



Beta Presentation

ML/AI Pipeline For Condition-Based Maintenance

The Capstone Experience

Team Magna AI4CBM

Michael Gryn

Daniel Chen

Hector Dominguez 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

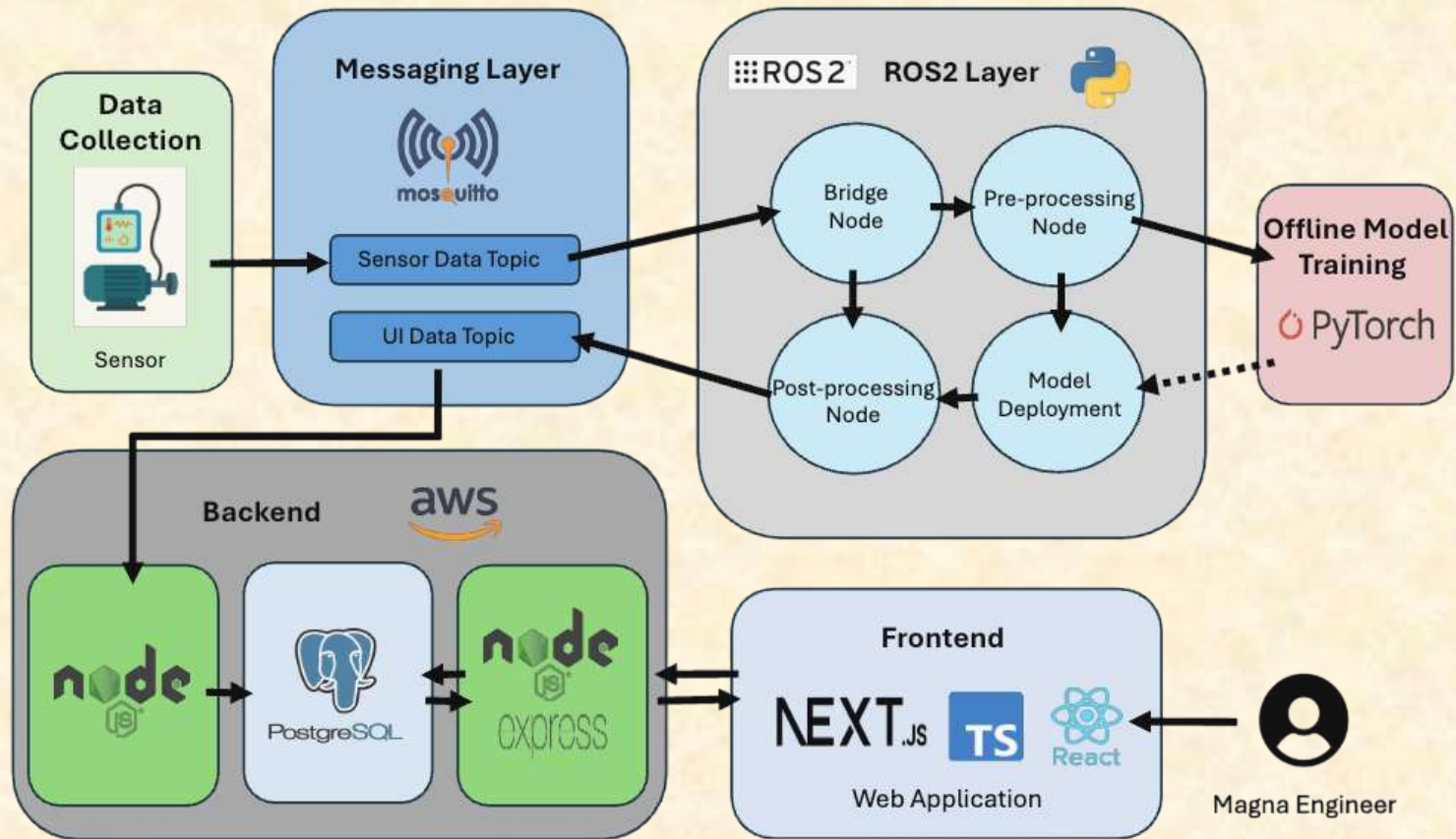
- Team Member Daniel Chen
 - Dataset Selection
 - Training Preprocessing
 - Model Architecture
 - Model Training
- Team Member Ethan Springer
 - MQTT Bridge Node
 - Admin Control Node/Handling
 - Model Deployment Node(in part)
 - Simulation/Testing Node
 - ROS2 Pipeline Adjustments throughout Development
- 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

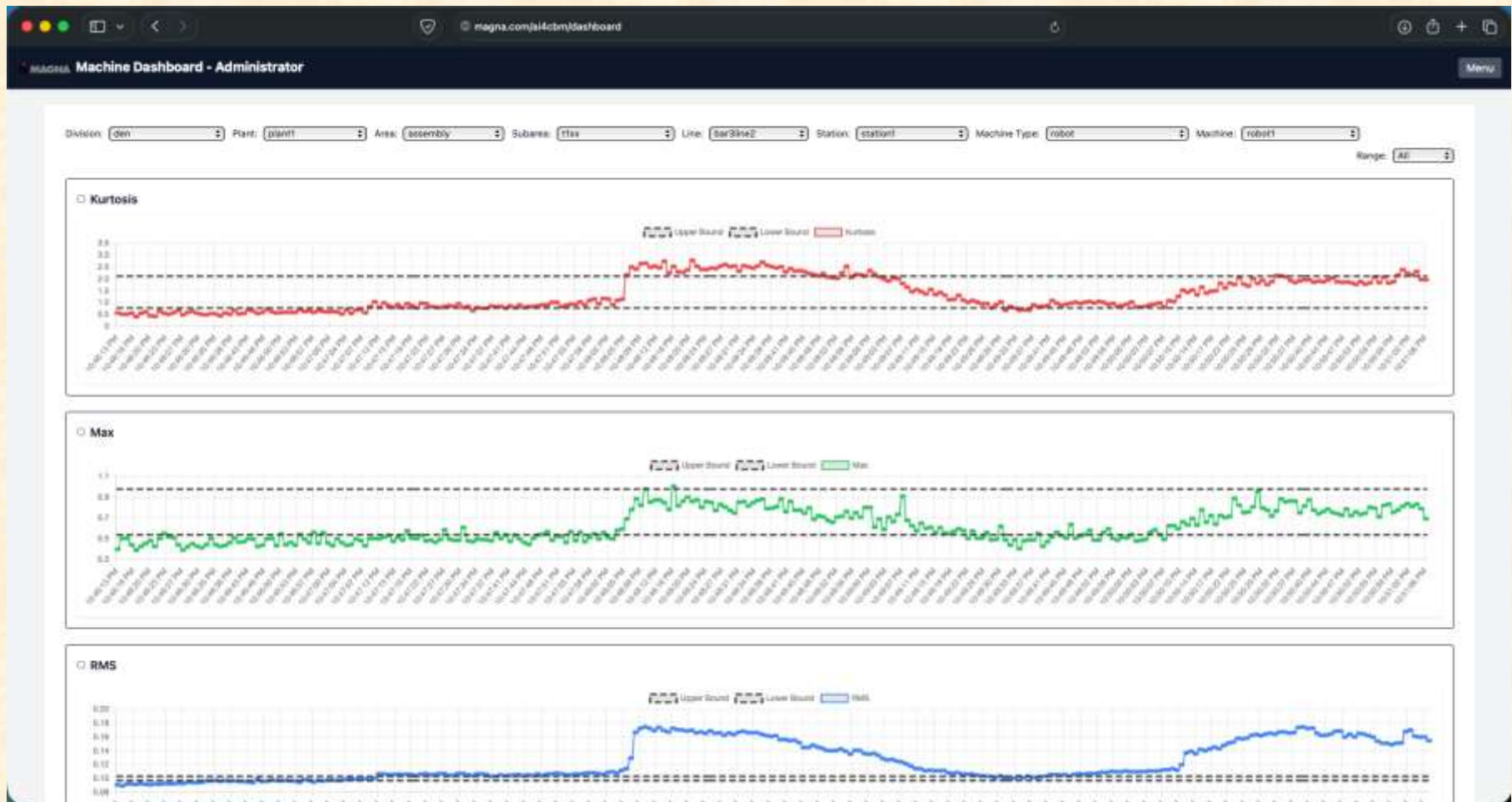
- Team Member Daniel Chen
 - Dataset Selection
 - Training Preprocessing
 - Model Architecture
 - Model Training
- Team Member Ethan Springer
 - MQTT Bridge Node
 - Admin Control Node/Handling
 - Model Deployment Node(in part)
 - Simulation/Testing Node
 - ROS2 Pipeline Adjustments throughout Development
- 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



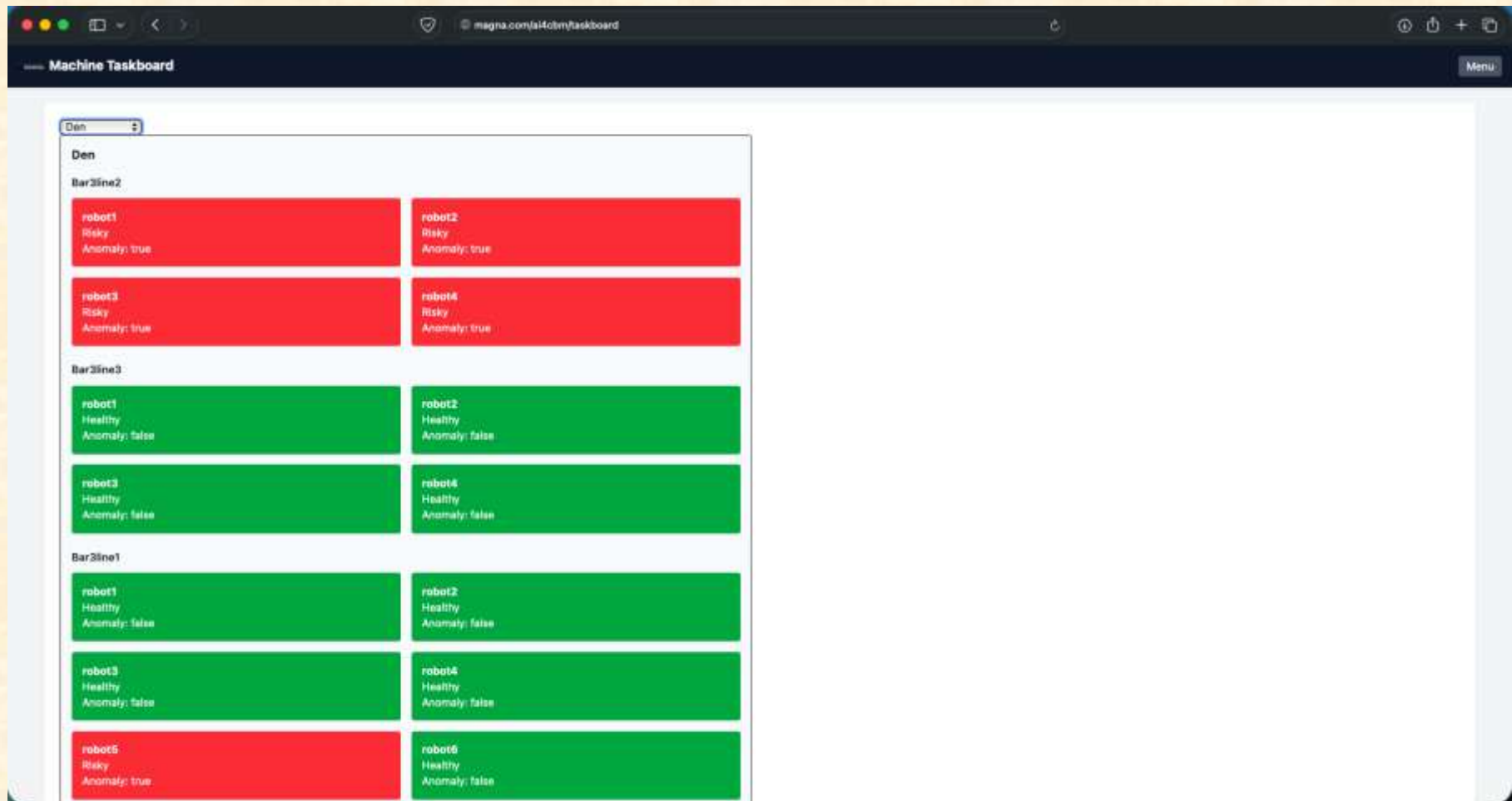
System Architecture



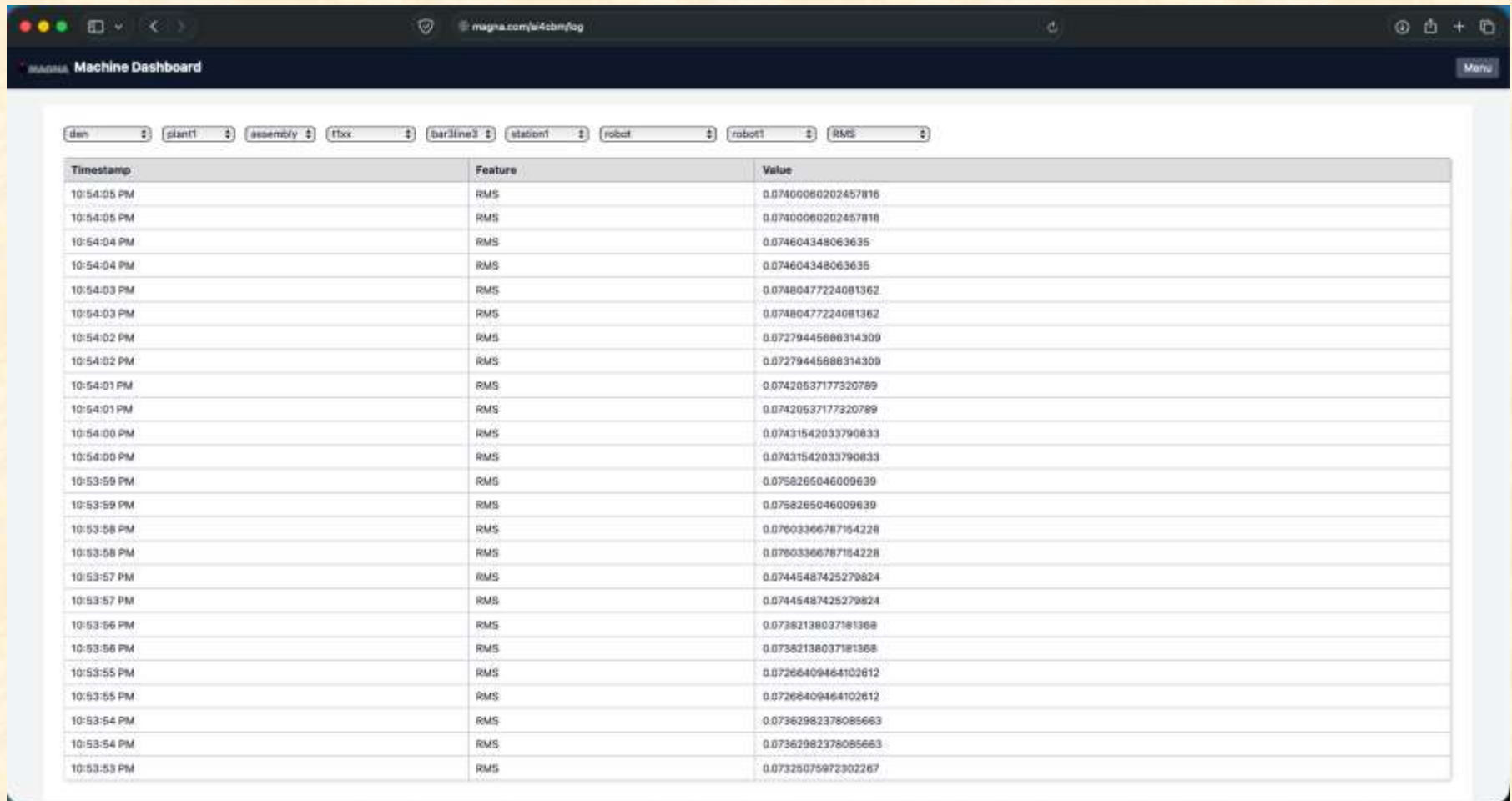
Dashboard



Taskboard



Log



The screenshot displays the 'Magna Machine Dashboard' in a web browser. The dashboard includes a header with the Magna logo and a 'Menu' button. Below the header, there are several filter buttons: 'den', 'plant1', 'assembly', 'line', 'bar3line3', 'station1', 'robot', 'robot1', and 'RMS'. The main content area is a table with three columns: 'Timestamp', 'Feature', and 'Value'. The table contains 25 rows of data, all showing 'RMS' as the feature and numerical values as the data points. The values range from approximately 0.0732 to 0.0746.

Timestamp	Feature	Value
10:54:05 PM	RMS	0.07400060202457816
10:54:05 PM	RMS	0.07400060202457816
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	0.07279445688314309
10:54:02 PM	RMS	0.07279445688314309
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	0.07382138037181368
10:53:56 PM	RMS	0.07382138037181368
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

The screenshot shows a web browser window with the URL `magna.com/si4cbm/manage-notifications`. The page title is "Machine Dashboard" and there is a "Menu" button in the top right. The main content area is titled "Manage Notifications" and contains a table with 10 columns: Pick, Email, Division, Line, Machine, Plant, Area, Subarea, Station, and Machine Type. There are 8 rows of data, each with a radio button in the "Pick" column. Below the table is an "Add" button. Below the "Add" button is a modal form titled "Add Notification". The modal form has a text input field with the value "csecapstone12345@gmail.com" and a dropdown menu with the value "den". Below these are four more input fields: "bar2line1", "ALL", "assembly", and "station1". At the bottom of the modal are "Save" and "Cancel" buttons.

Pick	Email	Division	Line	Machine	Plant	Area	Subarea	Station	Machine Type
☐	csecapstone12345@gmail.com	division1	line3	machine1	plant1	area1	subarea1	st1	machine
☐	csecapstone12345@gmail.com	division1	line1	machine1	plant1	area1	subarea1	st1	machine
☐	csecapstone12345@gmail.com	ALL	ALL	ALL	plant1	area1	subarea1	st1	machine
☐	csecapstone12345@gmail.com	ALL	ALL	ALL	plant1	area1	subarea1	station1	machine
☐	csecapstone12345@gmail.com	Automotive	line2	machine1	Troy	Body	Panels	station_1	Press
☐	csecapstone12345@gmail.com	Automotive	line1	machine8	Troy	Body	Panels	station_1	Press
☐	csecapstone12345@gmail.com	Automotive	line1	machine7	Troy	Body	Panels	station_1	Press
☐	csecapstone12345@gmail.com	den	bar3line1	robot1	plant1	assembly	tlxx	station1	robot

Add

Add Notification

Write "ALL" in all caps if you want any from that category

Save **Cancel**



What's left to do?

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

Questions?

?

?

?

?

?

?

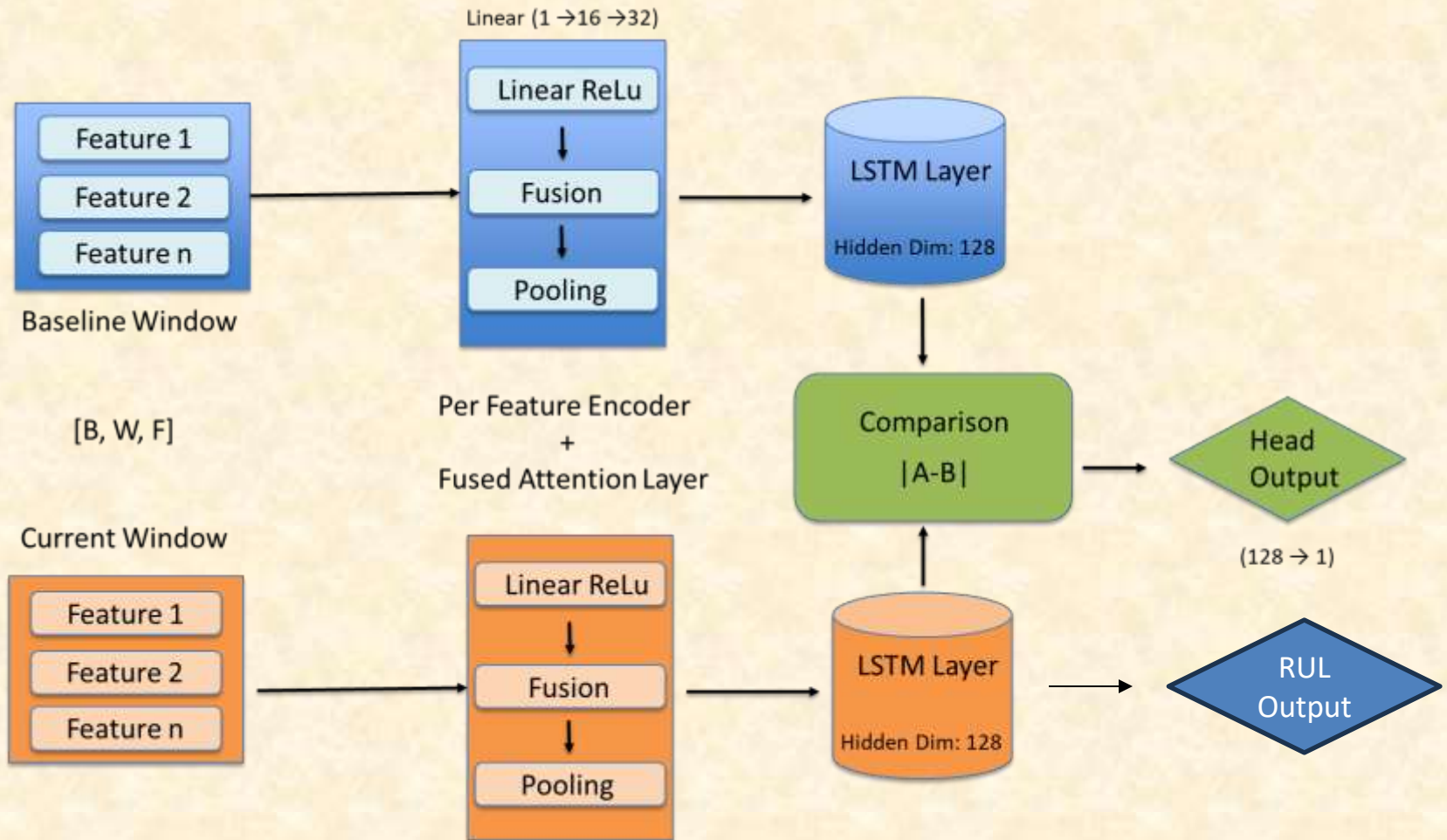
?

?

?



Model Architecture



Binary Task Test Results

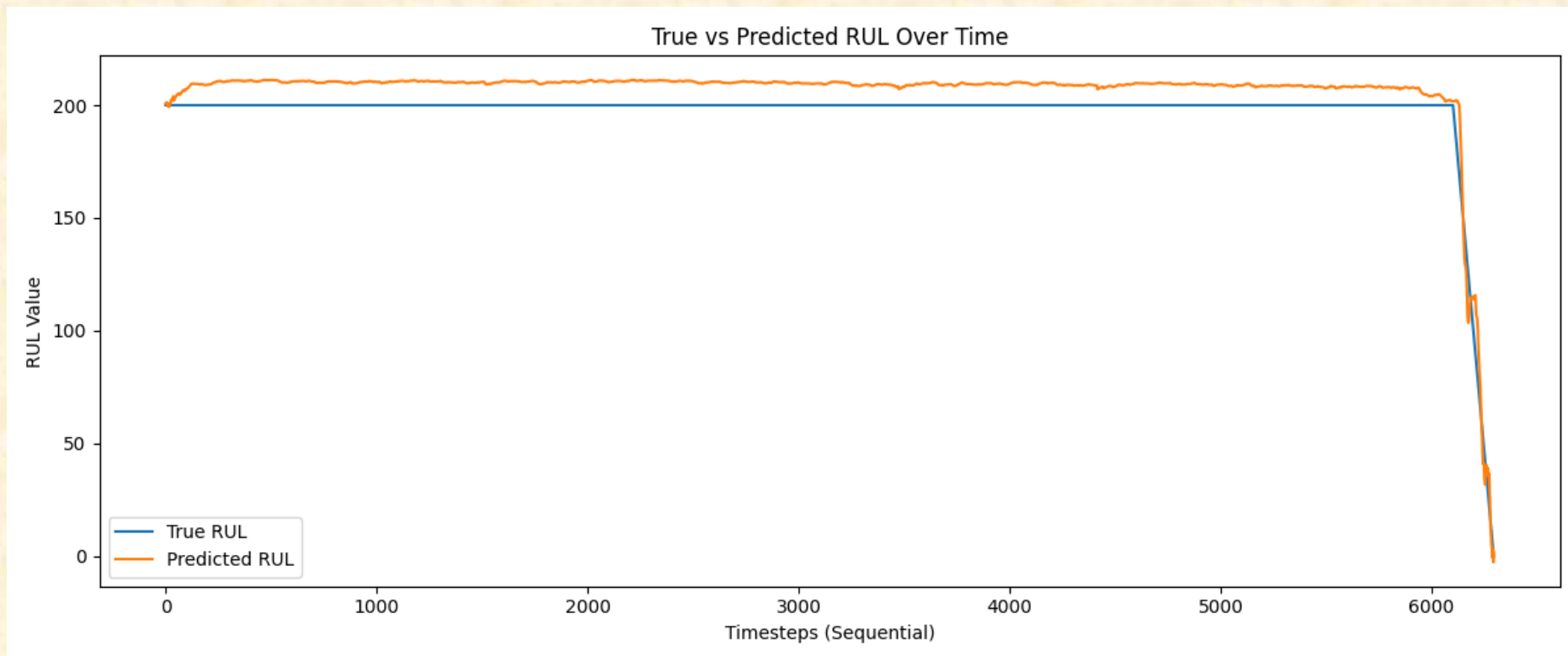
- Acc: 0.99, F1: 0.91, Precision: 1.00, Recall: 0.83

	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 $(4 \times H, I)$
 - W_{hh} (hidden-to-hidden): shape $(4 \times H, H)$
 - $bias_{ih}$: shape $(4 \times H,)$
 - $bias_{hh}$: shape $(4 \times H,)$
 - Per layer params = $4 \times H \times I + 4 \times H \times H + 8 \times H$
- Layer 1: $I=32, H=128$
 - $4 \times 128 \times 32 = 16,384$
 - $4 \times 128 \times 128 = 65,536$
 - $8 \times 128 = 1,024$
 - Layer1 total = 82,944
- Layer 2: $I=128, H=128$
 - $4 \times 128 \times 128 = 65,536$
 - $4 \times 128 \times 128 = 65,536$
 - $8 \times 128 = 1,024$
 - Layer2 total = 132,096



Model Parameters

- Feature encoders (per feature): two linear layers: $(1 \times 16 + 16) + (16 \times 32 + 32) = 576$. With 4 features $\rightarrow 2,304$.
- Fused Attention (embed_dim=32, heads=4):
 - Q/K/V projections: $3 * 32 * 32 = 3,072$
 - output projection: $32 * 32 = 1,024$
 - biases: 128
 - $\rightarrow 4,224$
- Head Linear(128,1): 129

