PrognosAI: AI-Driven Predictive Maintenance System Using Time-Series Sensor Data

Agenda

The project focuses on predicting the Remaining Useful Life (RUL) of industrial machinery like turbines and pumps using multivariate time-series sensor data and LSTM deep learning models. This enables proactive maintenance, reducing unplanned downtime and optimizing asset utilization. The workflow includes data ingestion, feature engineering, model training, risk thresholding, and interactive dashboards for monitoring. The project showcases the effective application of AI-driven pattern recognition for real-world predictive maintenance in critical industrial settings..

Milestone 1: Data Preparation and Feature Engineering

Objective

- Load, clean, and preprocess the NASA CMAPSS sensor dataset.
- Generate rolling window sequences of sensor readings.
- Compute Remaining Useful Life (RUL) targets for each engine cycle.

Key Steps

Data Loading & Cleaning:

- Imported raw sensor data CSV and identified the time or cycle column.
- Handled missing values using median imputation for numerical sensor features.

RUL Calculation:

- Created overlapping sequences (window size of 50 cycles) sliding by stride 1.
- Extracted sensor features and label (RUL) for each sequence.

Batch Processing & Verification:

- Processed data in batches for efficient handling.
- Verified sensor value distributions and RUL ranges within batches.

Milestone 2: Model Development and Training

Objective

- Build and train a deep learning model to predict the Remaining Useful Life (RUL) from sensor data sequences.

Model Architecture:

Used a Sequential deep learning model with these layers:

- Two stacked LSTM layers (first with 128 units returning sequences, second with 64 units).
- Dropout layers (0.2 rate) after each LSTM for regularization.
- Dense layers: one hidden with 32 units (ReLU activation), output layer with 1 unit (linear activation) for RUL prediction.

Training Details:

- Loss function: Mean Absolute Error (MAE).
- Optimizer: Adam with learning rate 0.001.
- Metrics: Mean Squared Error (MSE).
- Training data split: 80% train, 20% validation.
- Batch size: 64.
- Epochs: Up to 100 with early stopping patience of 10 (restores best weights).
- Learning rate reduction on plateau (factor 0.5, patience 5).

Workflow:

- Preprocessed sequences (from Milestone 1) fed into the model.
- Model trained with callbacks for adaptive learning and early stopping.
- Evaluated performance with validation MAE and MSE.
- Output predictions compared visually against actual RUL values.

Milestone 3: Model Evaluation and Performance Assessment

Objective:

- Assess prediction accuracy of the trained model on the test dataset.

Key Metrics:

- Root Mean Square Error (RMSE): 15.9971 cycles.
- Mean Absolute Error (MAE): 13.1381 cycles.
- R^2 Score: 0.6958 (explains nearly 70% variance in RUL).

Evaluation Process:

- Generated predictions on the held-out test set.
- Compared predicted vs actual RUL with scatter plots and time series visualization
- Analyzed prediction error distribution showing mean error and standard deviation
- Checked for overfitting by comparing training and validation losses across epochs

Outcome:

- Model demonstrates good predictive performance with acceptable error margins.
- Provides reliable sequential RUL estimations for maintenance scheduling decisions.

Milestone 4: Risk Thresholding and Maintenance Alerts

Objective:

- Define thresholds for RUL values to trigger warning and critical maintenance alerts.

Alert Thresholds:

- WARNING threshold: 40 cycles.
- CRITICAL threshold: 20 cycles.

Alert System Workflow:

- Predicted RUL values classified into normal, warning, or critical alert levels.
- Monitored distribution of alerts on test samples.
- Visualized alert frequency with bar charts and RUL distribution with threshold markers.

Alert Performance:

- Normal operation detected in ~60.5% of predictions.
- No warning alert triggered (0%).
- Critical alerts triggered in ~39.5% of predictions.

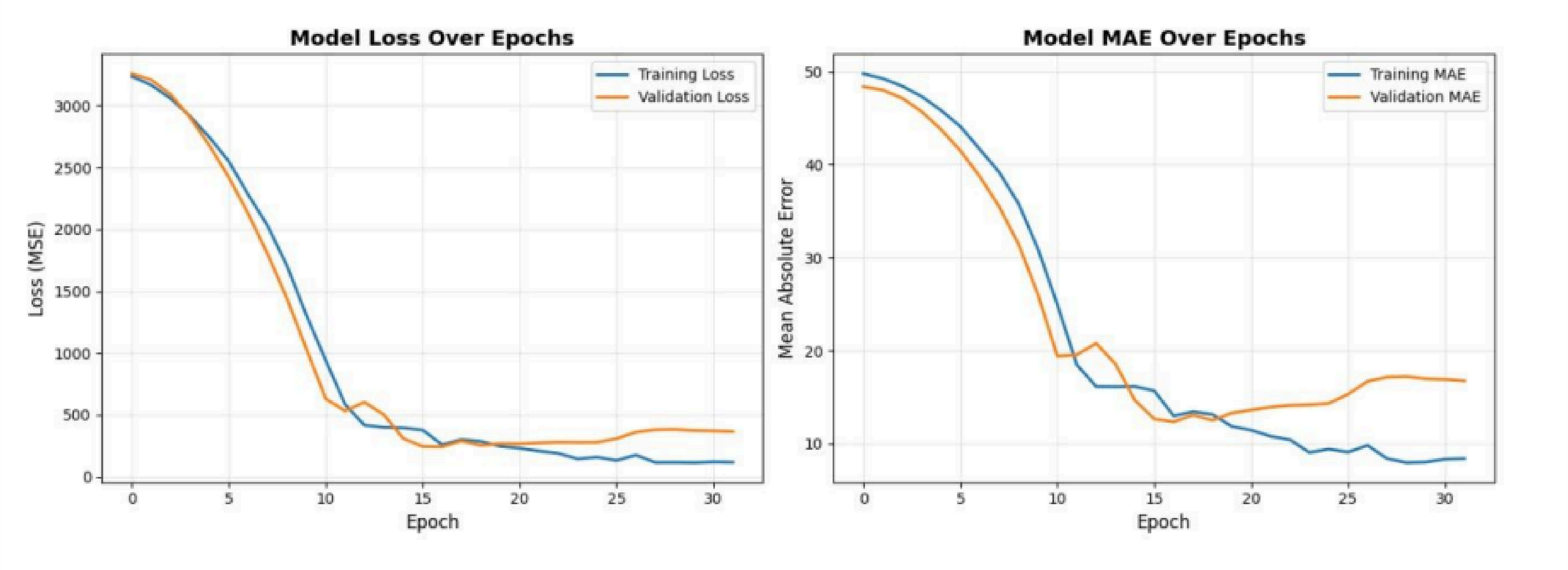
Outcome:

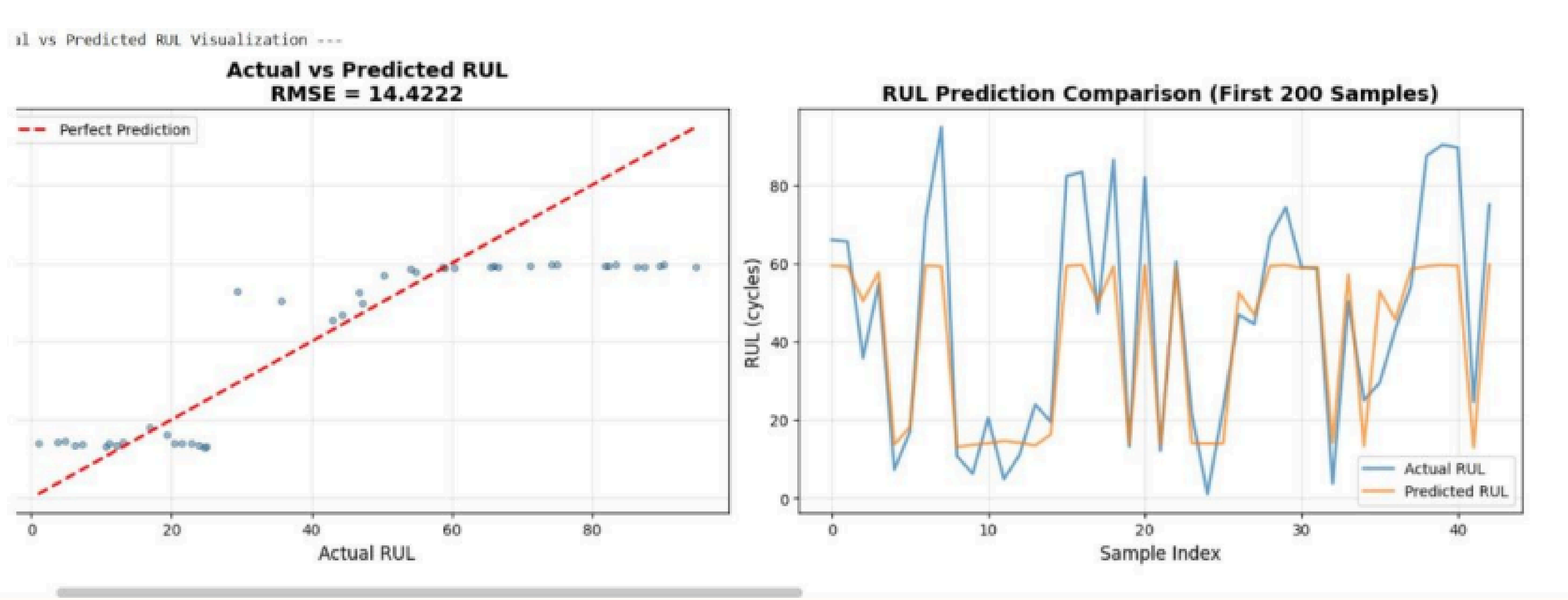
- Alerting mechanism enables proactive maintenance planning.
- Highlights critical machinery components approaching failure margins.

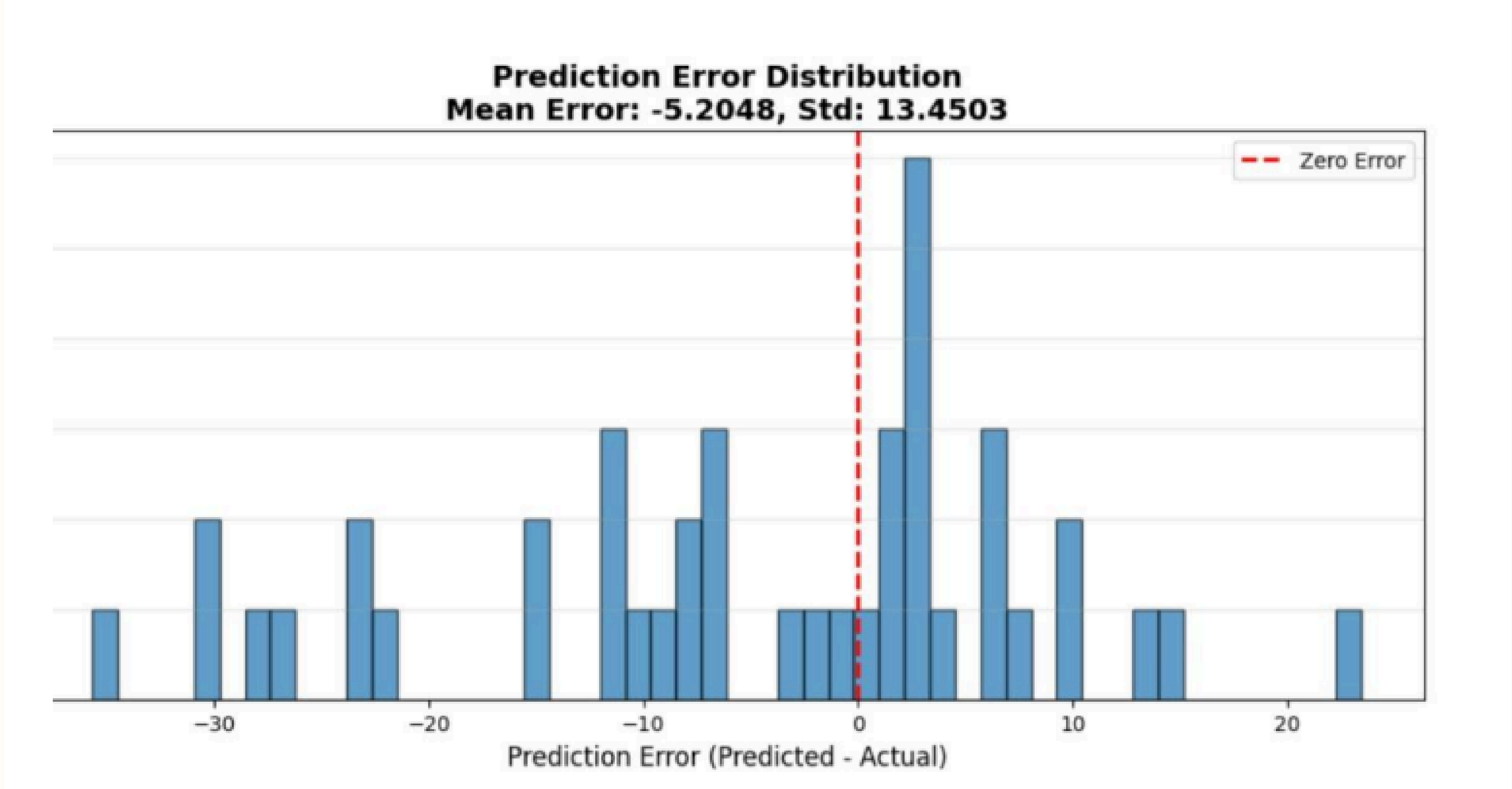
Output for Milestone 3 and 4:

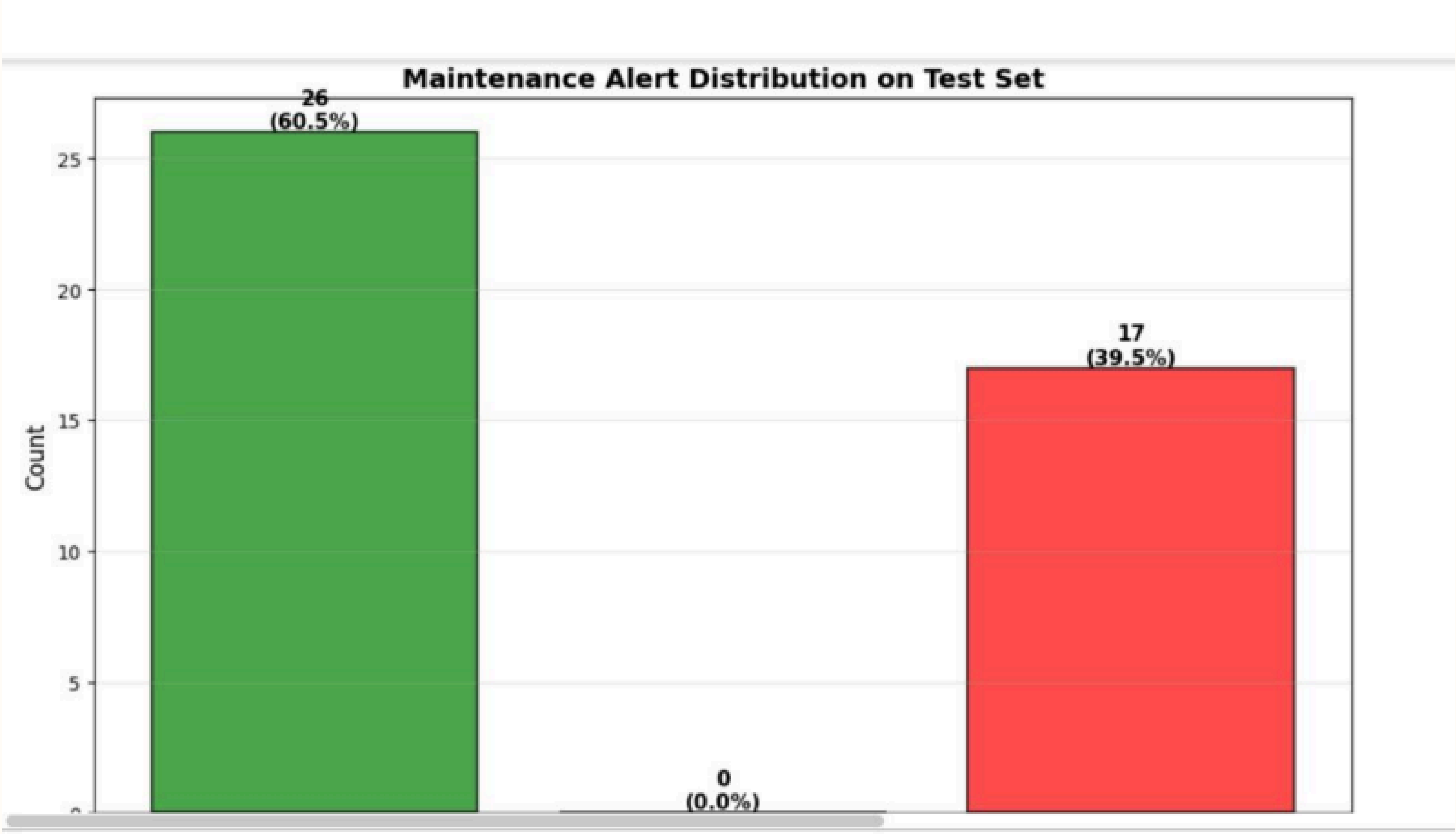
Model: "sequential"

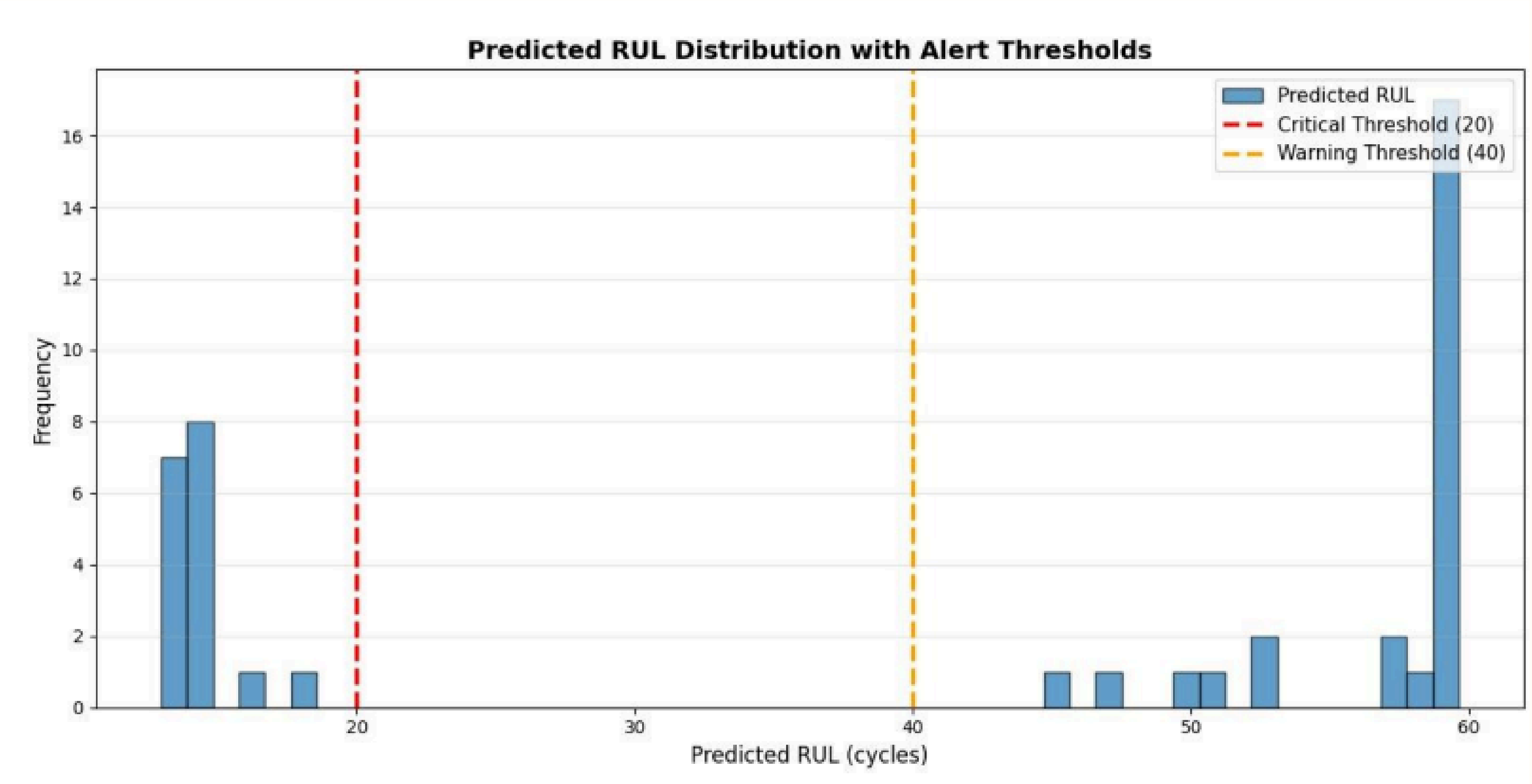
Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 30, 64)	640
batch_normalization (BatchNormalization)	(None, 30, 64)	256
dropout (Dropout)	(None, 30, 64)	0
conv1d_1 (Conv1D)	(None, 30, 32)	6,176
batch_normalization_1 (BatchNormalization)	(None, 30, 32)	128
dropout_1 (Dropout)	(None, 30, 32)	0
bidirectional (Bidirectional)	(None, 30, 128)	49,664
dropout_2 (Dropout)	(None, 30, 128)	0
bidirectional_1 (Bidirectional)	(None, 64)	41,216
dropout_3 (Dropout)	(None, 64)	0
dense (Dense)	(None, 64)	4,160
dropout_4 (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 32)	2,080
dense_2 (Dense)	(None, 1)	33






Prediction Error Distribution
Mean Error: -5.2048, Std: 13.4503
Zero Error
−30
−20
−10
0
10
20
Prediction Error (Predicted - Actual)


Maintenance Alert Distribution on Test Set
Count
26
(60.5%)
0
(0.0%)
17
(39.5%)
25
20
15
10
5


Predicted RUL Distribution with Alert Thresholds
Predicted RUL
Critical Threshold (20)
Warning Threshold (40)
Frequency
Predicted RUL (cycles)
0
2
4
6
8
10
12
14
16
20
30
40
50
60

Milestone 5: Visualization Dashboard Development

Objective:

- Develop an interactive dashboard to visualize RUL predictions, alerts, and model performance.
- Provide an intuitive user interface for monitoring machine health and maintenance decisions.

Features:

- Display RUL trends over time with interactive charts.
- Show alert zones categorized by Normal, Warning, and Critical levels based on thresholding.
- Visualization of model evaluation metrics including training loss, MAE, and validation performance.
- Provide alert distribution and confusion matrix heatmap to assess alert system accuracy.
- Ability to explore actual vs predicted RUL values with optional perfect prediction reference line.
- Sample maintenance alerts and full test predictions with filtering options.
- Additional reports on system performance, data quality, and model configuration

Technology Stack:

- Built with Streamlit for rapid and responsive web dashboard creation.
- Used Matplotlib and Seaborn for plotting data visualizations.
- Integrated data from CSV and text reports for comprehensive insights.

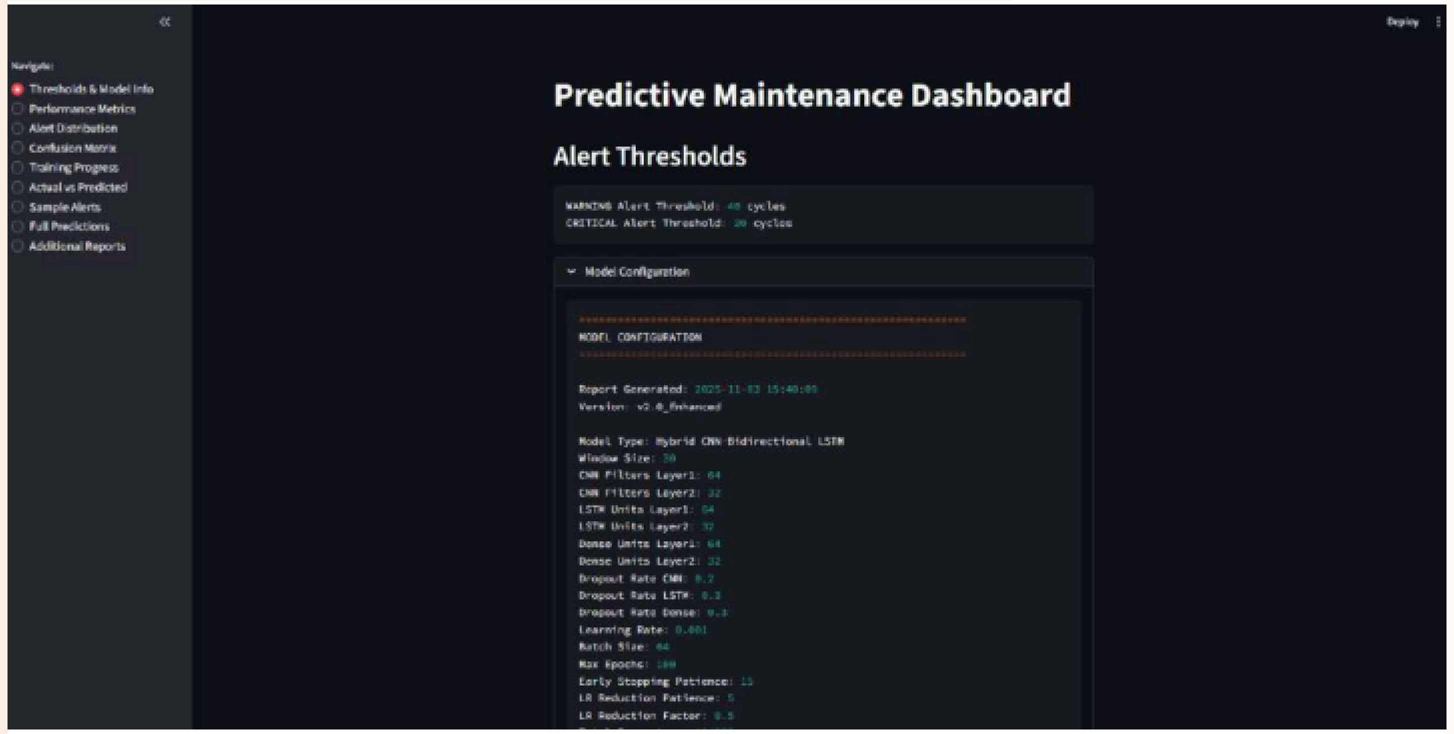
Deliverables:

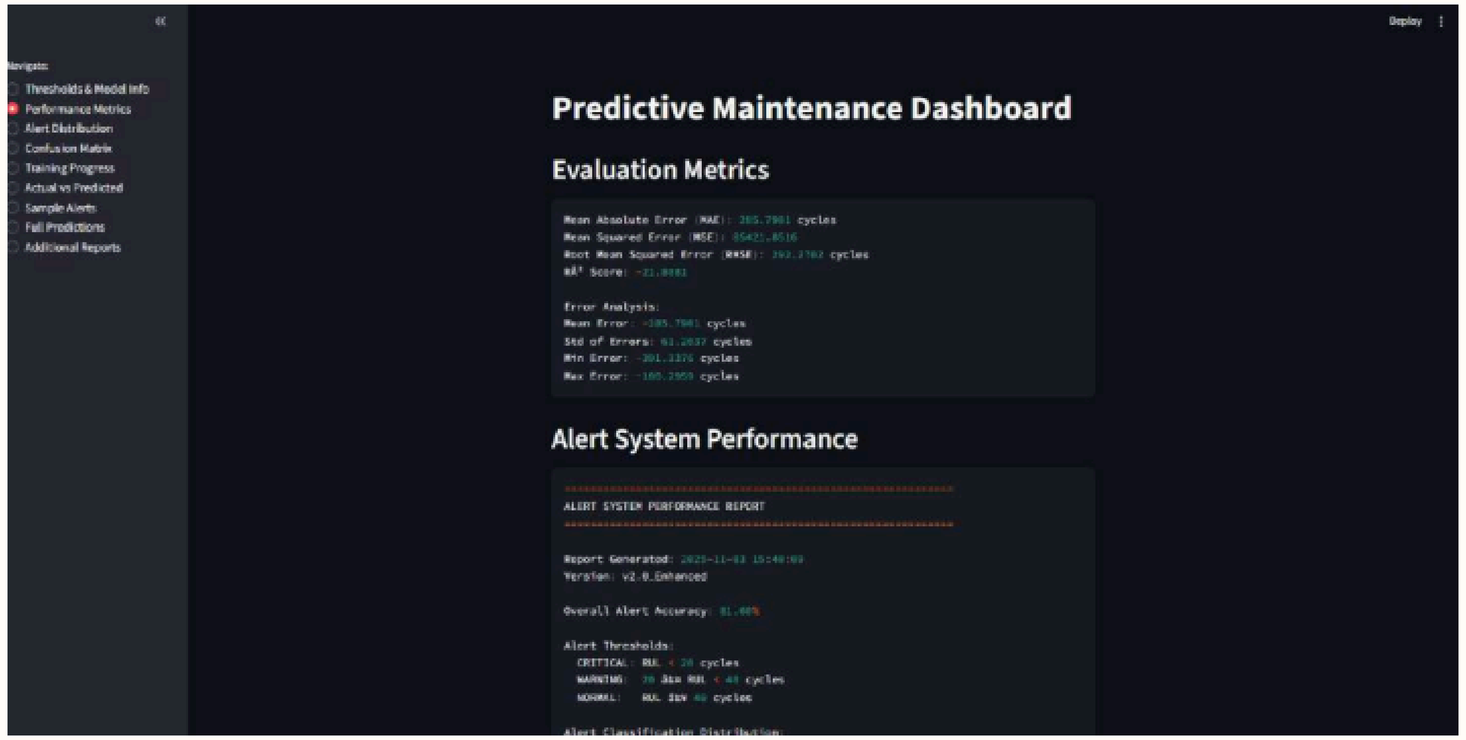
- Fully functional dashboard Python application.
- Clear, actionable graphical outputs aiding proactive maintenance.
- Documentation for deployment and usage.

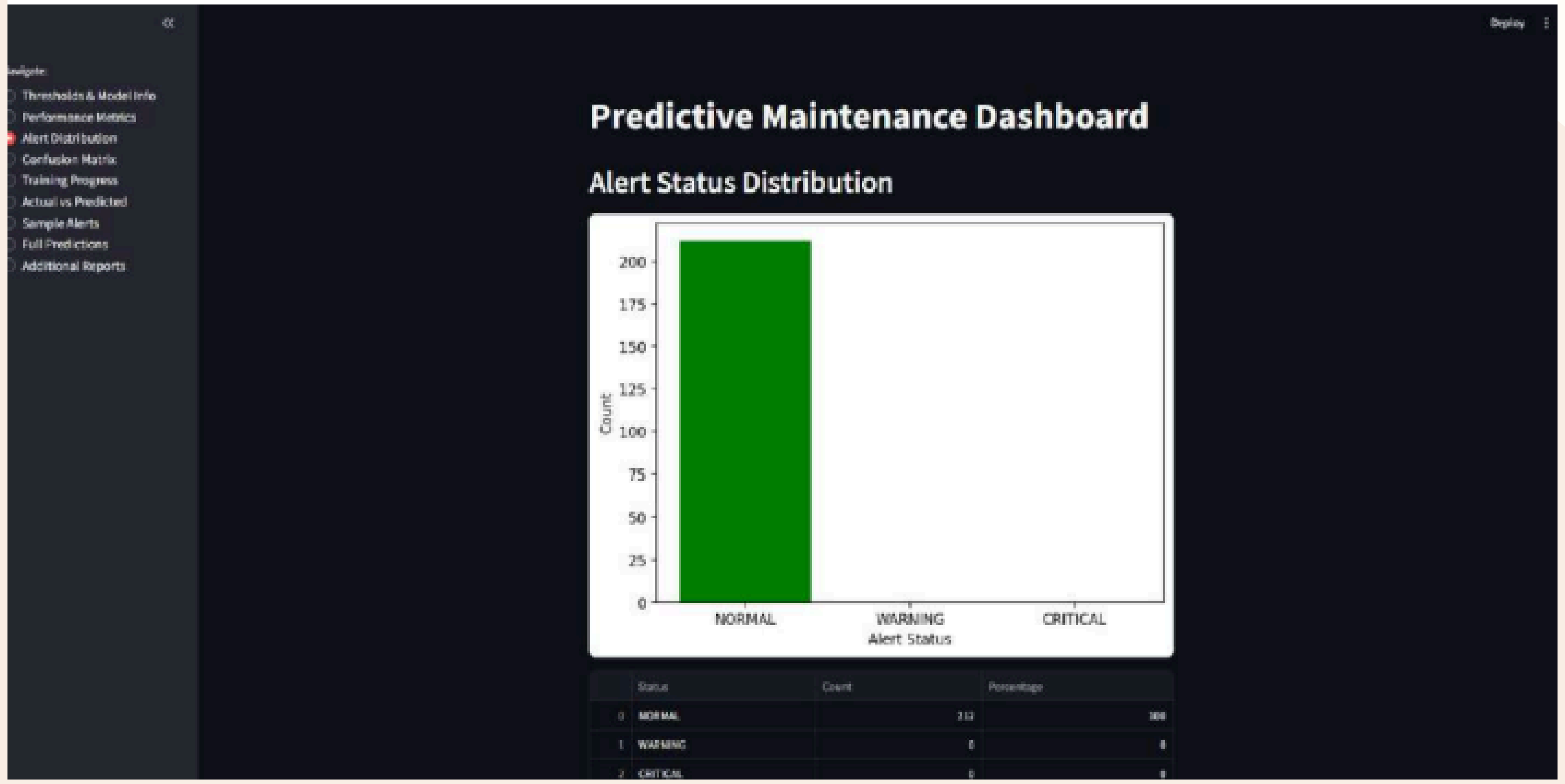
Outcome:

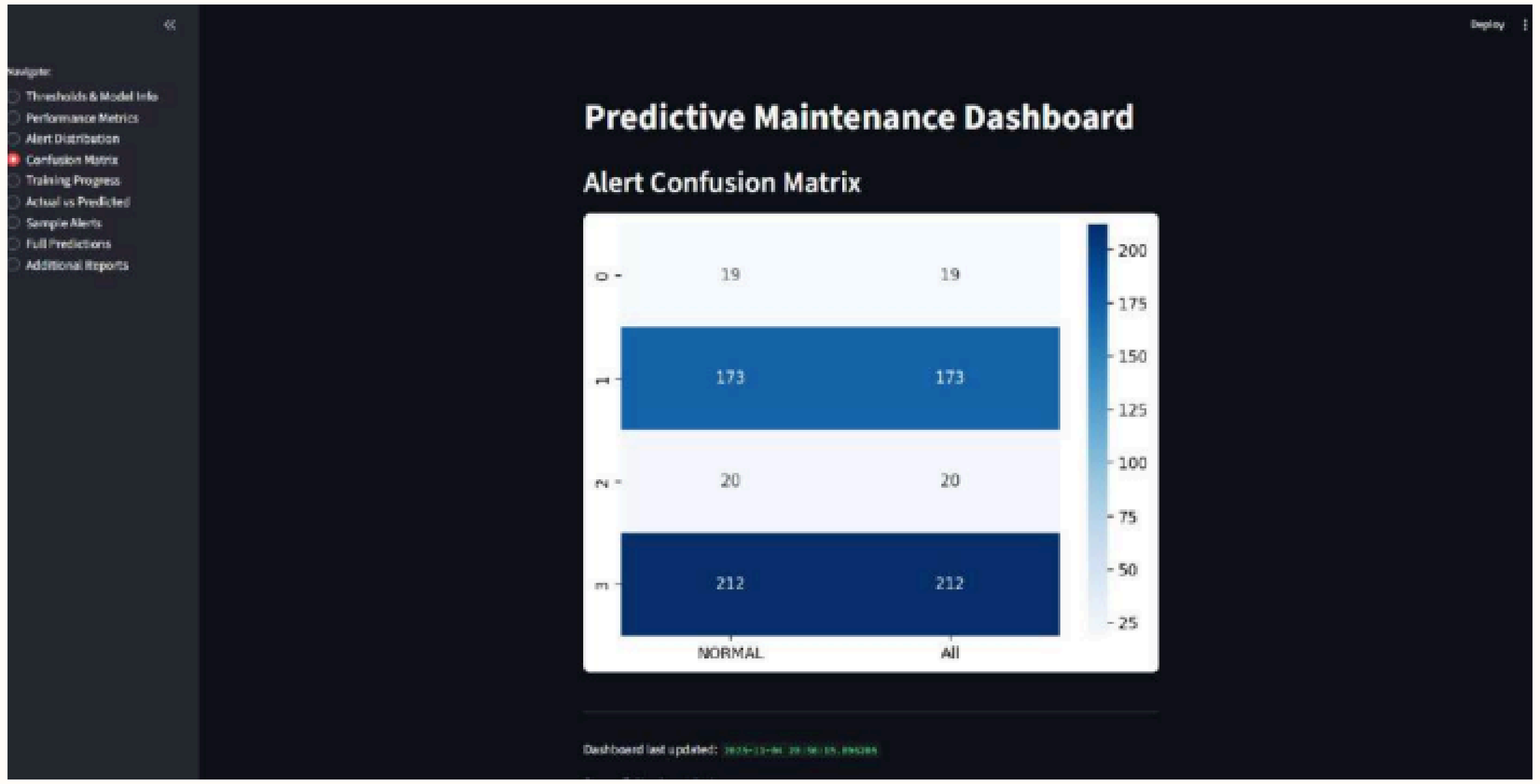
- Enhanced user experience in accessing critical asset health information.
- Real-time visualization supports timely maintenance interventions.
- Demonstrates end-to-end AI predictive maintenance solution from data to insights.

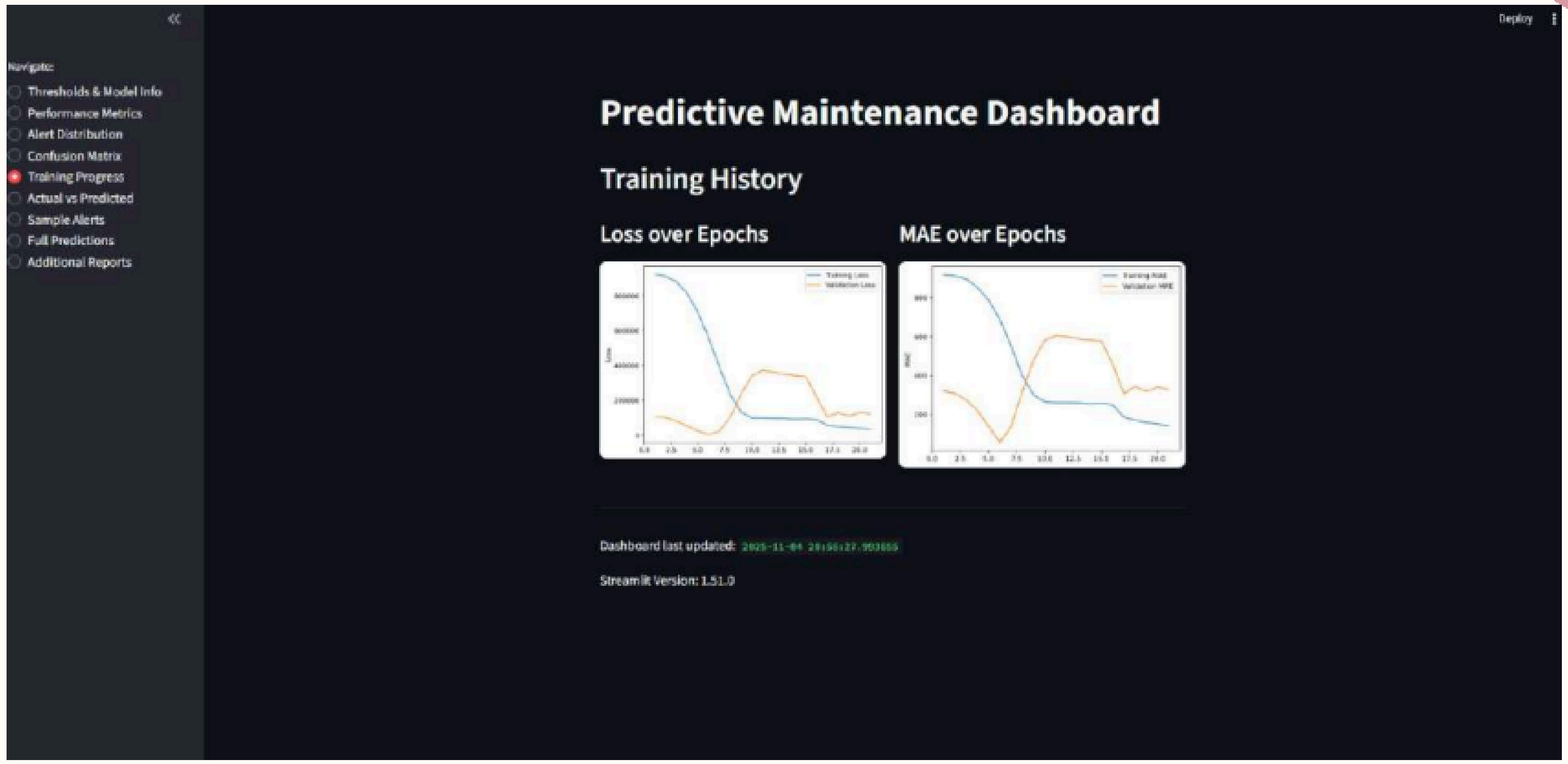
Dashboard Application Screenshot:

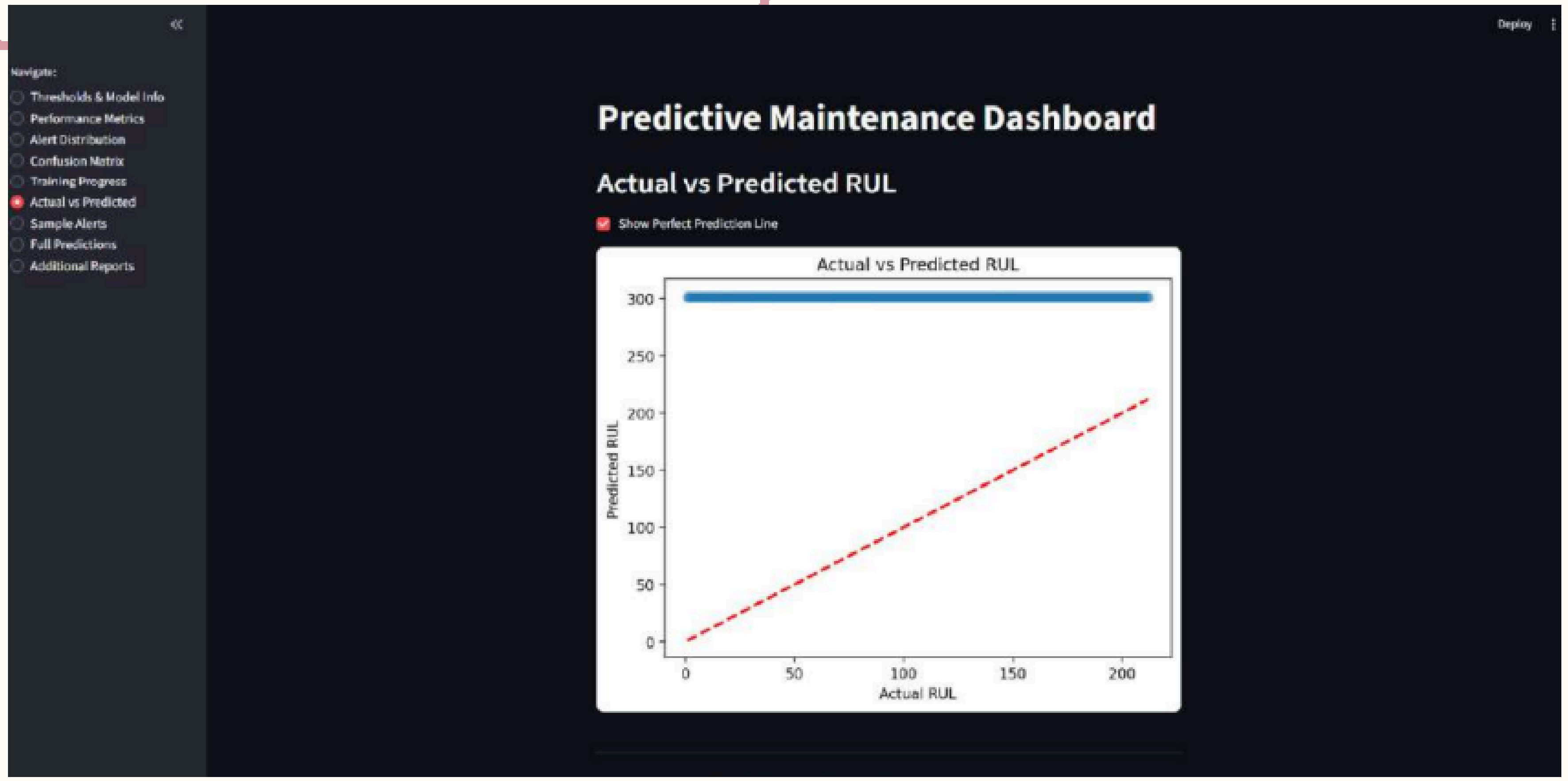


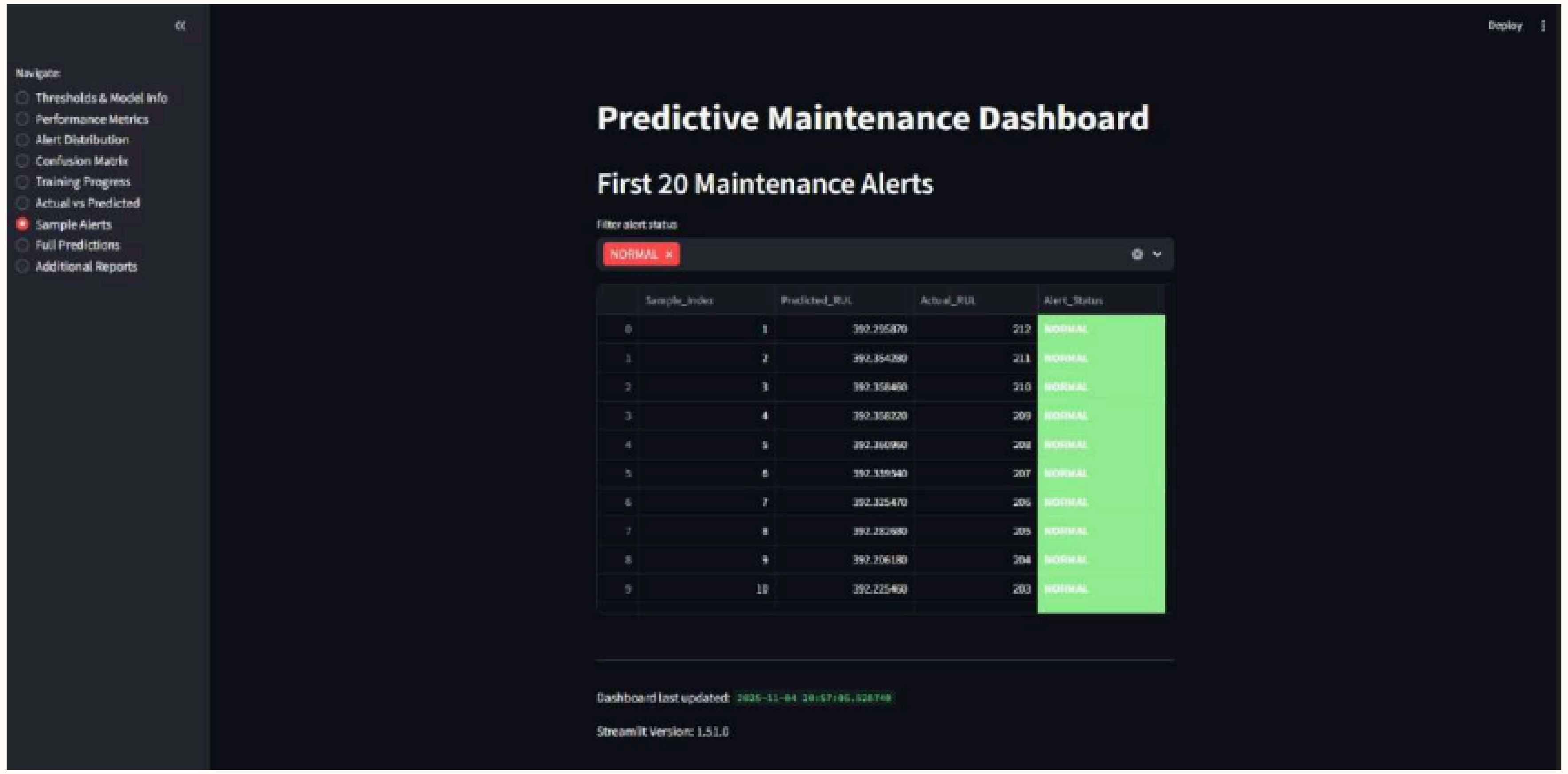


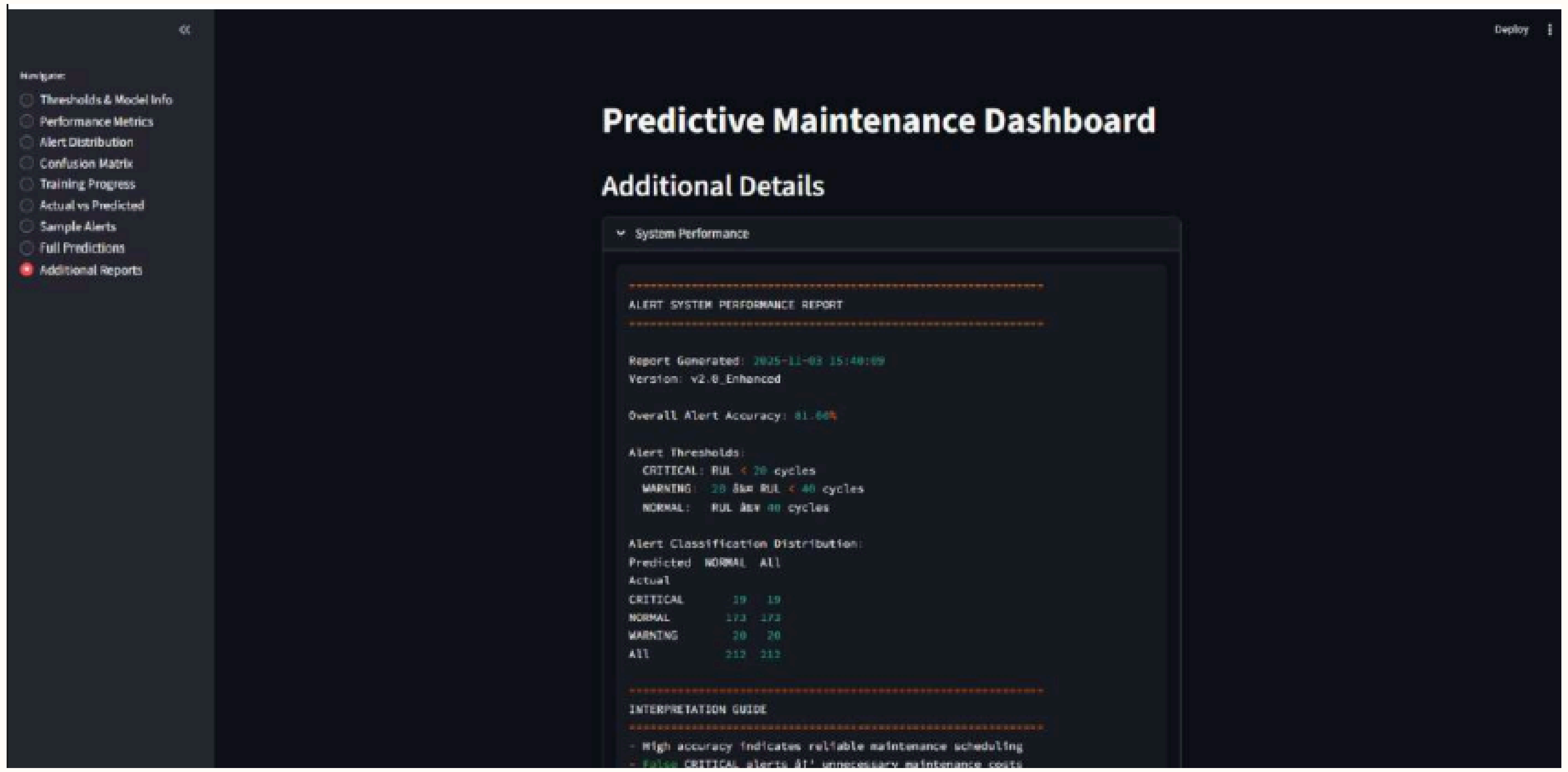





Deploy
Navigate:
Thresholds & Model Info
Performance Metrics
Alert Distribution
Confusion Matrix
Training Progress
Actual vs Predicted
Sample Alerts
Full Predictions
Additional Reports
Predictive Maintenance Dashboard
Training History
Loss over Epochs
MAE over Epochs
Dashboard last updated:
Streamlit Version: 1.51.0


Deploy
Navigate:
Thresholds & Model Info
Performance Metrics
Alert Distribution
Confusion Matrix
Training Progress
Actual vs Predicted
Sample Alerts
Full Predictions
Additional Reports
Predictive Maintenance Dashboard
Actual vs Predicted RUL
Show Perfect Prediction Line
Actual vs Predicted RUL
Predicted RUL
Actual RUL
0
50
100
150
200
250
300


Deploy
Navigate:
Thresholds & Model Info
Performance Metrics
Alert Distribution
Confusion Matrix
Training Progress
Actual vs Predicted
Sample Alerts
Full Predictions
Additional Reports
Predictive Maintenance Dashboard
First 20 Maintenance Alerts
Filter alert status
NORMAL ×
Sample_Index Predicted_RUL Actual_RUL Alert_Status
0 1 392.295870 212 NORMAL
1 2 392.354290 211 NORMAL
2 3 392.358460 210 NORMAL
3 4 392.358220 209 NORMAL
4 5 392.360960 208 NORMAL
5 6 392.339540 207 NORMAL
6 7 392.325470 206 NORMAL
7 8 392.282680 205 NORMAL
8 9 392.206180 204 NORMAL
9 10 392.225460 203 NORMAL
Dashboard last updated: 2025-11-04 20:57:06.528748
Streamlit Version: 1.51.0


Deploy
Navigate:
Thresholds & Model Info
Performance Metrics
Alert Distribution
Confusion Matrix
Training Progress
Actual vs Predicted
Sample Alerts
Full Predictions
Additional Reports
Predictive Maintenance Dashboard
Additional Details
System Performance
ALERT SYSTEM PERFORMANCE REPORT
Report Generated: 2025-11-03 15:40:09
Version: v2.0_Enhanced
Overall Alert Accuracy: 81.60%
Alert Thresholds:
CRITICAL: RUL < 20 cycles
WARNING: 20 â‰¤ RUL < 40 cycles
NORMAL: RUL â‰¥ 40 cycles
Alert Classification Distribution:
Predicted NORMAL All
Actual
CRITICAL 19 19
NORMAL 173 173
WARNING 20 20
All 212 212
INTERPRETATION GUIDE
- High accuracy indicates reliable maintenance scheduling

References

- NASA. CMAPSS Turbofan Engine Dataset.
- GeeksforGeeks. LSTM Neural Networks in Python. Available at: https://www.geeksforgeeks.org/deep-learning/long-short-term-memory-networks-using-pytorch/
- TensorFlow Documentation. Keras API Guide. Available at: https://www.tensorflow.org/guide/keras
- Scikit-learn Documentation. Data Preprocessing & Metrics Modules. Available at: https://scikit-learn.org/stable/modules/preprocessing.html
- Matplotlib and Seaborn Documentation. Data Visualization Libraries in Python. Available at: https://matplotlib.org/stable/contents.html and https://seaborn.pydata.org/
- NumPy Documentation. Array and Numerical Computation Library. Available at: https://numpy.org/doc/
- Streamlit Documentation. Interactive Data Applications in Python. Available at: https://docs.streamlit.io/

Thank you!