EV Society™

EV BATTERY SAFETY SYSTEM

AI POWERED EV BATTERY FIRE PREVENTION SYSTEM

SHREELAKSHMI

Intern,Guest Speaker

WWW.EVSOCIETY.ORG

25 FEB 2026

8:00 pm IST

INTRODUCTION

An EV battery pack is made up of many lithium-ion cells, either hundreds of large cells or a few thousand smaller ones, depending on the design.

Now, the problem usually begins with just one single cell. That first cell may fail due to reasons such as a battery management system (BMS) malfunction, an internal short circuit, improper or faulty charging, a manufacturing defect, or severe external damage to the battery pack. When that one cell fails, it starts generating excessive heat. If the heat is not controlled, it spreads to the neighbouring cells. This triggers a chain reaction known as thermal runaway, where one overheated cell causes the next one to overheat and ignite.

As this process continues from cell to cell, the fire grows stronger and spreads through the battery pack. That is why, although EV fires are uncommon, they can appear intense and are relatively more difficult to extinguish once the chain reaction begins.

Mahindra BE.6 Electric SUV Catches Fire in UP! The Company Explains What Really Caused the Blaze.

The sensor data showed the BE.6 was driven over 10 minutes on a fully deflated rear tyre, causing heat buildup that ignited the rubber and all occupants escaped safely.

INSTANCES OF ELECTRIC VEHICLES CATCHING FIRE

- EV fires usually start from a single failing cell → heat spreads → thermal runaway
 (Not every EV fire is battery-related; some are caused by crashes, tyres, wiring, or external heat)

- EVs can catch fire — but they're statistically less likely to than petrol/diesel vehicles
 (~25 EV fires per 100,000 vehicles vs ~1,500+ per 100,000 gasoline vehicles)

...

DATASET

ANAMOLY DETECTION

STATE OF HEALTH

STATE OF CHARGE

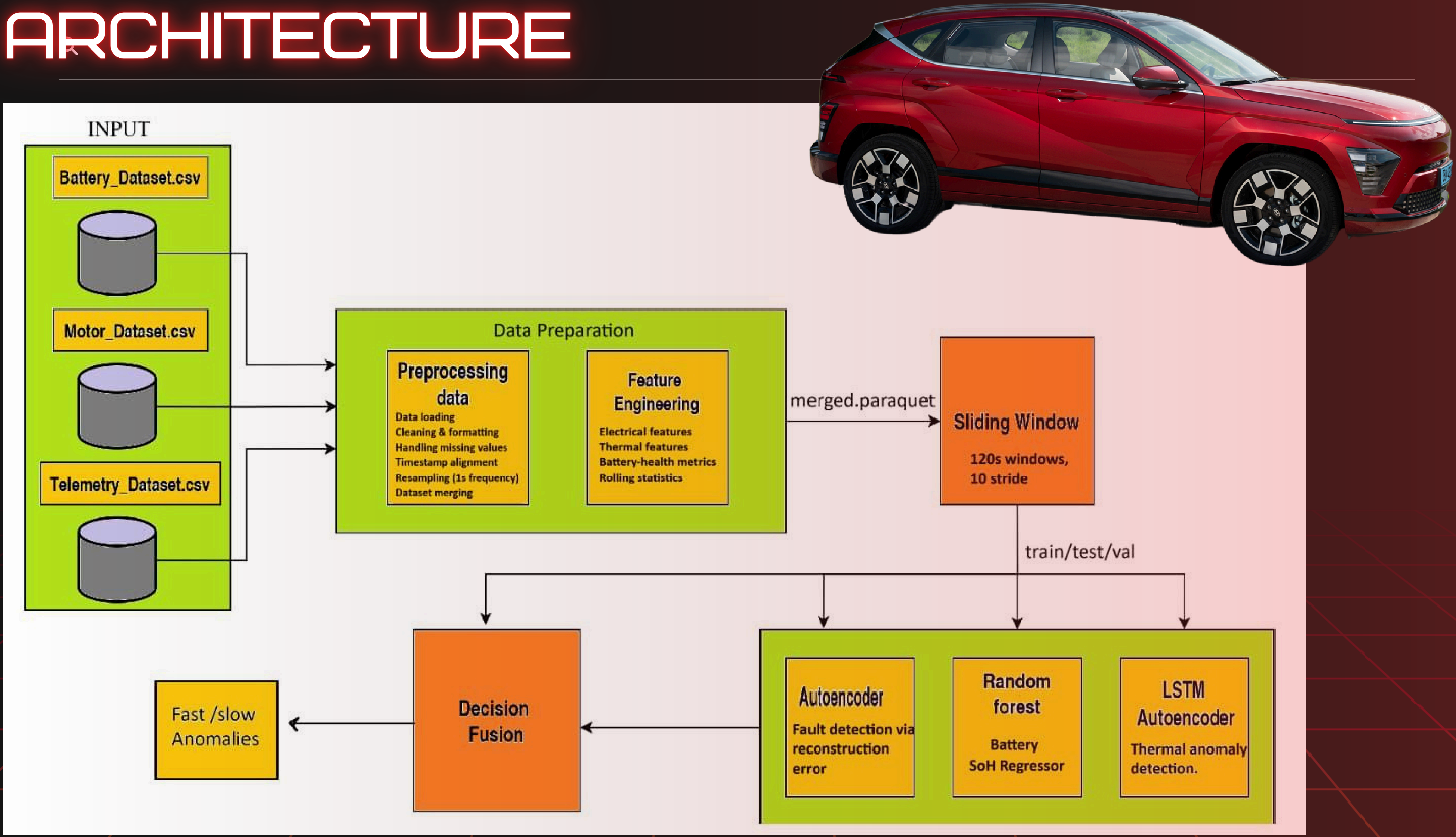
PROBLEM STATEMENT:

...

For a given dataset consisting of EV battery operational parameters such as voltage, current, temperature, state of charge (SOC), number of charge/discharge cycles, and other sensor readings,develop an AI-powered Battery Health Prediction and Failure Detection System using machine learning algorithms to:
Predict State of Charge (SOC) ,State of Health (SOH) and
Perform real-time anomaly detection

How do we find solution using AIML?

ARCHITECTURE



INPUT

Battery_Dataset.csv

Motor_Dataset.csv

Telemetry_Dataset.csv

Data Preparation

Preprocessing data
- Data loading
- Cleaning & formatting
- Handling missing values
- Timestamp alignment
- Resampling (1s frequency)
- Dataset merging

Feature Engineering
- Electrical features
- Thermal features
- Battery-health metrics
- Rolling statistics

merged.paraquet

Sliding Window

120s windows,
10 stride

train/test/val

Autoencoder

Fault detection via reconstruction error

Random forest

Battery SoH Regressor

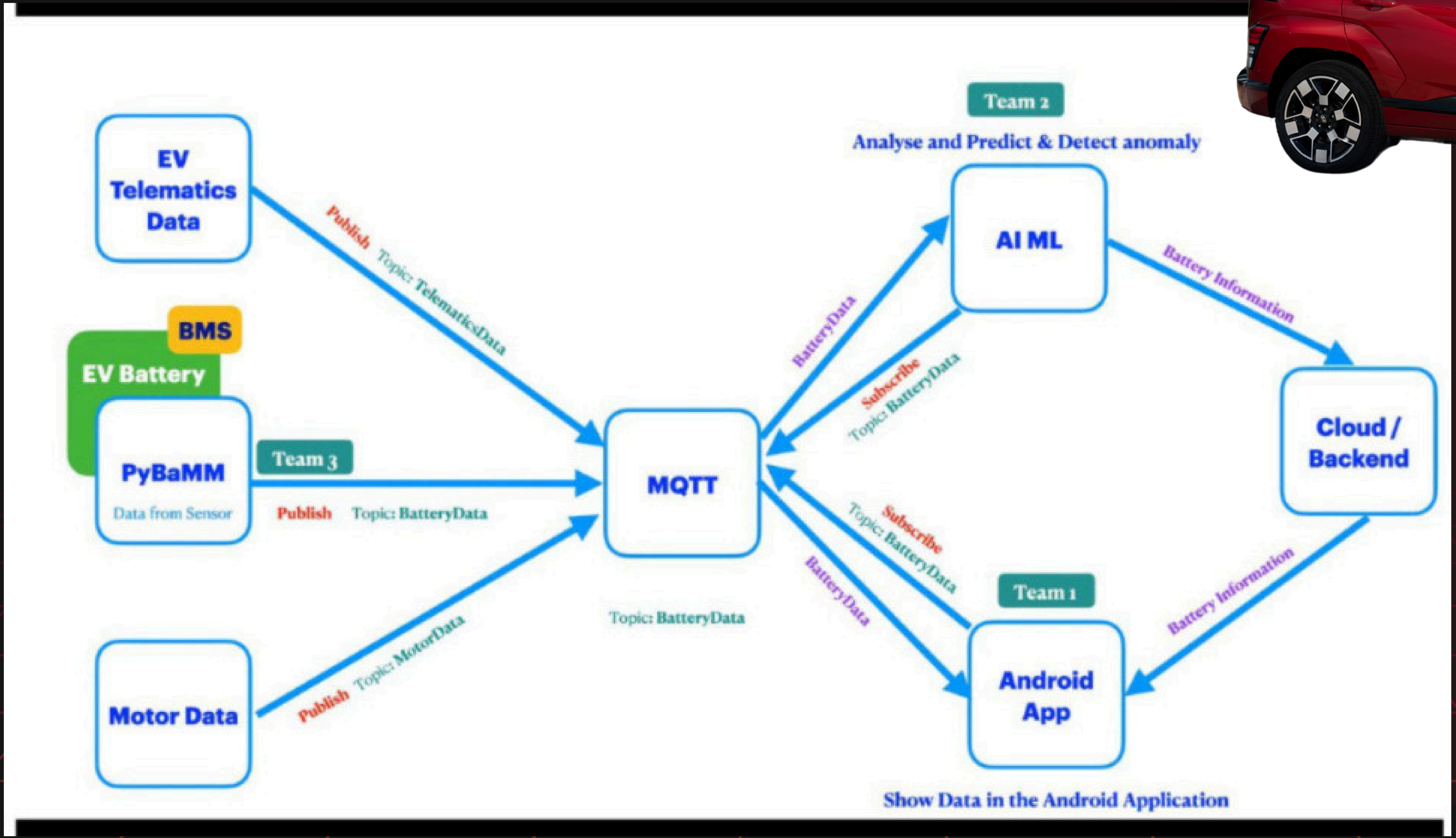
LSTM Autoencoder

Thermal anomaly detection.

Decision Fusion

Fast /slow Anomalies

OVERALL ARCHITECTURE

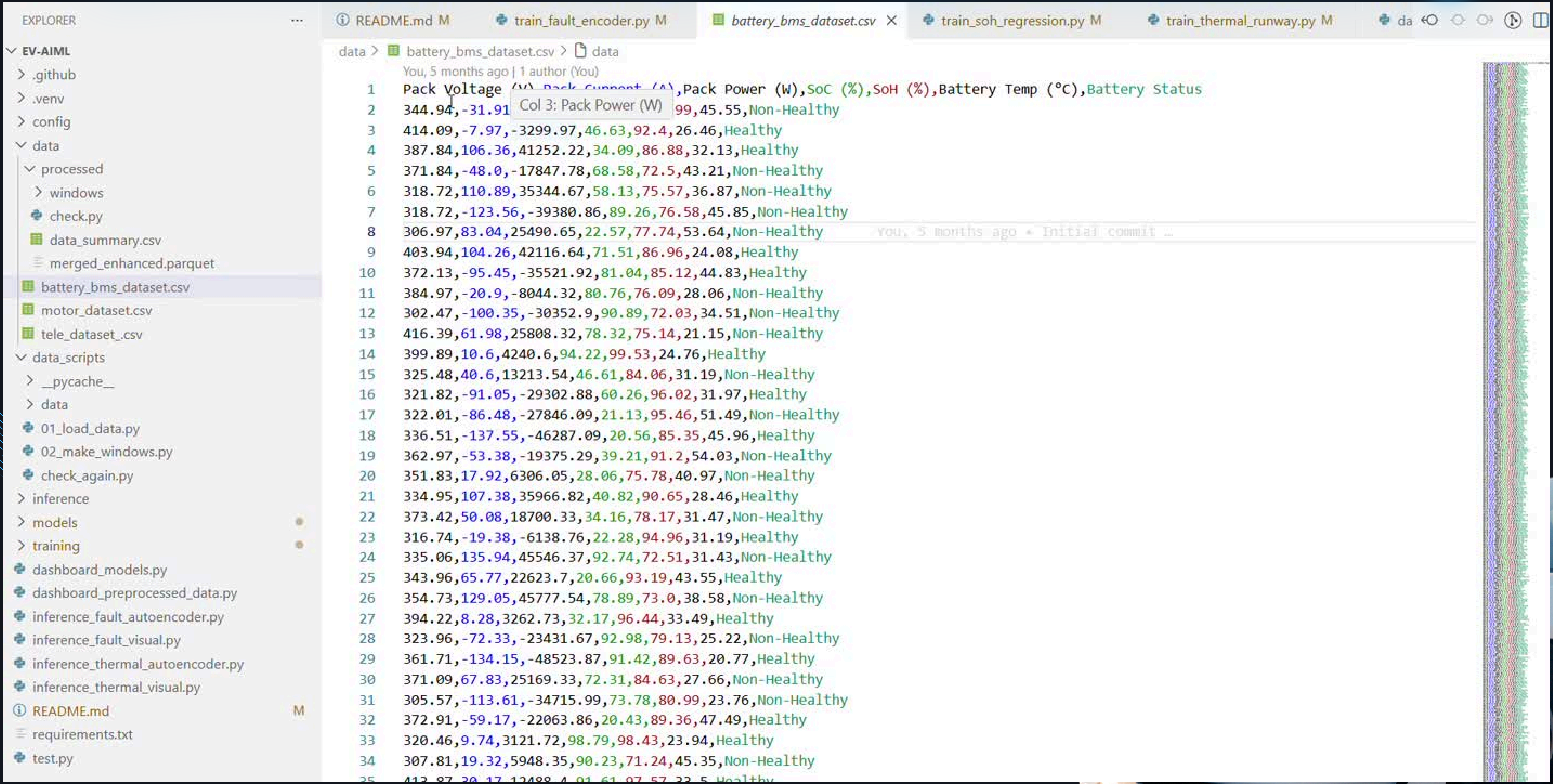


DATASET USED

Dataset Sources Overview

Dataset Name	No. of Rows	Source System	Purpose
Battery BMS Dataset	6026 rows	Battery Management System	Battery electrical & thermal behavior
Motor Dataset	5000 rows	Electric Motor Controller	Motor performance & fault detection
Telemetry Dataset	5431 rows	Vehicle Sensors	Driving behavior & environment
Final Merged Dataset	6025 rows	Combined	ML training & prediction

DATASET AND PREPROCESSING


README.md M
train_fault_encoder.py M
battery_bms_dataset.csv
train_soh_regression.py M
train_thermal_runway.py M
da
data > battery_bms_dataset.csv > data
You, 5 months ago | 1 author (You)
Col 3: Pack Power (W)
You, 5 months ago • Initial commit …

```
EXPLORER
EV-AIML
  .github
  .venv
  config
  data
    processed
      windows
      check.py
      data_summary.csv
      merged_enhanced.parquet
    battery_bms_dataset.csv
    motor_dataset.csv
    tele_dataset_.csv
  data_scripts
    __pycache__
    data
    01_load_data.py
    02_make_windows.py
    check_again.py
  inference
  models
  training
  dashboard_models.py
  dashboard_preprocessed_data.py
  inference_fault_autoencoder.py
  inference_fault_visual.py
  inference_thermal_autoencoder.py
  inference_thermal_visual.py
  README.md                          M
  requirements.txt
  test.py
```

```
Pack Voltage (V),Pack Current (A),Pack Power (W),SoC (%),SoH (%),Battery Temp (°C),Battery Status
344.94,-31.91 Col 3: Pack Power (W) 99,45.55,Non-Healthy
414.09,-7.97,-3299.97,46.63,92.4,26.46,Healthy
387.84,106.36,41252.22,34.09,86.88,32.13,Healthy
371.84,-48.0,-17847.78,68.58,72.5,43.21,Non-Healthy
318.72,110.89,35344.67,58.13,75.57,36.87,Non-Healthy
318.72,-123.56,-39380.86,89.26,76.58,45.85,Non-Healthy
306.97,83.04,25490.65,22.57,77.74,53.64,Non-Healthy
403.94,104.26,42116.64,71.51,86.96,24.08,Healthy
372.13,-95.45,-35521.92,81.04,85.12,44.83,Healthy
384.97,-20.9,-8044.32,80.76,76.09,28.06,Non-Healthy
302.47,-100.35,-30352.9,90.89,72.03,34.51,Non-Healthy
416.39,61.98,25808.32,78.32,75.14,21.15,Non-Healthy
399.89,10.6,4240.6,94.22,99.53,24.76,Healthy
325.48,40.6,13213.54,46.61,84.06,31.19,Non-Healthy
321.82,-91.05,-29302.88,60.26,96.02,31.97,Healthy
322.01,-86.48,-27846.09,21.13,95.46,51.49,Non-Healthy
336.51,-137.55,-46287.09,20.56,85.35,45.96,Healthy
362.97,-53.38,-19375.29,39.21,91.2,54.03,Non-Healthy
351.83,17.92,6306.05,28.06,75.78,40.97,Non-Healthy
334.95,107.38,35966.82,40.82,90.65,28.46,Healthy
373.42,50.08,18700.33,34.16,78.17,31.47,Non-Healthy
316.74,-19.38,-6138.76,22.28,94.96,31.19,Healthy
335.06,135.94,45546.37,92.74,72.51,31.43,Non-Healthy
343.96,65.77,22623.7,20.66,93.19,43.55,Healthy
354.73,129.05,45777.54,78.89,73.0,38.58,Non-Healthy
394.22,8.28,3262.73,32.17,96.44,33.49,Healthy
323.96,-72.33,-23431.67,92.98,79.13,25.22,Non-Healthy
361.71,-134.15,-48523.87,91.42,89.63,20.77,Healthy
371.09,67.83,25169.33,72.31,84.63,27.66,Non-Healthy
305.57,-113.61,-34715.99,73.78,80.99,23.76,Non-Healthy
372.91,-59.17,-22063.86,20.43,89.36,47.49,Healthy
320.46,9.74,3121.72,98.79,98.43,23.94,Healthy
307.81,19.32,5948.35,90.23,71.24,45.35,Non-Healthy
413.87,30.17,12488.4,91.61,97.57,33.5,Healthy
```

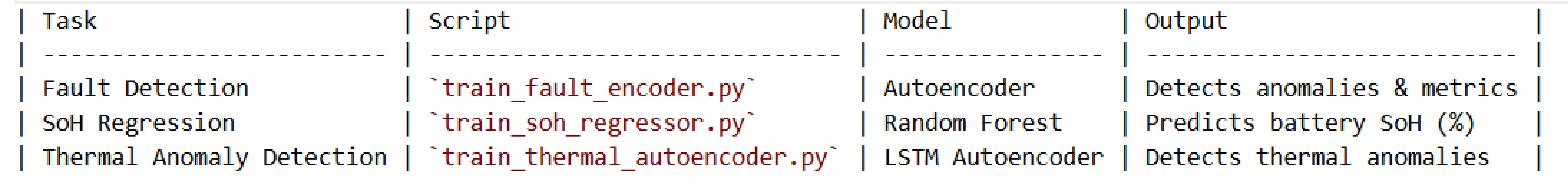
ML AND DL MODELS USED

AUTOENCODER

RANDOM FOREST

LSTM AUTOENCODER

- WHAT IS AUTOENCODER?
- WHAT IS RANDOM FOREST?
- HOW DOES THESE WORK?

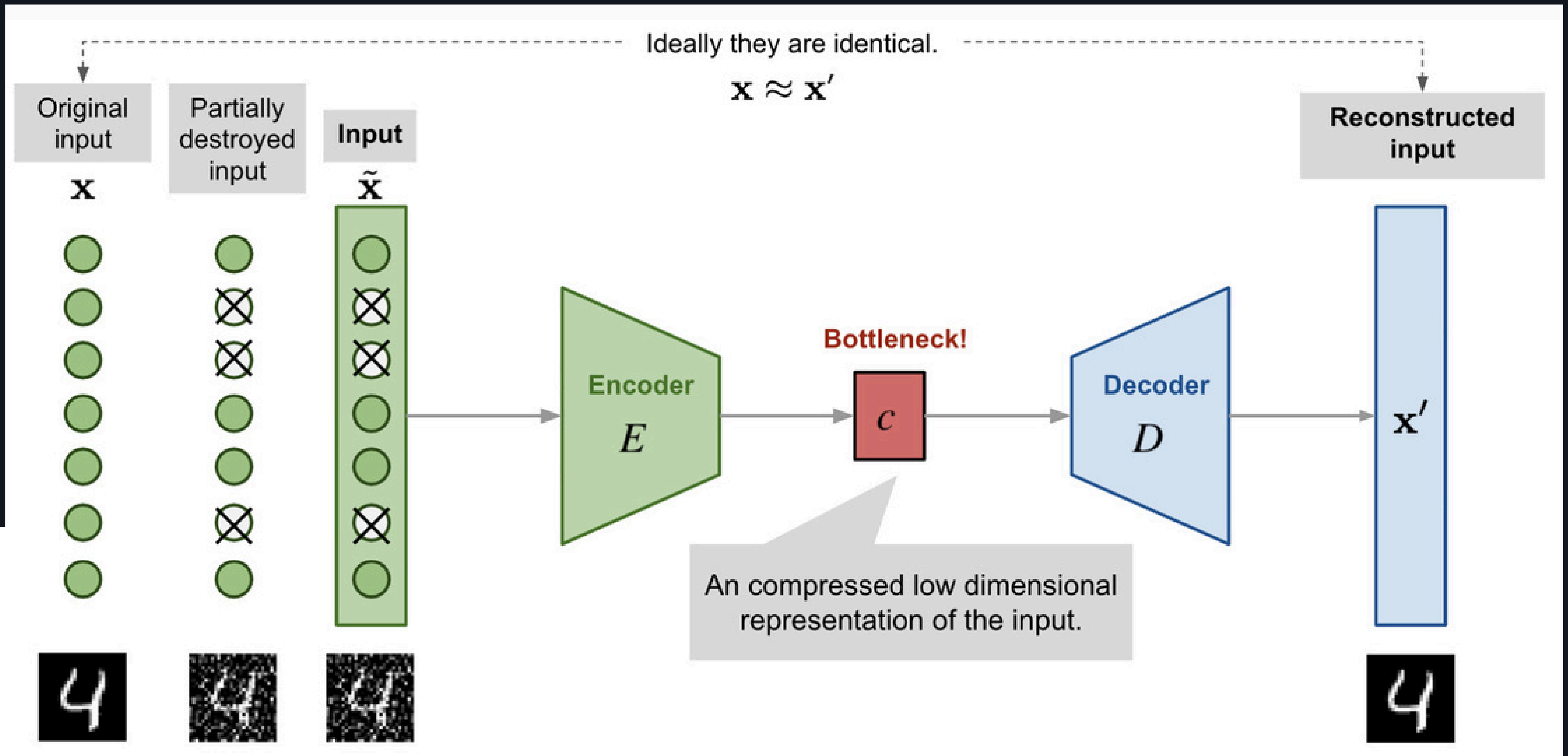


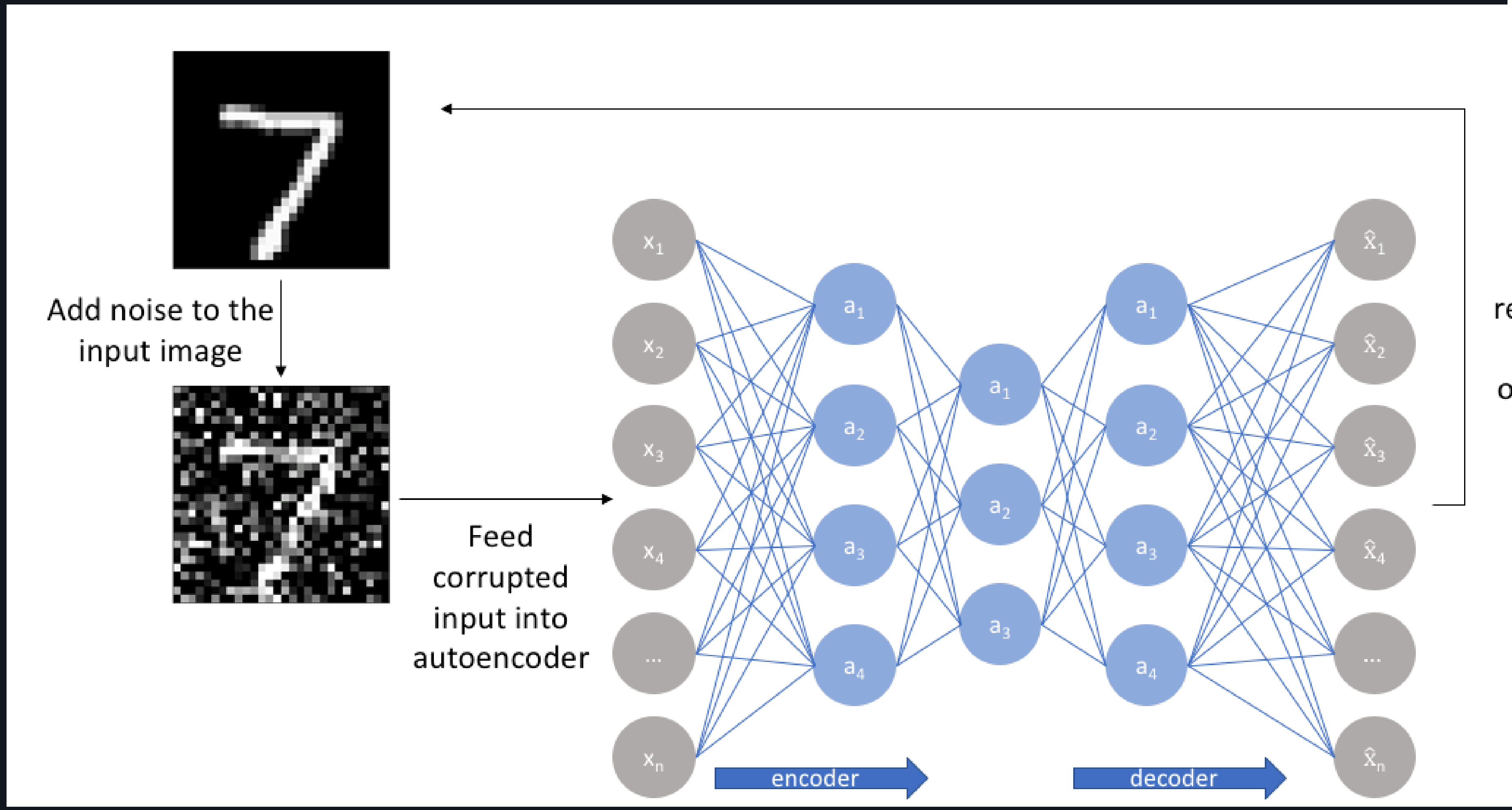
Task	Script	Model	Output
Fault Detection	`train_fault_encoder.py`	Autoencoder	Detects anomalies & metrics
SoH Regression	`train_soh_regressor.py`	Random Forest	Predicts battery SoH (%)
Thermal Anomaly Detection	`train_thermal_autoencoder.py`	LSTM Autoencoder	Detects thermal anomalies

How does Autoencoders work?

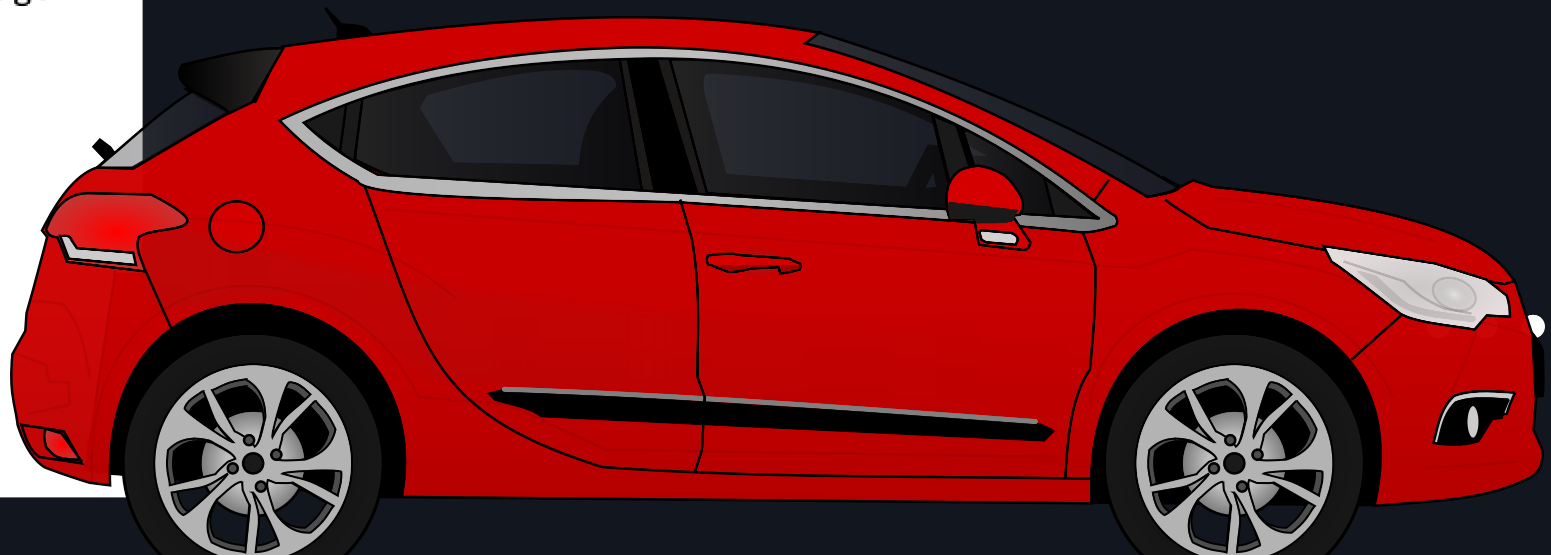
LSTM AUTOENCODER

AUTOENCODER

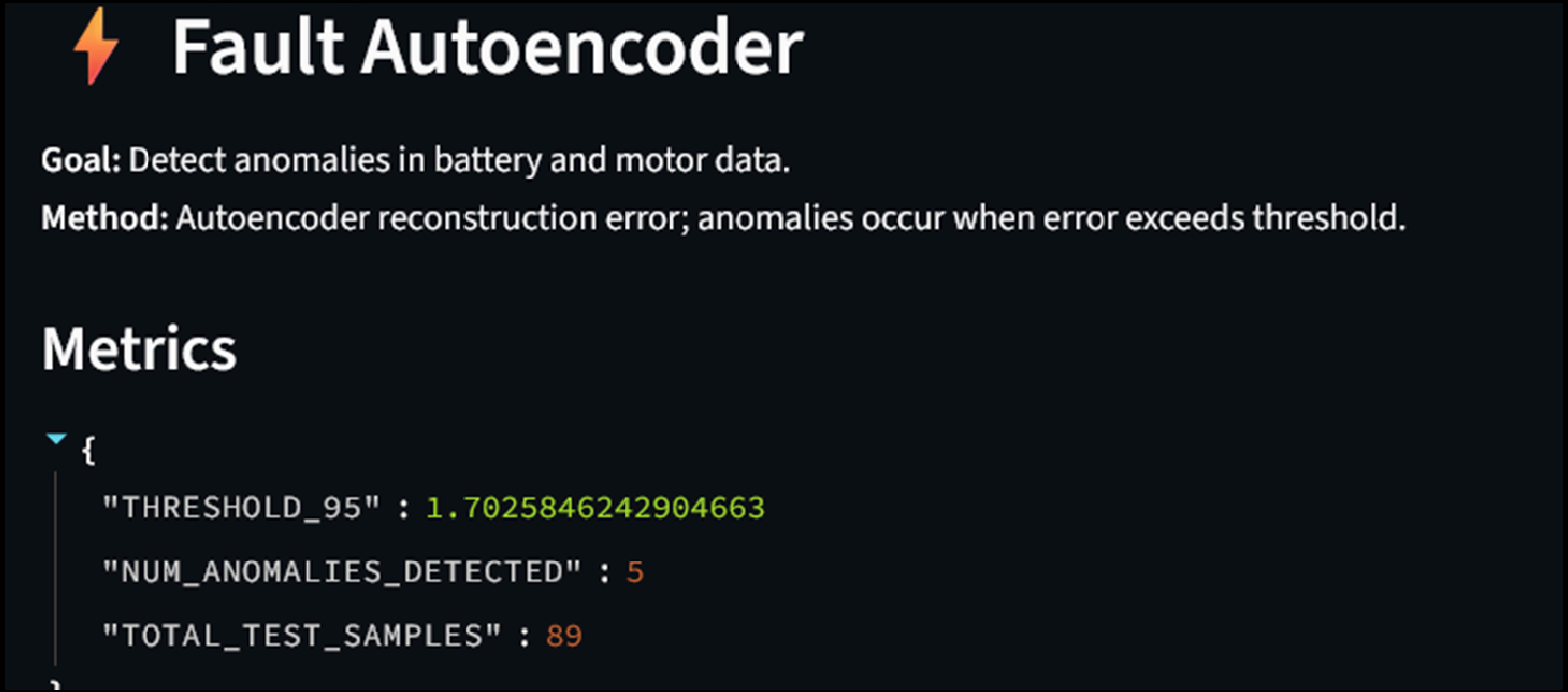

Ideally they are identical.
$\mathbf{x} \approx \mathbf{x}'$
Original input
$\mathbf{x}$
Partially destroyed input
Input
$\tilde{\mathbf{x}}$
Reconstructed input
Encoder
E
Bottleneck!
c
Decoder
D
$\mathbf{x}'$
An compressed low dimensional representation of the input.



Measure reconstruction loss against original image



IMPLEMENTATION



⚡ Fault Autoencoder

Goal: Detect anomalies in battery and motor data.

Method: Autoencoder reconstruction error; anomalies occur when error exceeds threshold.

Metrics

```
▾ {
    "THRESHOLD_95" : 1.7025846242904663
    "NUM_ANOMALIES_DETECTED" : 5
    "TOTAL_TEST_SAMPLES" : 89
```

Goal: Detect operational faults in battery and motor data.

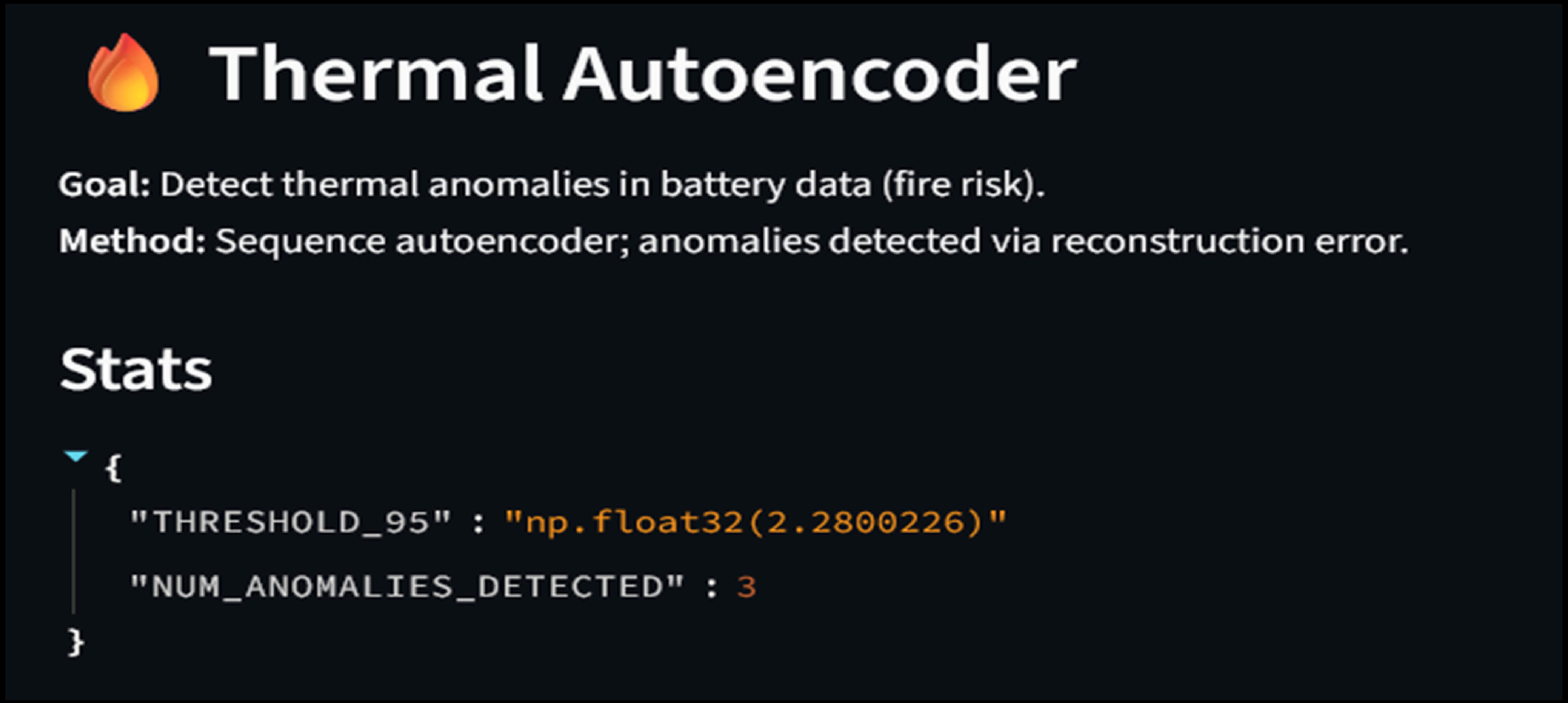
Method: Autoencoder detects anomalies using reconstruction error.

Results: Threshold = 1.70, Anomalies Detected = 5 / 89 samples

Performance: The model is balanced, shows no overfitting, and accurately identifies unusual behavior in test data.

Conclusion: Works reliably for detecting real-time motor or battery faults.

IMPLEMENTATION



🔥 Thermal Autoencoder

Goal: Detect thermal anomalies in battery data (fire risk).

Method: Sequence autoencoder; anomalies detected via reconstruction error.

Stats

```
{
  "THRESHOLD_95" : "np.float32(2.2800226)"
  "NUM_ANOMALIES_DETECTED" : 3
}
```

Goal: Detect temperature-related or fire-risk anomalies.

Method: Sequence autoencoder analyzes temperature patterns over time.

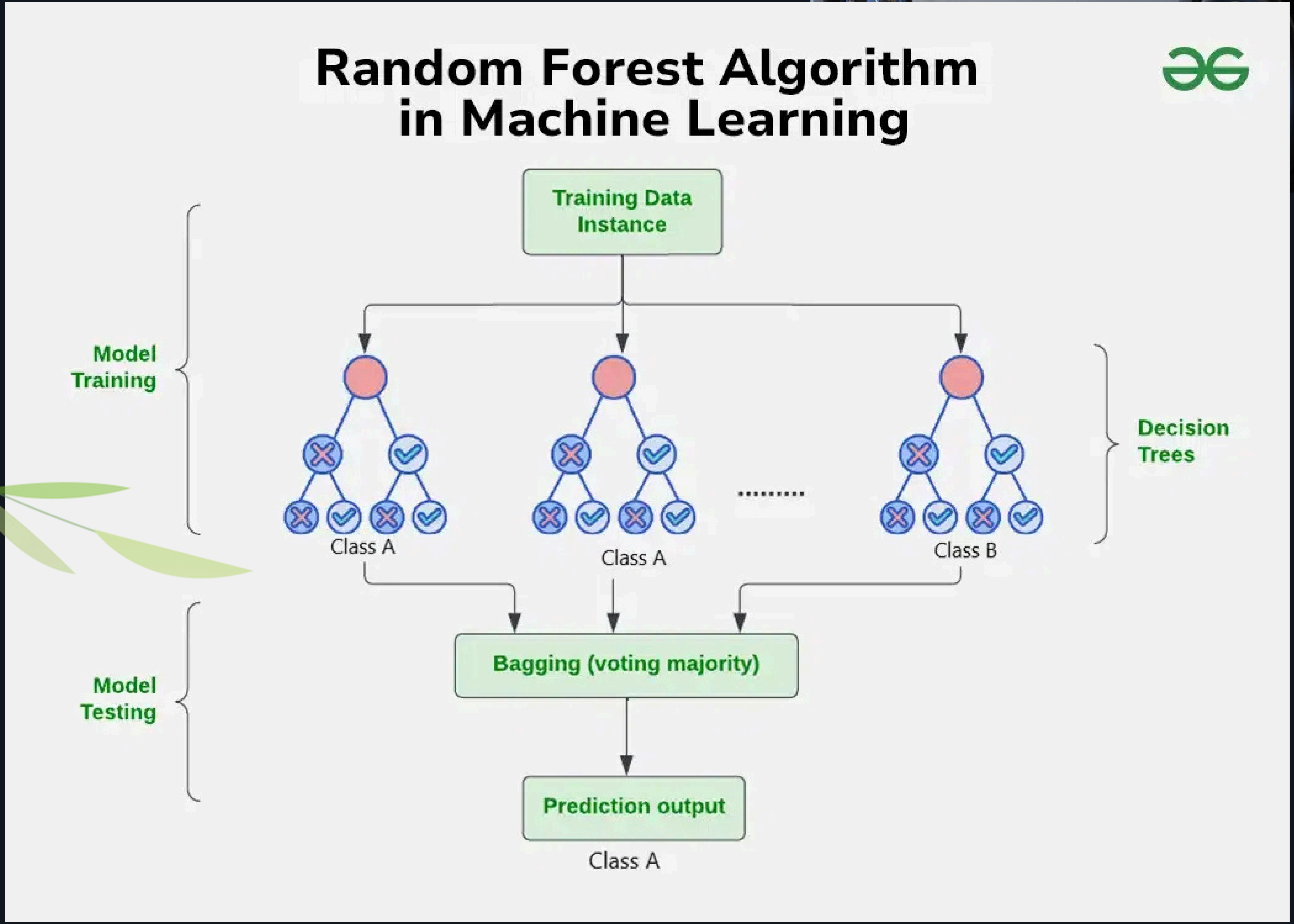
Results: Threshold = 2.28, Anomalies Detected=3

Performance: The model is stable, captures time-based variations effectively, and shows balanced generalization.

Conclusion: Provides accurate thermal anomaly detection and supports early fire-risk prediction

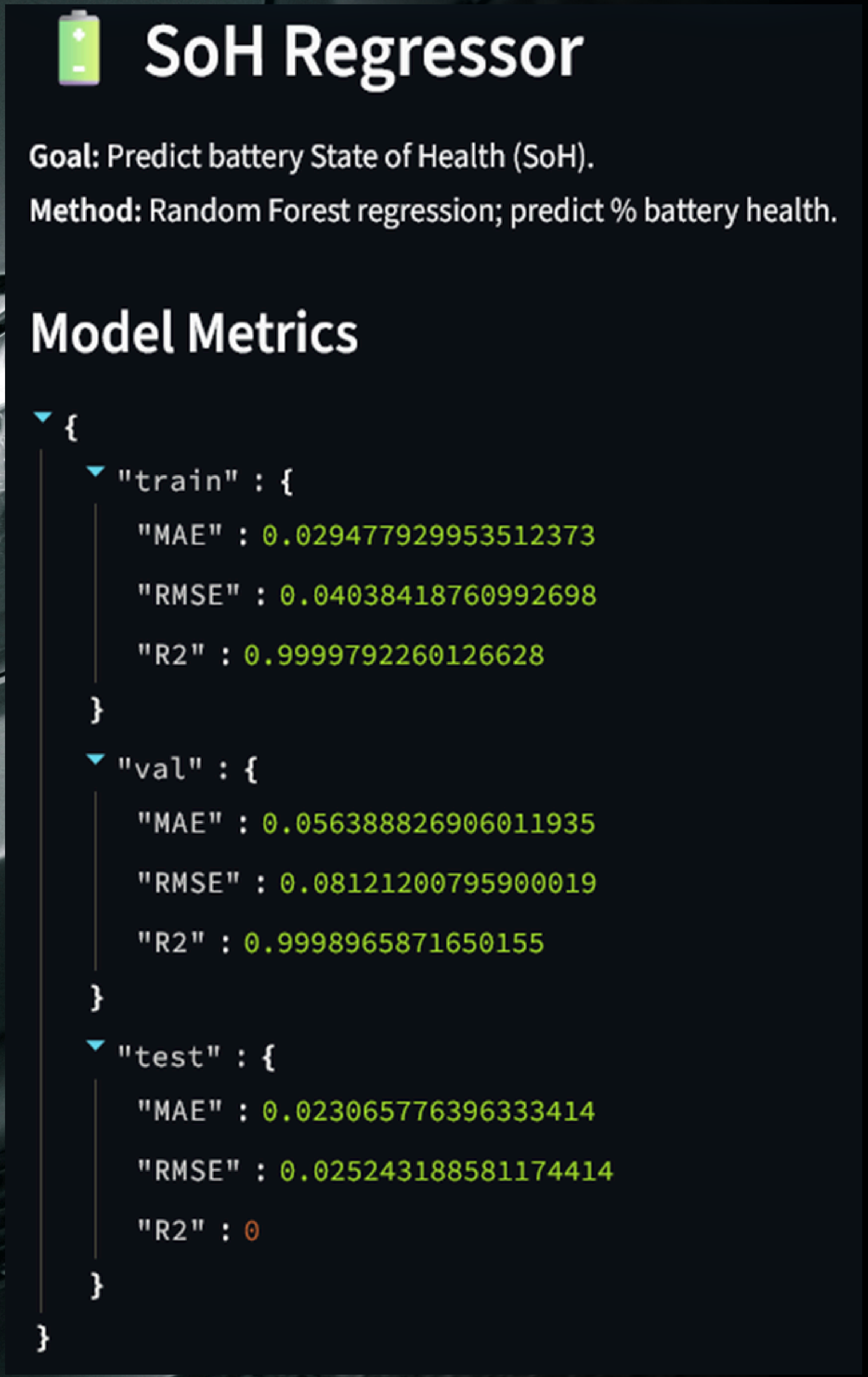
How does random forest work ?

RANDOM FOREST



Random Forest Algorithm in Machine Learning

IMPLEMENTATION



🔋 SoH Regressor

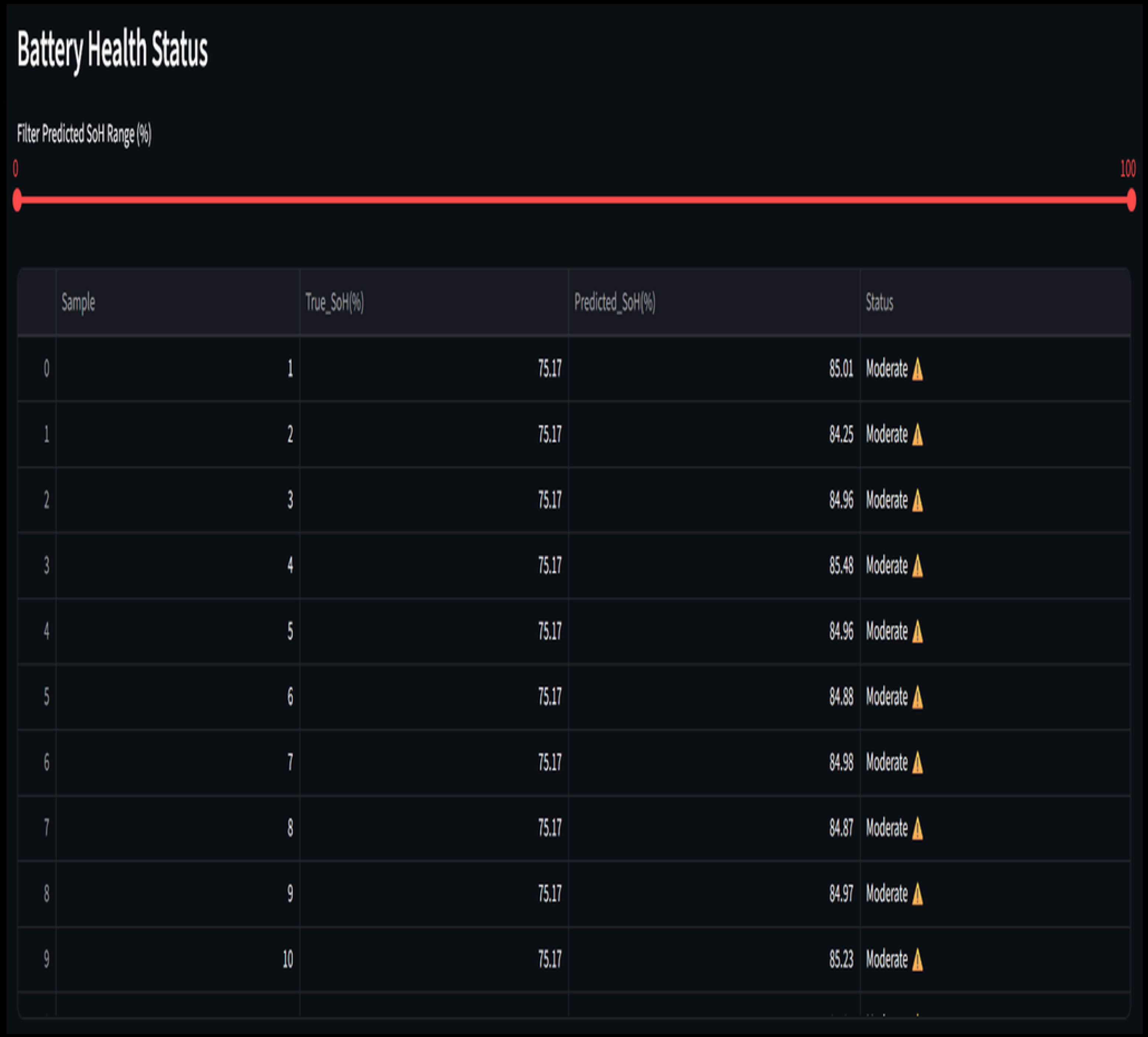
Goal: Predict battery State of Health (SoH).

Method: Random Forest regression; predict % battery health.

Model Metrics

```
{
  "train" : {
    "MAE" : 0.029477929953512373
    "RMSE" : 0.04038418760992698
    "R2" : 0.9999792260126628
  }
  "val" : {
    "MAE" : 0.056388826906011935
    "RMSE" : 0.08121200795900019
    "R2" : 0.9998965871650155
  }
  "test" : {
    "MAE" : 0.023065776396333414
    "RMSE" : 0.025243188581174414
    "R2" : 0
  }
}
```

The battery is middle aged so soh and soc values are around 75.16



Battery Health Status

Filter Predicted SoH Range (%)

0 – 100

	Sample	True_SoH(%)	Predicted_SoH(%)	Status
0	1	75.17	85.01	Moderate ⚠
1	2	75.17	84.25	Moderate ⚠
2	3	75.17	84.96	Moderate ⚠
3	4	75.17	85.48	Moderate ⚠
4	5	75.17	84.96	Moderate ⚠
5	6	75.17	84.88	Moderate ⚠
6	7	75.17	84.98	Moderate ⚠
7	8	75.17	84.87	Moderate ⚠
8	9	75.17	84.97	Moderate ⚠
9	10	75.17	85.23	Moderate ⚠

Goal: Estimate State of Health (SoH) and State of Charge (SoC) of the battery.

Results: SoH ≈ 75.16%

Performance: The model is overfitting because the dataset contains limited variation in SoH values (mostly mid-range).

Conclusion: Predicts mid-life battery condition well, but needs more diverse data to learn full degradation patterns

FUTURE WORK

CLOUD DEPLOYEMNET

Cloud Deployment: Dockerize and deploy on AWS EC2 with FastAPI/ Flask APIs for live and historical data.

UI Integration

Provide APIs for CS team dashboard showing SOC, SOH, charging state, real-time charts, and AI results.

Real-Time MQTT Data:

EC team sends live motor, battery, and telematics data; AI system processes and displays anomalies.

Fault Localization

Use Explainable AI to pinpoint issues like rotor fault or battery module overheating

CONCLUSION

THIS project develops an AI-based EV battery health and fault detection system that integrates battery, motor, and vehicle telemetry data into a unified dataset. Using random forest for State of Health (SOH) prediction and autoencoder-based models for electrical and thermal anomaly detection,

The system identifies potential faults and fire-risk conditions. By combining multi-sensor data with machine learning, the framework enables safer battery monitoring, smarter charging decisions, and improved reliability in electric vehicles.

THANK YOU