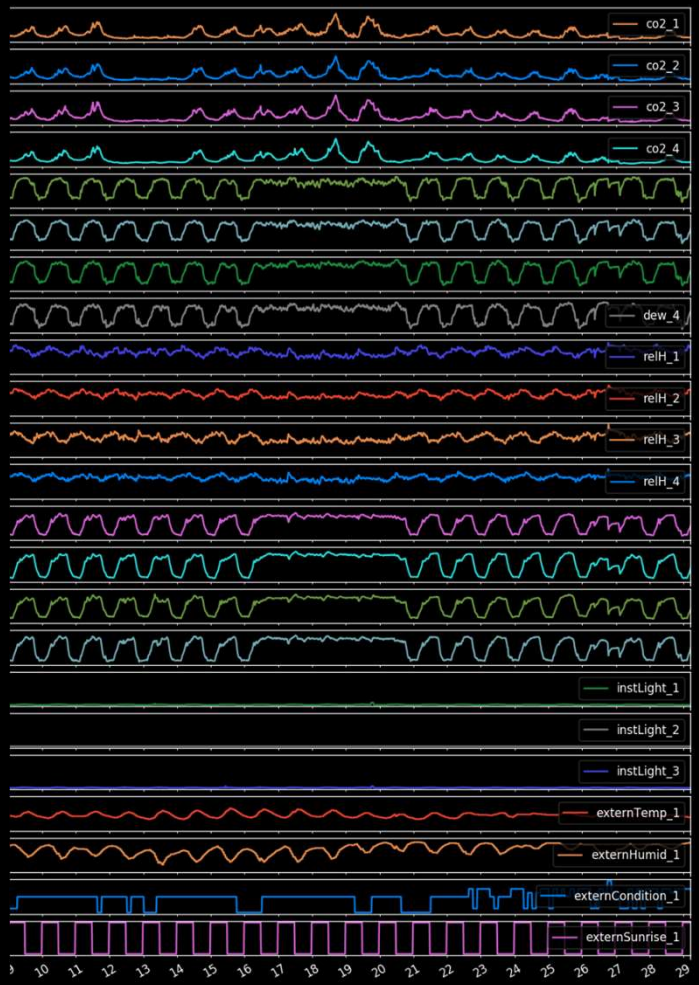


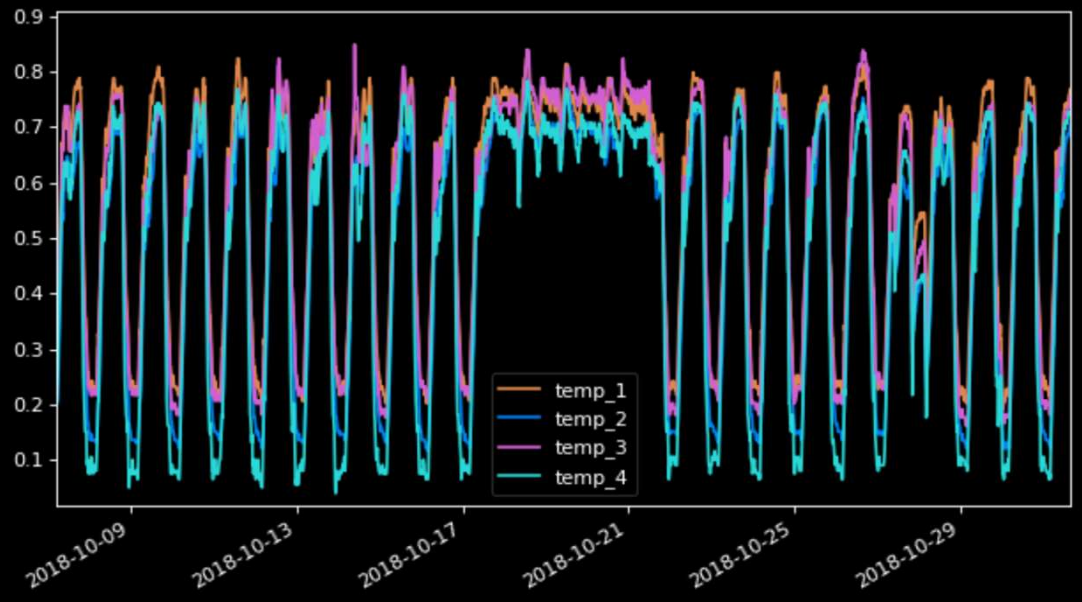
[Synthetic] Anomaly Detection





Real Anomaly Detection

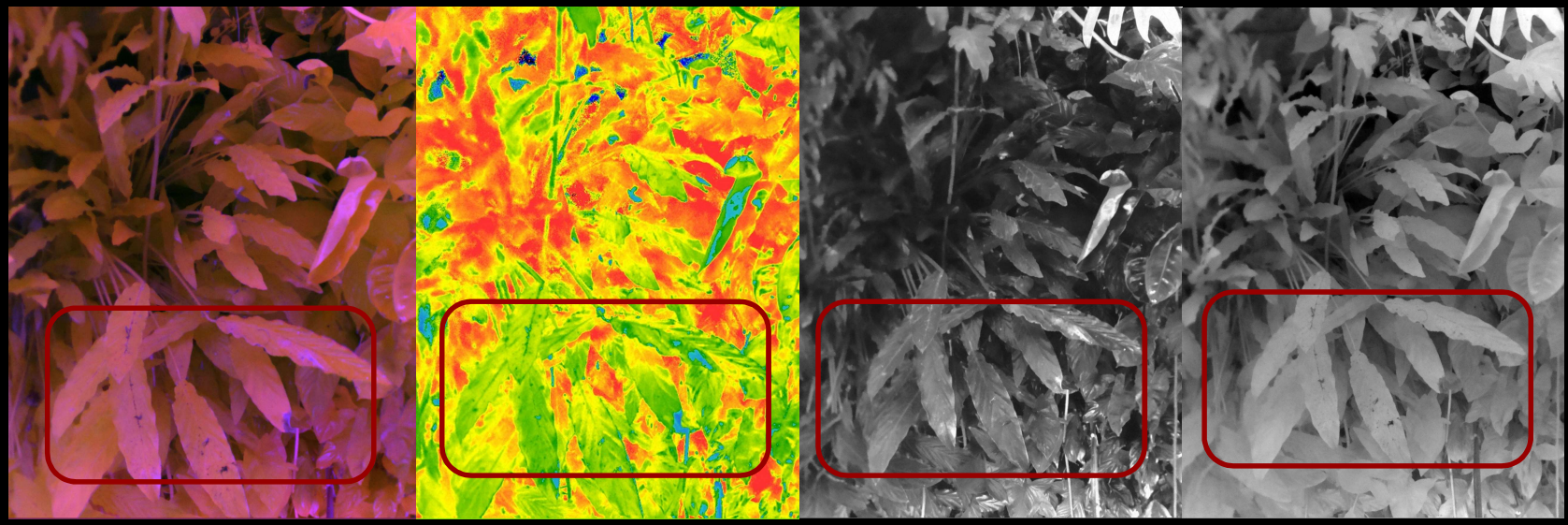


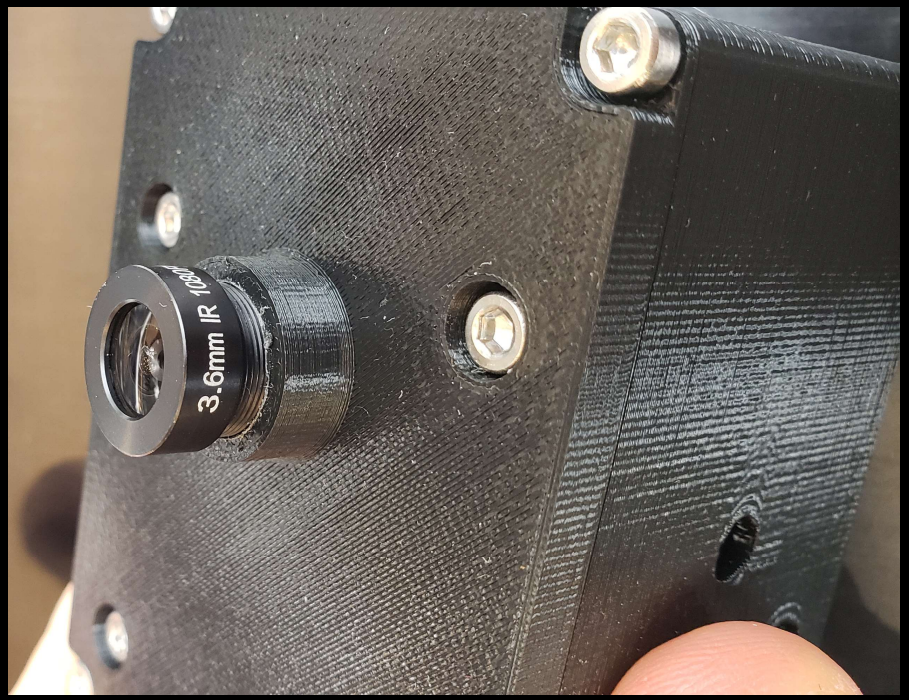


“ Nice catch. We altered the climate to encourage the blooming of our Amorphophallus titanum plant. The corpse flower is more accustomed to warmer temps and higher humidity than the normal spheres operating parameters. “

Future Work

Multi-spectral Imaging




3.6mm IR

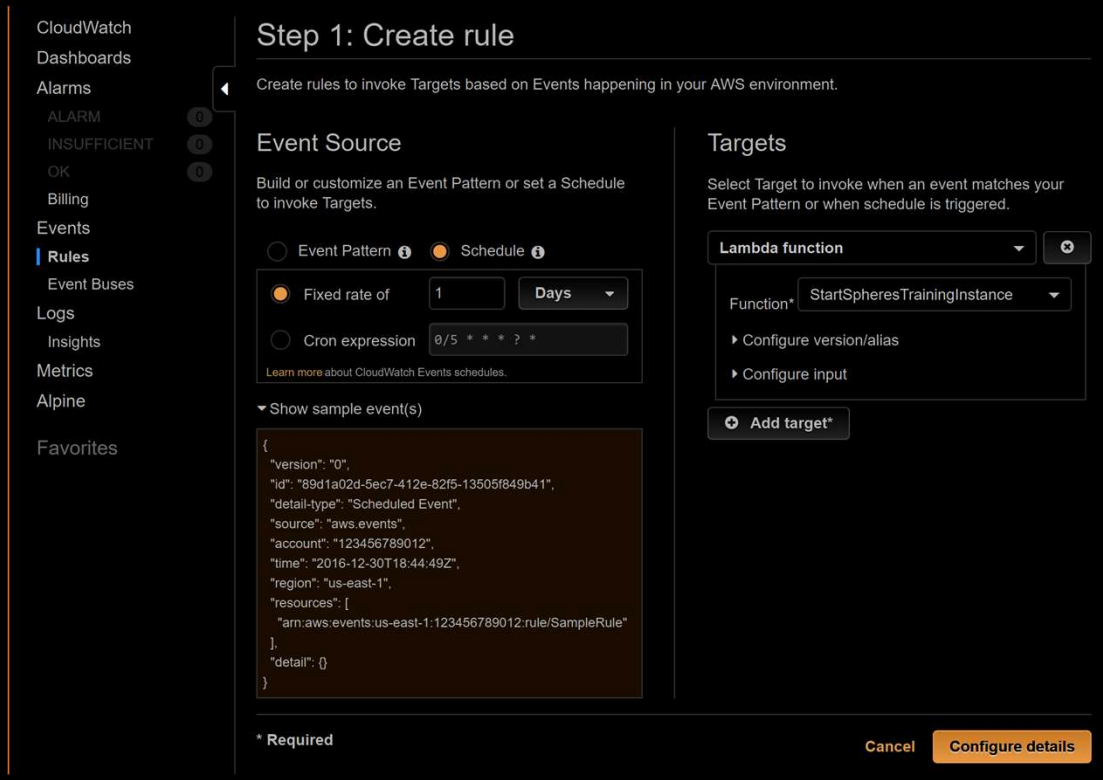
Discussion & Q/A

Thank you!

WenMing Ye - wye@amazon.com

Miro Enev - menev@nvidia.com

Scheduled Lambdas Trigger Training and Batch Inference

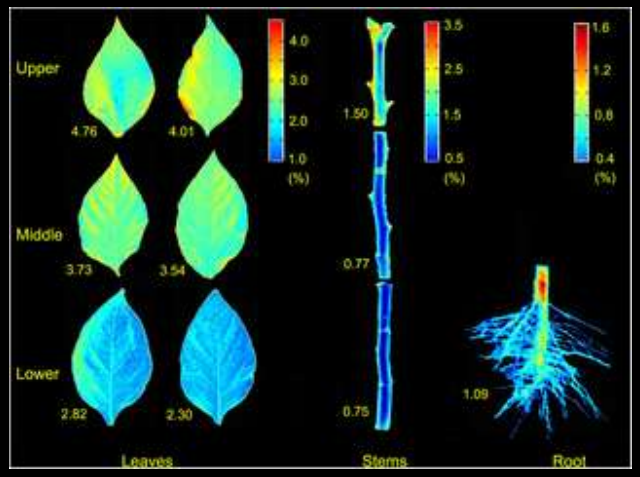



```
def lambda_handler(event, context):
    ec2 = boto3.client('ec2', region_name=region)
    ec2.start_instances(InstanceIds=instances)
    print 'started your instances: ' + str(instances)
```

```
def lambda_handler(event, context):
    ec2 = boto3.client('ec2', region_name=region)
    ec2.stop_instances(InstanceIds=instances)
    print 'stopped your instances: ' + str(instances)
```

https://aws.amazon.com/premiumsupport/knowledge-center/start-stop-lambda-cloudwatch/

Multi-spectral Imaging & Computer Vision



Edge Processing + TensorRT