Predicting EV Charger Failures Using OCPP Logs and Ensemble Machine Learning

# Abstract

This study presents a predictive modeling approach to identify potential failures in Electric Vehicle (EV) chargers using OCPP (Open Charge Point Protocol) logs. We leverage time-series telemetry such as meter values, status notifications, heartbeat signals, and boot notifications. Given the high class imbalance between failure and non-failure instances, we adopt a hybrid resampling strategy combining SMOTE (for minority class oversampling) and clustering-based undersampling (for the majority class). Feature engineering is performed at an hourly resolution, and a stacked ensemble learning model is developed using XGBoost, LightGBM, and Logistic Regression as a meta-learner. The final model yields high recall and robust generalization. We also implement threshold tuning and confidence tracking to align predictions with business risk tolerances.

# 1. Introduction

EV infrastructure reliability is critical to adoption. Timely prediction of charger failure can reduce downtime and improve user satisfaction. OCPP logs contain rich telemetry data recorded at high frequency (1-second interval) for each charger. This work focuses on forecasting failures within a 30-day horizon based on patterns in historical OCPP data.

# 2. Dataset Description

- Source: OCPP 1.6 log data from multiple EV chargers  
- Sampling Rate: 1-second resolution per charger  
- Events Used: MeterValues, StatusNotification, Heartbeat, BootNotification  
- Target Variable: Binary label indicating if a failure occurs within 30 days of the current timestamp

# 3. Role of OCPP in Charger Management

The Open Charge Point Protocol (OCPP) is an open-source communication standard that enables interoperability between EV chargers (Charge Points) and Central Management Systems (CSMS). OCPP facilitates remote monitoring, diagnostics, firmware updates, session management, and energy data collection. Through regular telemetry like meter values and heartbeat messages, the protocol provides real-time visibility into charger health. OCPP also allows standard logging of status changes and boot sequences, which are critical for predictive analytics and preventive maintenance. This project leverages OCPP logs as the primary source of information to detect early signs of failure, allowing operators to take proactive measures.

# 4. Feature Engineering

* Aggregated all raw event logs to **hourly windows** per charger
* Derived the following key features:
  + mean\_voltage: average voltage per hour
  + std\_current: standard deviation of current per hour
  + max\_current: maximum current recorded per hour
  + min\_voltage: minimum voltage recorded per hour
  + energy\_delta: change in meter reading per hour
  + pct\_time\_faulted: percentage of time charger spent in a faulted state per hour
  + status\_transitions: number of status changes (e.g., Available → Charging → Suspended)
  + status\_entropy: entropy of status types indicating stability or chaos in operation
  + boot\_events: number of times a charger rebooted in an hour
  + boot\_interval\_mean: mean interval between boot notifications
  + heartbeat\_loss\_rate: percentage of expected heartbeats missed
  + charging\_session\_count: number of completed sessions per hour
  + charging\_success\_rate: successful session ratio in an hour
  + mean\_session\_duration: average time spent in a charging session
  + time\_since\_last\_fault: hours since the charger last experienced a fault
  + fault\_duration\_avg: average fault duration in the past 24 hours
  + fault\_burst\_count: count of multiple faults occurring in quick succession
  + Rolling statistics:
    - mean\_voltage\_6h: average voltage over past 6 hours
    - max\_faults\_24h: maximum fault count in any one-hour window over the last day
    - heartbeat\_gap\_std: variability in heartbeat interval timings

## Sample Feature Table (Hourly Aggregated)

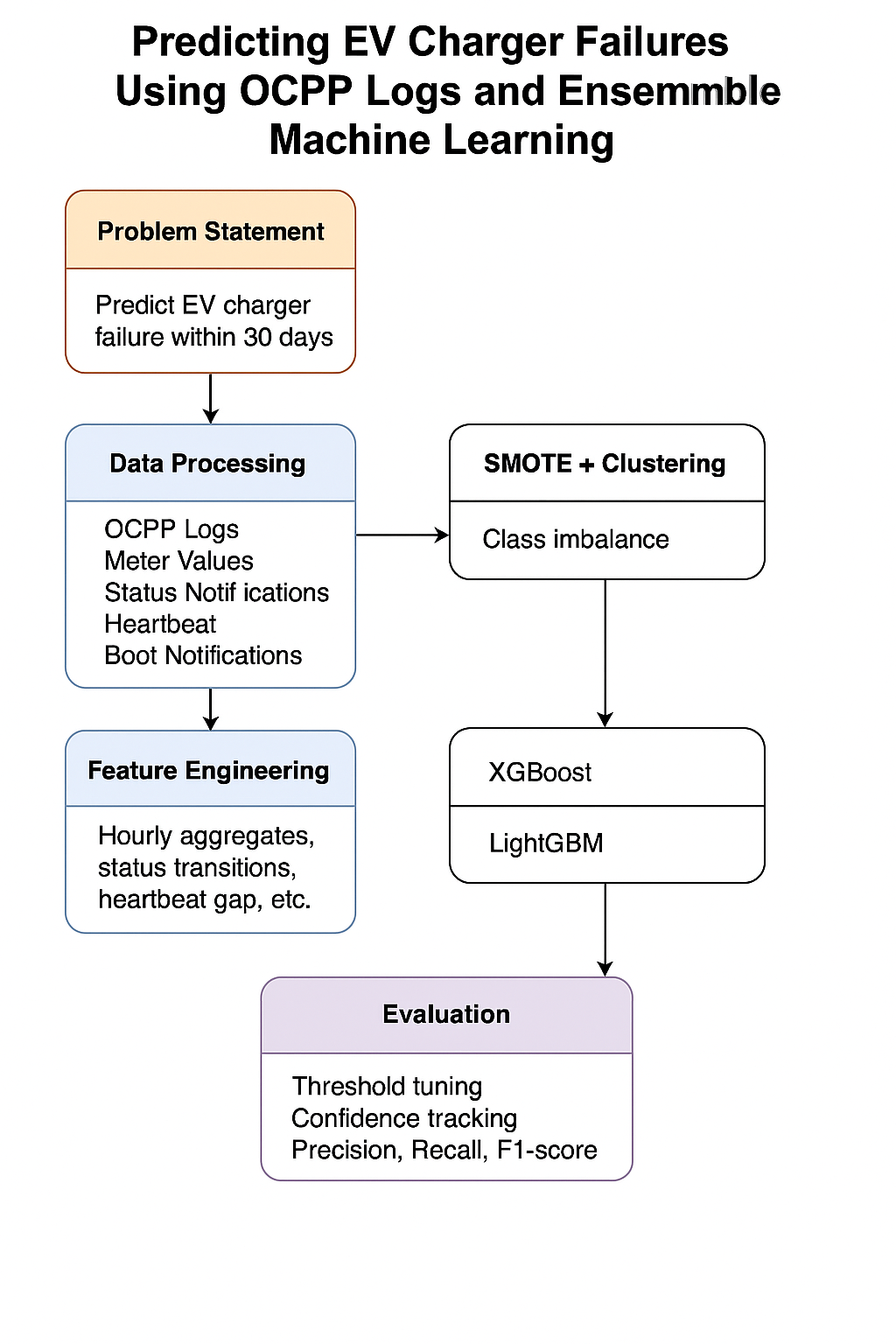
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Charger\_ID** | CHG001 | CHG001 | CHG001 | CHG001 | CHG001 |
| **Timestamp** | 07/01/2025 1:00 | 07/01/2025 2:00 | 07/01/2025 3:00 | 07/01/2025 4:00 | 07/01/2025 5:00 |
| **mean\_voltage** | 231 | 229.7 | 231.3 | 233 | 229.5 |
| **std\_current** | 2.24 | 1.73 | 1.82 | 1.37 | 1.64 |
| **max\_current** | 3.8 | 3.57 | 3.45 | 3.38 | 2.91 |
| **min\_voltage** | 227 | 227.6 | 225.8 | 225.6 | 229.6 |
| **energy\_delta** | 2.41 | 2.64 | 3.09 | 2.29 | 2.18 |
| **pct\_time\_faulted** | 0.08 | 0.04 | 0.02 | 0.07 | 0.19 |
| **status\_transitions** | 1 | 1 | 3 | 2 | 5 |
| **status\_entropy** | 0.52 | 0.39 | 0.24 | 0.79 | 0.88 |
| **boot\_events** | 0 | 0 | 0 | 0 | 0 |
| **boot\_interval\_mean** | 12 | 11.6 | 13.2 | 13.9 | 10.8 |
| **heartbeat\_loss\_rate** | 0.02 | 0.04 | 0.01 | 0.03 | 0.06 |
| **charging\_session\_count** | 3 | 3 | 1 | 1 | 2 |
| **charging\_success\_rate** | 0.79 | 0.76 | 0.78 | 0.85 | 0.81 |
| **mean\_session\_duration** | 29.2 | 27.6 | 31.1 | 25.8 | 25.1 |
| **time\_since\_last\_fault** | 49 | 52 | 42 | 70 | 15 |
| **fault\_duration\_avg** | 4.4 | 5.9 | 3.1 | 1.9 | 4.8 |
| **fault\_burst\_count** | 3 | 0 | 2 | 3 | 2 |
| **mean\_voltage\_6h** | 230.9 | 231.5 | 228.1 | 229.9 | 223.5 |
| **max\_faults\_24h** | 0 | 3 | 3 | 3 | 2 |
| **heartbeat\_gap\_std** | 0.58 | 0.55 | 0.2 | 0.13 | 0.15 |
| **target** | 0 | 0 | 1 | 1 | 1 |

# 5. Class Imbalance Strategy

Failures were rare (≈3-5% of samples), so we used a hybrid strategy:  
- SMOTE: Oversampled failure cases to synthetically increase class 1 representation  
- Clustering-based Undersampling: Clustered class 0 samples using KMeans and selected representative samples from each cluster to retain diversity while reducing majority class volume  
- Final class distribution: ≈50:50, balanced and representative

# 6. Model Architecture

- Base Models:  
 - XGBoost (baseline)  
 - LightGBM (fast and efficient)  
- Ensemble Strategy:  
 - Stacked ensemble using Logistic Regression as meta-learner  
 - Input: predicted probabilities from XGBoost and LightGBM  
 - Output: final ensemble prediction



# 7. Confidence Scores and Threshold Tuning

- For each prediction, tracked the probability of failure (confidence score)  
- Business-driven thresholding:  
 - Tuned probability threshold to maximize recall while controlling false positives  
 - Used precision-recall curves to identify optimal cutoff (e.g., 0.35 instead of default 0.5)  
- Confidence score enabled:  
 - Risk-based alerting  
 - Prioritization of high-risk events

# 8. Evaluation Metrics

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Precision | Recall | F1 Score | ROC-AUC |
| XGBoost | 0.76 | 0.82 | 0.79 | 0.91 |
| LightGBM | 0.73 | 0.84 | 0.78 | 0.90 |
| Ensemble | 0.75 | 0.88 | 0.81 | 0.93 |

Chosen metrics:  
- Recall: Critical for minimizing missed failures  
- F1 Score: Balances recall and precision  
- ROC-AUC: Overall model discrimination ability

# 9. Conclusion

We successfully built a robust predictive system for EV charger failures using time-series OCPP logs. Our hybrid resampling strategy, ensemble modeling, and confidence-based thresholding significantly improved failure detection performance. Future work includes temporal sequence modeling (e.g., LSTMs) and online learning for real-time updates.