MICHIGAN STATE UNIVERSITY

Beta Presentation

ML/AI Pipeline For Condition-Based Maintenance

The Capstone Experience

Team Magna AI4CBM

Michael Gryn
Daniel Chen
Hector Dominquez Rojas
Lizabeth Hanks
Ethan Springer
Athul Syam

Department of Computer Science and Engineering
Michigan State University

Fall 2025

From Students…
…to Professionals

Project Overview

- Unplanned downtime is costly
- Spot anomalies before becoming a problem
- Developed a ML pipeline to ingest machine diagnostic data
- Intuitive dashboard for monitoring all Magna machinery through live sensor readings and model output

Team Member's Technical Tasks

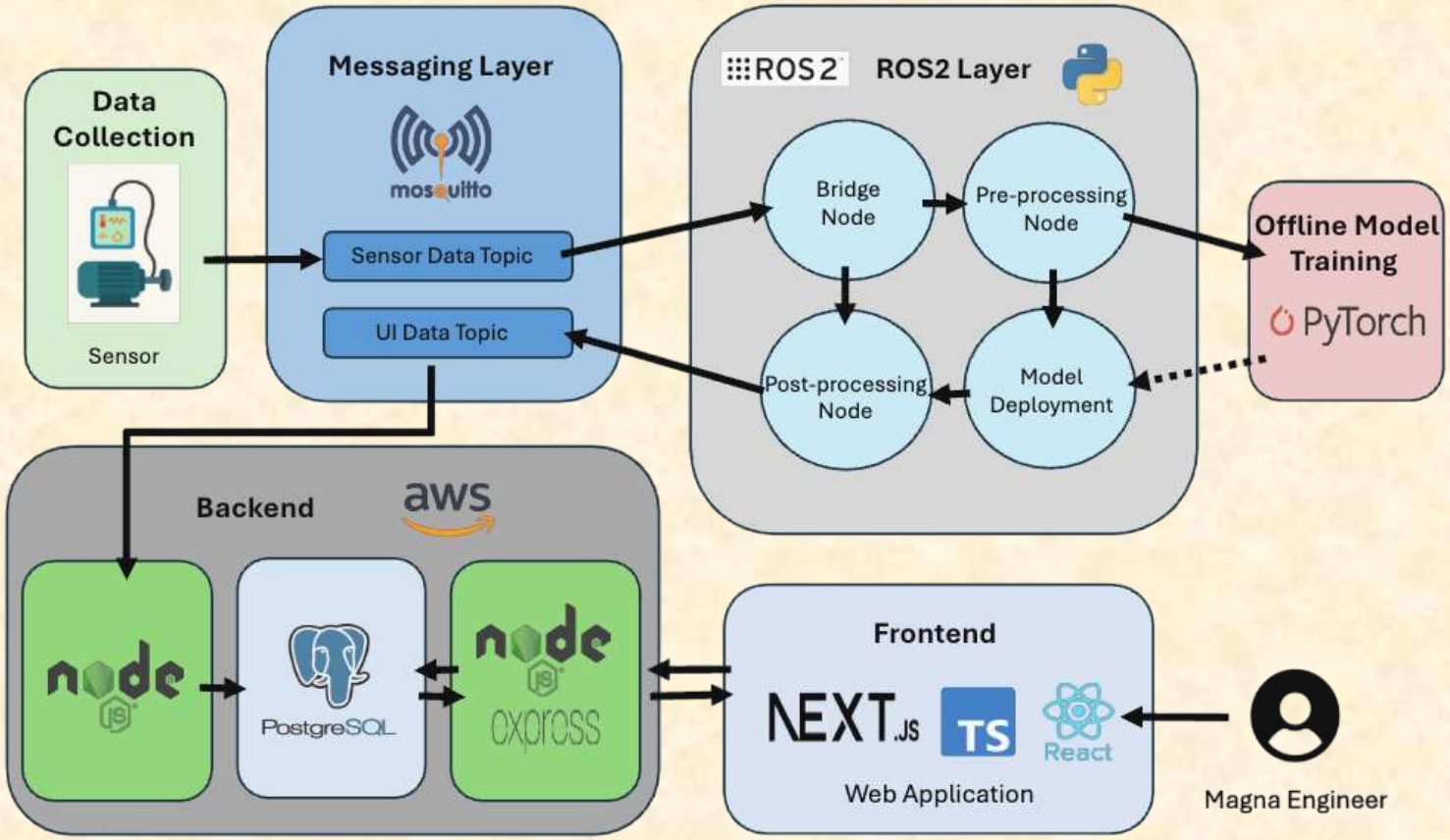
Technical Tasks Assigned

- Team Member Daniel Chen
  - Dataset Selection
  - Training Preprocessing
  - Model Architecture
  - Model Training
- Team Member Ethan Springer
  - MQTT Bridge Node
  - Admin Control Node/Handling
  - Model Deployment Node(in part)
  - Simulation/Testing Node
  - ROS2 Pipeline Adjustments throughout Development
- Team Member Athul Syam
  - Developed final Taskboard Page
  - Created initial Log Pages
  - Developed Notification signup/sending system
- Team Member Hector Dominguez Rojas
  - Developed final Dashboard Page
  - Developed final Log page
  - Developed feature selection and upper/lower bound on UI
  - Created initial Taskboard
- Team Member Liz Hanks
  - Postprocessing node (Formatting raw and model output data to store in db)
  - AWS server set up
  - Initial postgresql setup
  - Backend implementation
    - Mqtt subscriber node (Handling mqtt connections from frontend)
    - Backend node (Endpoints that query postgresql tables)
- Team Member Michael Gryn
  - Data preprocessing in pipeline
  - UI Backend endpoint support
  - Configuration file for pipeline instances

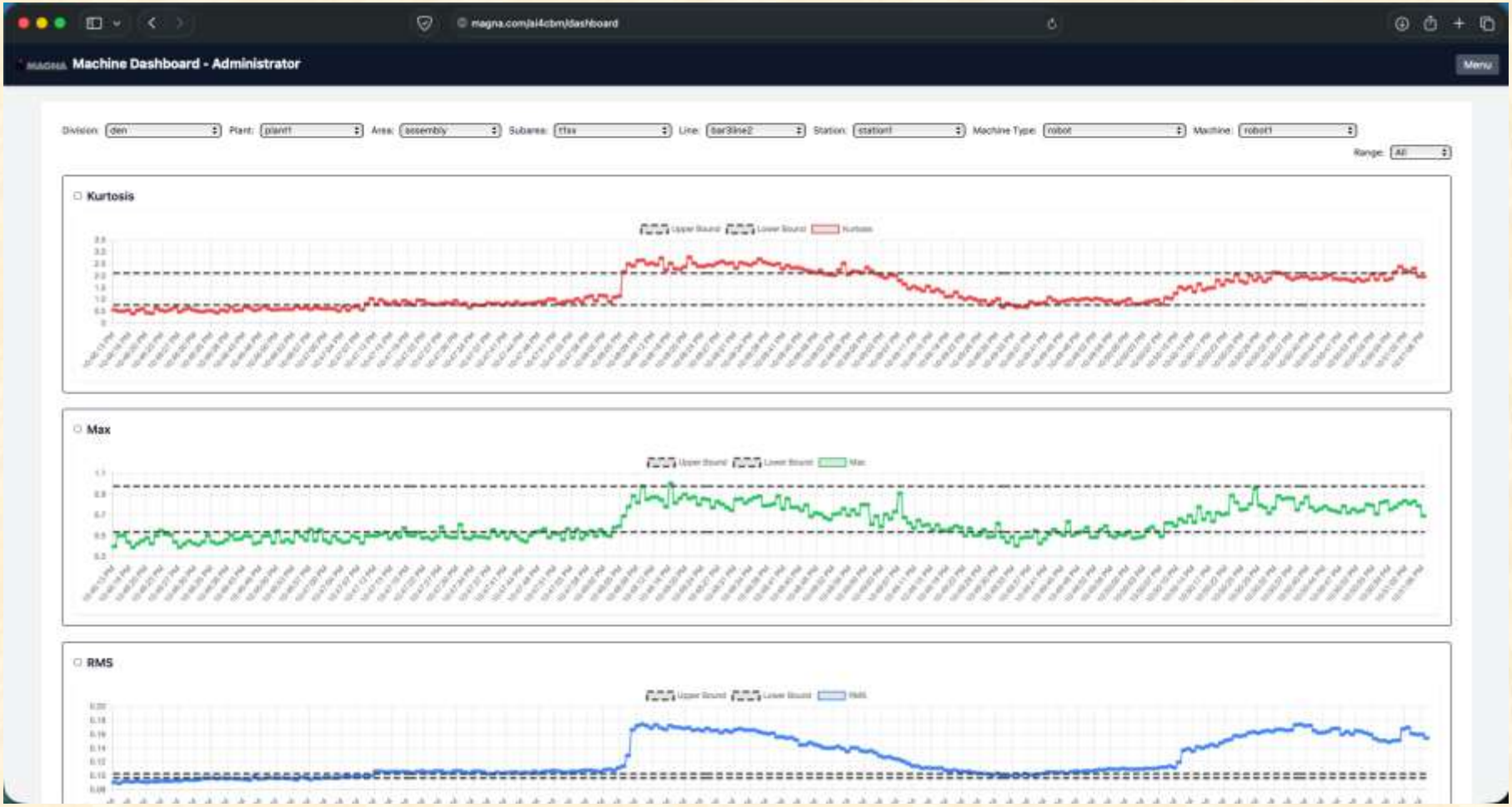
Technical Tasks Completed

- Team Member Daniel Chen
  - Dataset Selection
  - Training Preprocessing
  - Model Architecture
  - Model Training
- Team Member Ethan Springer
  - MQTT Bridge Node
  - Admin Control Node/Handling
  - Model Deployment Node(in part)
  - Simulation/Testing Node
  - ROS2 Pipeline Adjustments throughout Development
- Team Member Athul Syam
  - Developed final Taskboard Page
  - Created initial Log Pages
  - Developed Notification signup/sending system
- Team Member Hector Dominguez Rojas
  - Developed final Dashboard Page
  - Developed final Log page
  - Developed feature selection and upper/lower bound on UI
  - Created initial Taskboard
- Team Member Liz Hanks
  - Postprocessing node (Formatting raw and model output data to store in db)
  - AWS server set up
  - Initial postgresql setup
  - Backend implementation
    - Mqtt subscriber node (Handling mqtt connections from frontend)
    - Backend node (Endpoints that query postgresql tables)
- Team Member Michael Gryn
  - Data preprocessing in pipeline
  - UI Backend endpoint support
  - Configuration file for pipeline instances

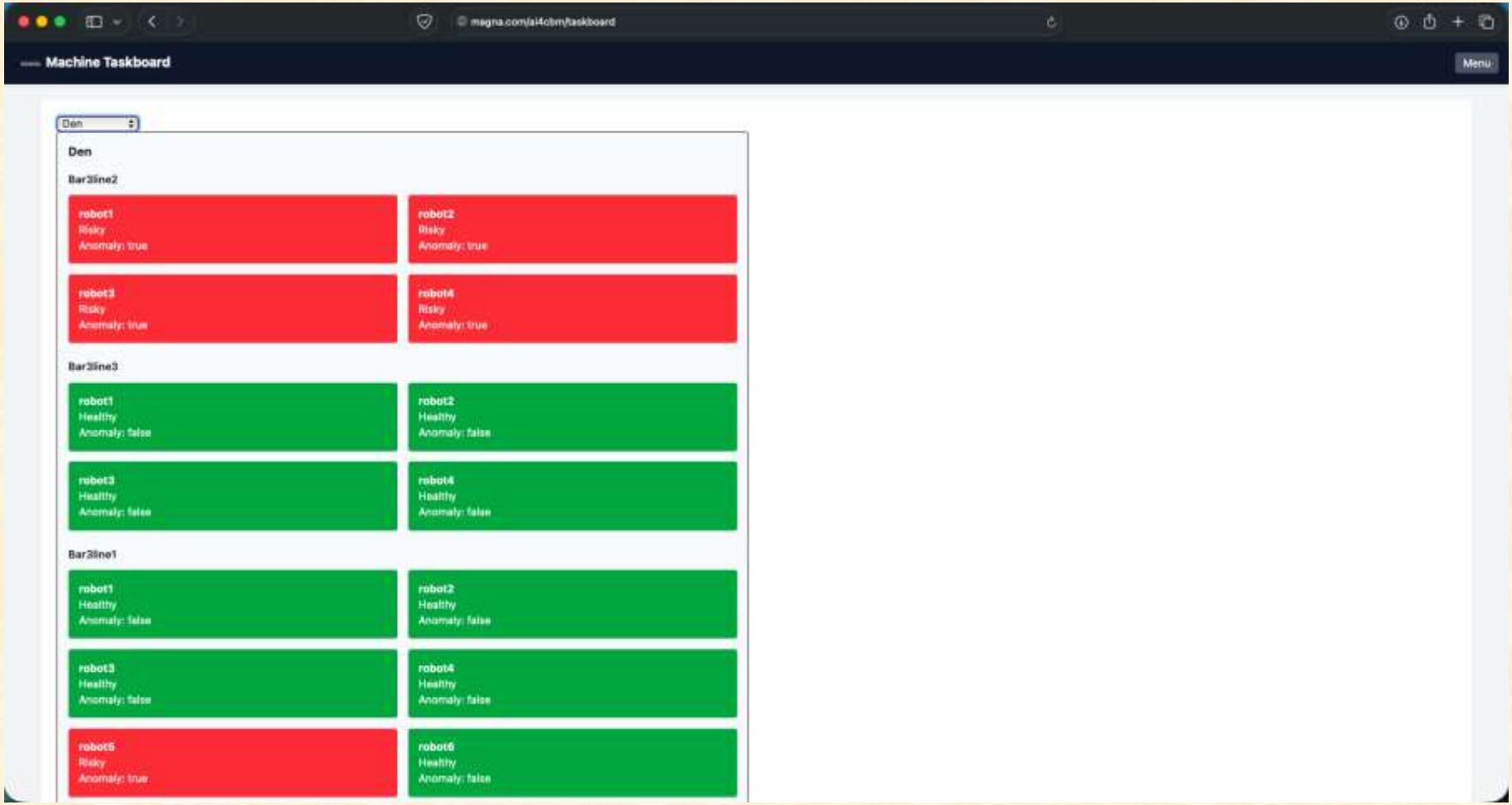
System Architecture



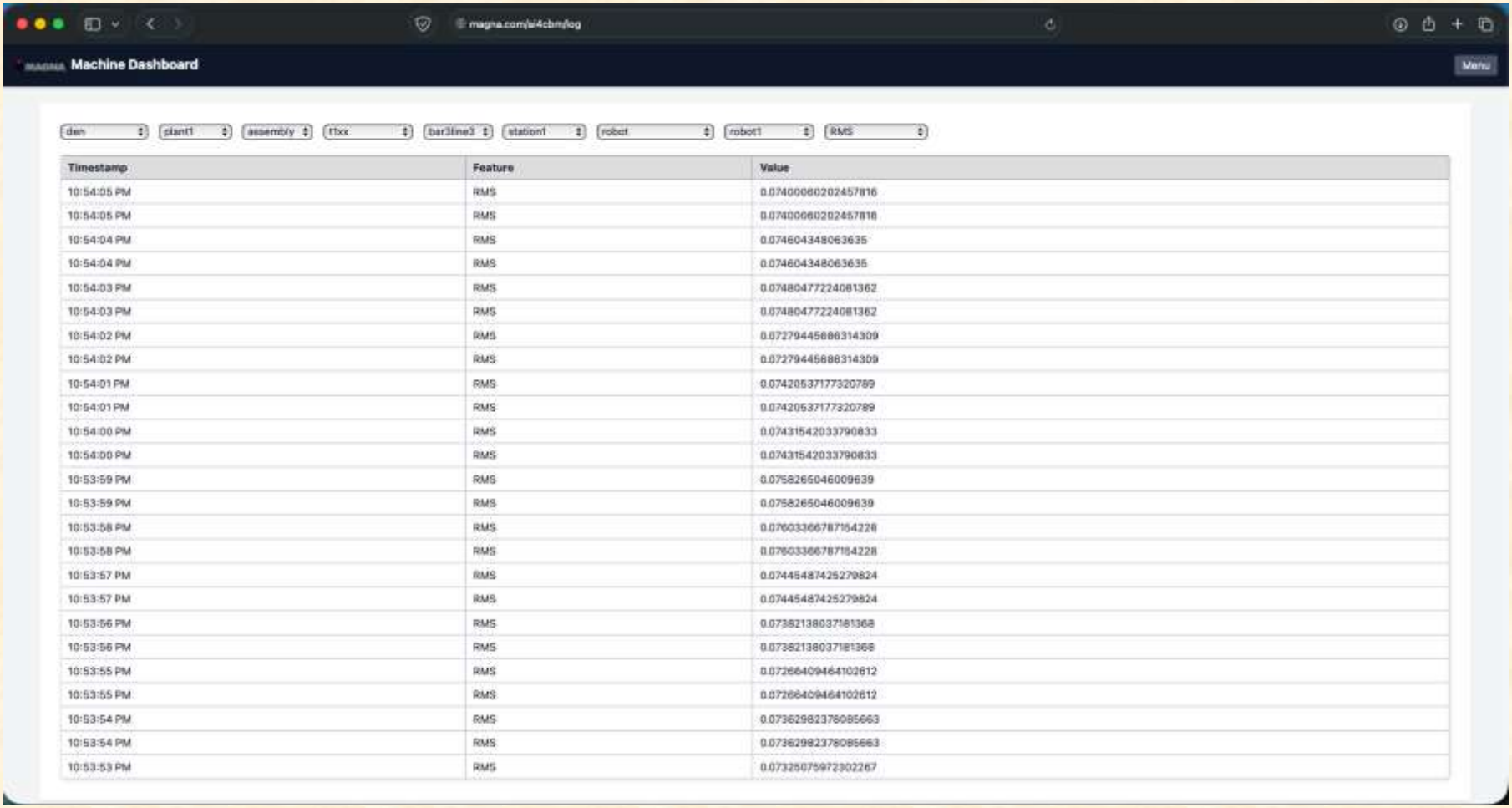
Dashboard


magna.com/ai4cbm/dashboard
Machine Dashboard - Administrator
Menu
Division: den
Plant: plant1
Area: assembly
Subarea: trax
Line: bar3line2
Station: station1
Machine Type: robot
Machine: robot1
Range: All
Kurtosis
Upper Bound
Lower Bound
Kurtosis
Max
Upper Bound
Lower Bound
Max
RMS
Upper Bound
Lower Bound
RMS

Taskboard

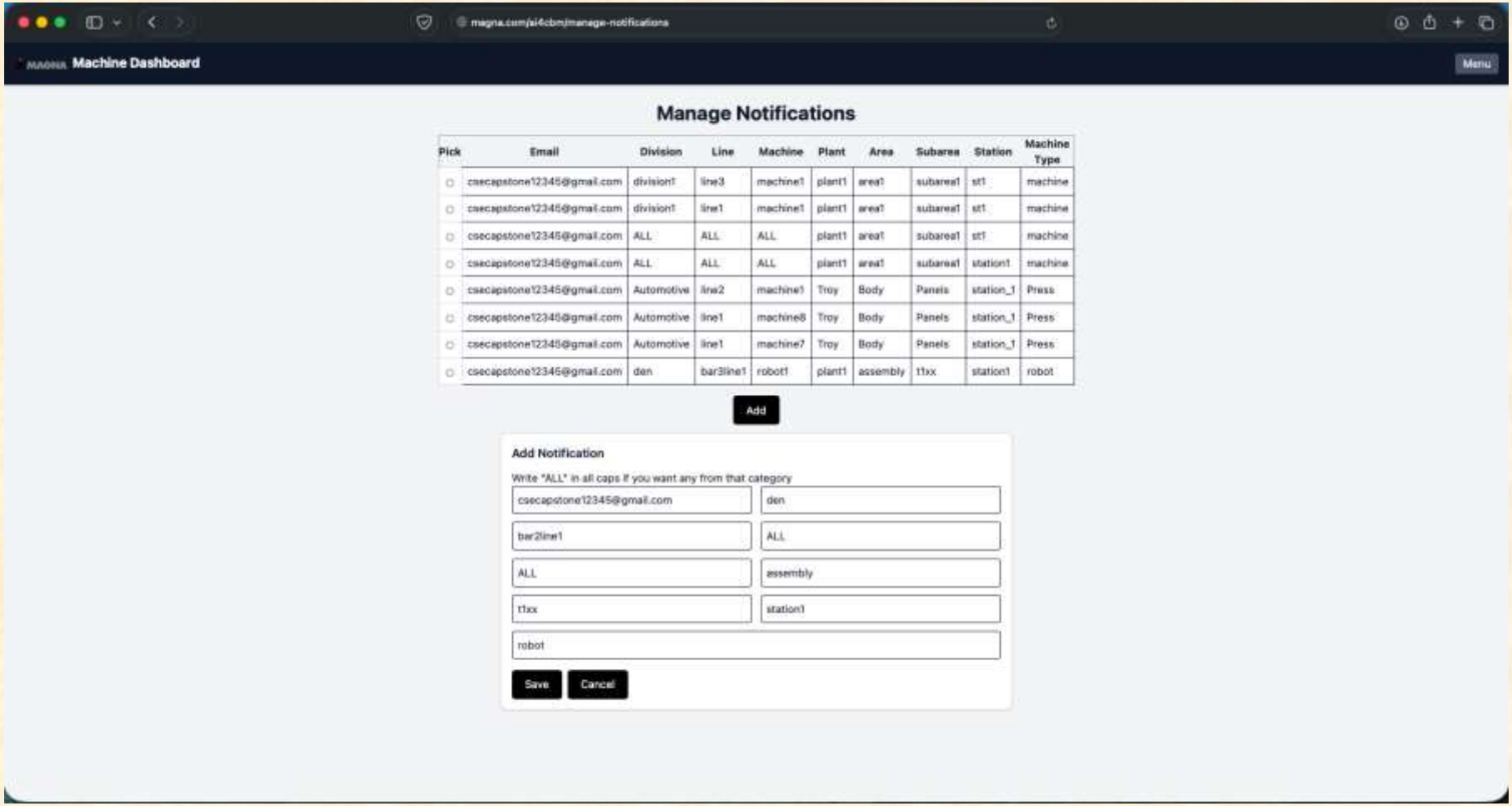

magna.com/ai4cbm/taskboard
Machine Taskboard
Menu
Den
Den
Bar3line2
robot1
Risky
Anomaly: true
robot2
Risky
Anomaly: true
robot3
Risky
Anomaly: true
robot4
Risky
Anomaly: true
Bar3line3
robot1
Healthy
Anomaly: false
robot2
Healthy
Anomaly: false
robot3
Healthy
Anomaly: false
robot4
Healthy
Anomaly: false
Bar3line1
robot1
Healthy
Anomaly: false
robot2
Healthy
Anomaly: false
robot3
Healthy
Anomaly: false
robot4
Healthy
Anomaly: false
robot5
Risky
Anomaly: true
robot6
Healthy
Anomaly: false

Log



Timestamp	Feature	Value
10:54:05 PM	RMS	0.0740060202457816
10:54:05 PM	RMS	0.0740060202457816
10:54:04 PM	RMS	0.074604348063635
10:54:04 PM	RMS	0.074604348063635
10:54:03 PM	RMS	0.07480477224081362
10:54:03 PM	RMS	0.07480477224081362
10:54:02 PM	RMS	0.07279445886314309
10:54:02 PM	RMS	0.07279445886314309
10:54:01 PM	RMS	0.07420537177320789
10:54:01 PM	RMS	0.07420537177320789
10:54:00 PM	RMS	0.07431542033790833
10:54:00 PM	RMS	0.07431542033790833
10:53:59 PM	RMS	0.0758265046009639
10:53:59 PM	RMS	0.0758265046009639
10:53:58 PM	RMS	0.07603366787154228
10:53:58 PM	RMS	0.07603366787154228
10:53:57 PM	RMS	0.07445487425279824
10:53:57 PM	RMS	0.07445487425279824
10:53:56 PM	RMS	0.0738213803718136
10:53:56 PM	RMS	0.0738213803718136
10:53:55 PM	RMS	0.07266409464102612
10:53:55 PM	RMS	0.07266409464102612
10:53:54 PM	RMS	0.07362982378085663
10:53:54 PM	RMS	0.07362982378085663
10:53:53 PM	RMS	0.07325075972302267

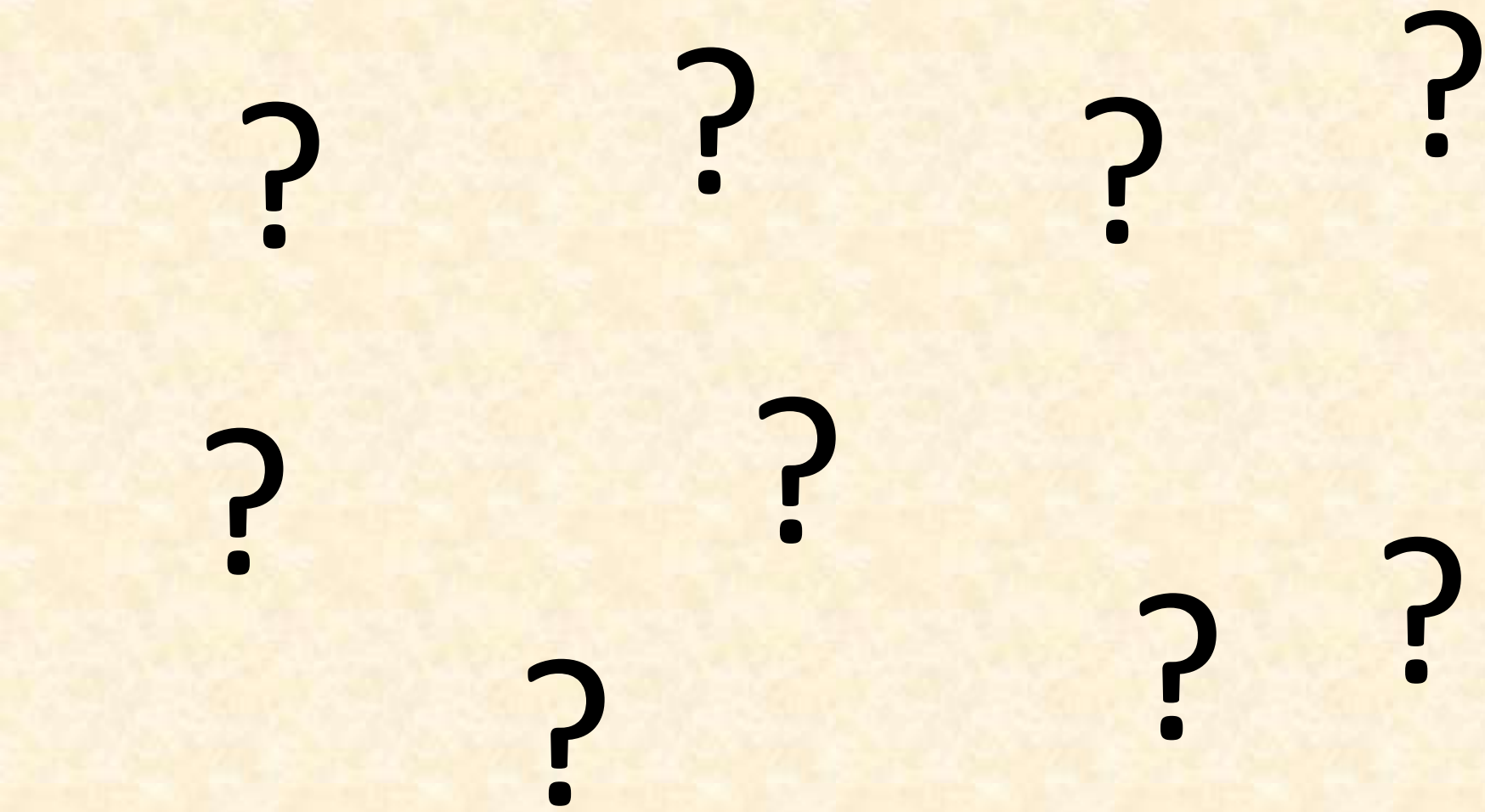
Manage Notifications



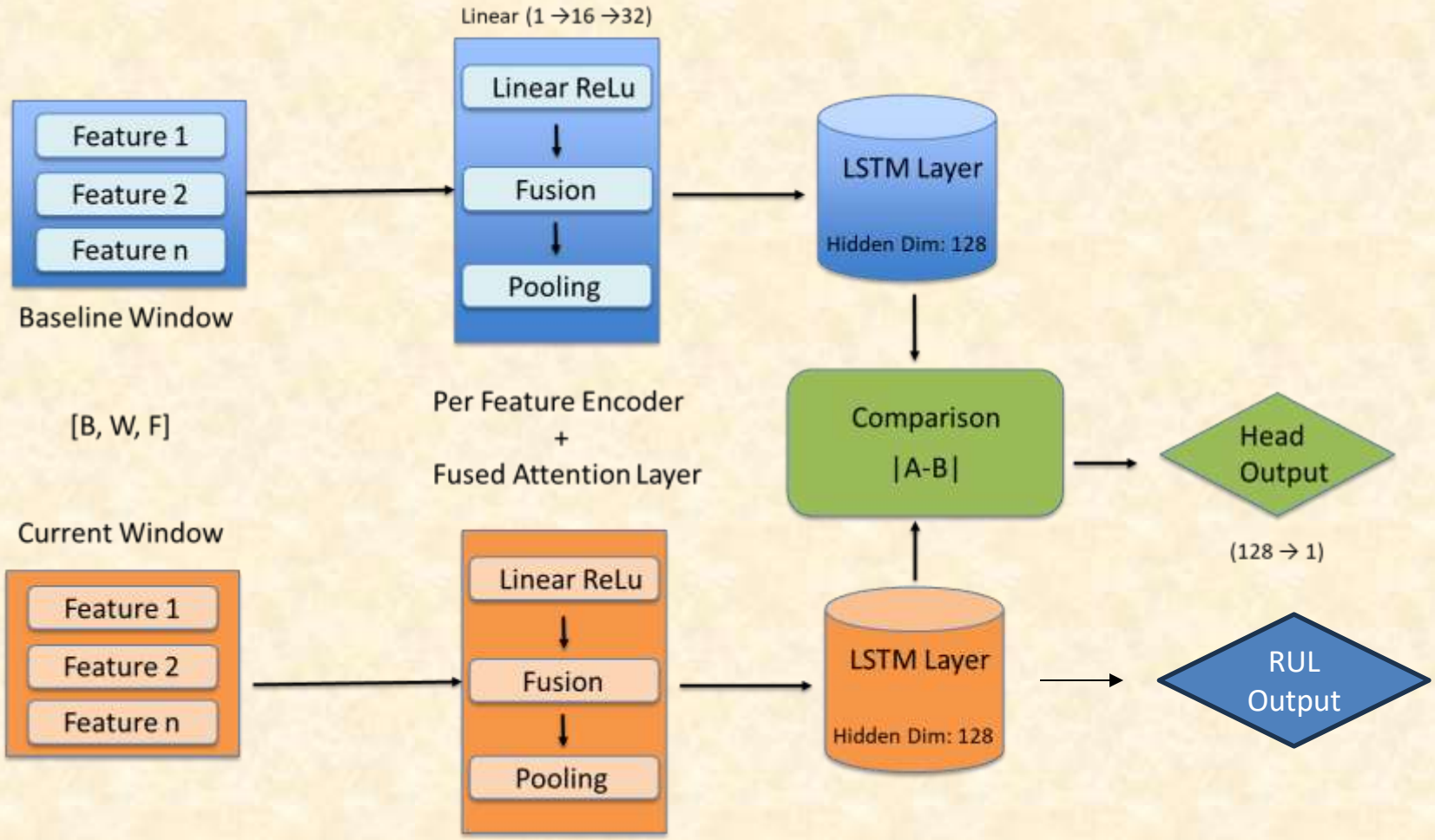
What’s left to do?

- Other Tasks
  - Migrate drop downs to a pop-up menu
  - Backend operation performance improvements

Questions?



Model Architecture



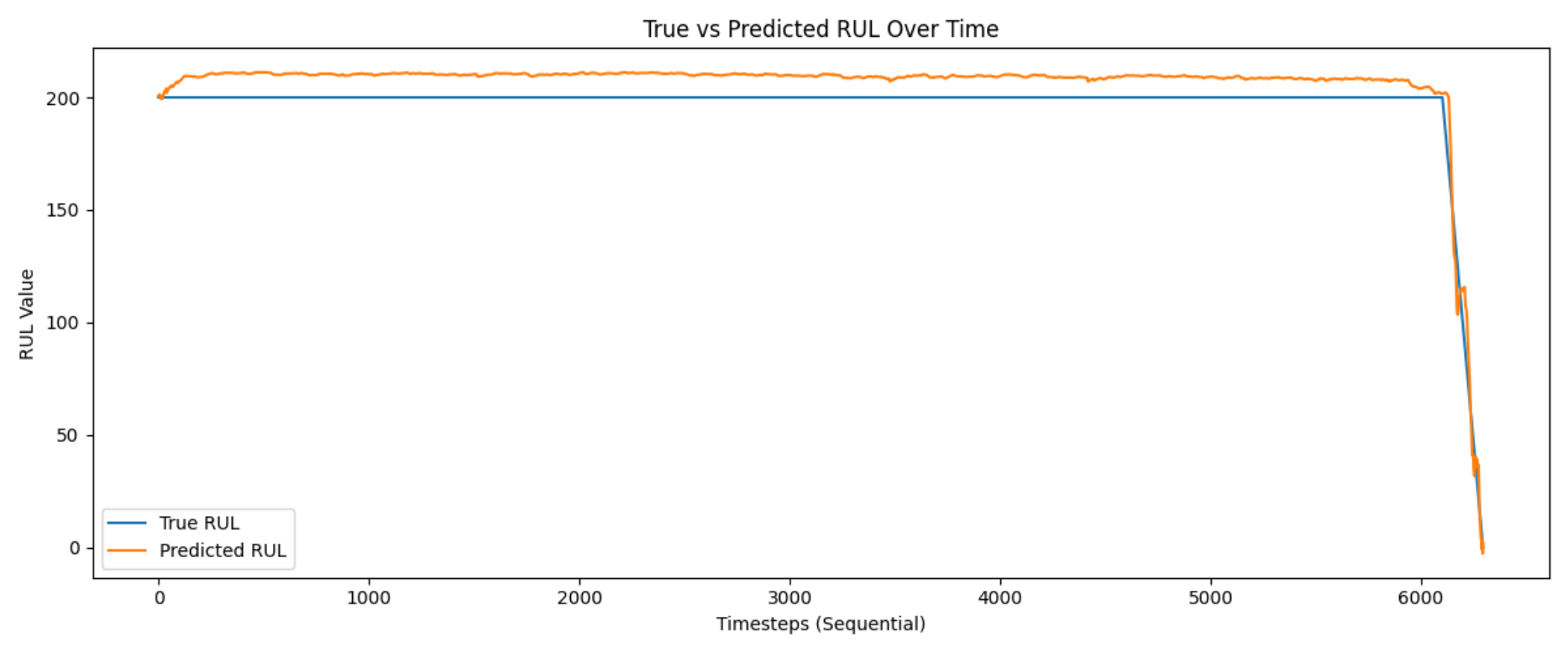
Binary Task Test Results

- Acc: 0.99, F1: 0.91, Precision: 1.00, Recall: 0.83

	Predicted Normal (0)	**Predicted Faulty (1)**
Actual Normal (0)	6101	0
Actual Faulty (1)	33	161

Regression Task Test Results

- RUL Task -> MAE: 15.48, RMSE: 15.76



Model Parameters

- LSTM:

- W_ih (input-to-hidden): shape (4*H, I)
- W_hh (hidden-to-hidden): shape (4*H, H)
- bias_ih: shape (4*H,)
- bias_hh: shape (4*H,)
- Per layer params = 4*H*I + 4*H*H + 8*H

- Layer 1: I=32, H=128
- 4*128*32 = 16,384
- 4*128*128 = 65,536
- 8*128 = 1,024
- Layer1 total = 82,944

- Layer 2: I=128, H=128
- 4*128*128 = 65,536
- 4*128*128 = 65,536
- 8*128 = 1,024
- Layer2 total = 132,096

Model Parameters

- Feature encoders (per feature): two linear layers: (1×16 +16) + (16×32 +32) = 576. With 4 features → 2,304.

- Fused Attention (embed_dim=32, heads=4):
- Q/K/V projections: 3 * 32 * 32 = 3,072
- output projection: 32 * 32 = 1,024
- biases: 128
- → 4,224

- Head Linear(128,1): 129