PROJECT ABSTRACT

Energy Consumption Anomaly Detection using Smart Meter Data

An LSTM autoencoder trained on the UCI household power dataset learns normal electricity usage and raises explainable alerts on abnormal consumption.

01. Title

Energy Consumption Anomaly Detection using Smart Meter Data

02. Introduction / Background

Households waste electricity silently: a failing refrigerator compressor, a water heater stuck on, or a meter fault can inflate bills for months. Smart meters record the revealing data, but raw kilowatt readings mean nothing without a model of normal. This project learns that model — an LSTM autoencoder trained on four years of minute-level consumption from the public UCI Individual Household Electric Power Consumption dataset reconstructs expected load, and windows it cannot reconstruct well are flagged as anomalies with plain-language explanations.

03. Problem Statement

Abnormal consumption goes unnoticed because nobody watches the meter data continuously, and simple threshold alarms either miss slow drifts or drown users in false alerts. The project addresses this with unsupervised deep learning that models each household's normal daily/weekly patterns and raises calibrated, explainable alerts — showing expected vs actual curves and a likely-cause hint per event.

04. Objectives

- Prepare the UCI household dataset: cleaning, 15-minute resampling, normalization.
- Train an LSTM autoencoder on anomaly-free periods to reconstruct normal 24 h load sequences.
- Calibrate the reconstruction-error threshold on validation data with a documented precision/recall trade-off.
- Classify events by shape (sustained elevation, spike, dropout) into likely-cause hints.
- Build an interactive demo: 7-day load chart with anomalies marked and an alert log.
- Evaluate detection precision/recall/F1 on held-out data with documented injected anomaly windows.

05. Proposed Methodology

Minute-level active power is resampled to 15-minute windows and normalized. A sequence-to-sequence LSTM autoencoder (2-layer encoder/decoder over 96-step daily sequences) trains only on normal periods to reconstruct expected consumption. Anomaly score = mean absolute reconstruction error; the alert threshold is chosen from the validation precision–recall curve. Consecutive flagged windows merge into events characterized by shape and mapped to cause hints. Evaluation uses injected anomaly windows in held-out data.

06. System Architecture / Working

Smart-meter readings → cleaning/resampling → LSTM autoencoder → reconstruction error per window → calibrated threshold → event merging → shape classifier → alert cards (window, severity, expected-vs-actual, cause hint) + interactive chart + exportable alert log. The demo visualizes a full week with the expected curve overlaid and each alert inspectable.

07. Hardware / Software Requirements

- Python 3.11, TensorFlow/Keras (LSTM autoencoder), pandas, NumPy, scikit-learn, Matplotlib
- UCI Individual Household Electric Power Consumption dataset (~2.07M readings, 2006–2010)
- Flask + HTML/CSS/JS demo with canvas charts
- Single GPU for training (~1–3 h); CPU inference under 2 s per day of data

08. Expected Outcomes

A detector that flags sustained elevations, spikes and dropouts with calibrated precision/recall/F1 reported as achieved figures after training (design targets until then); an explainable alert log; a documented retraining procedure for new households.

09. Applications

- Early warning for faulty appliances and stuck heating loads
- Meter-fault and data-gap detection for utilities
- Energy-waste awareness for households and small facilities
- Teaching unsupervised deep learning on real time-series data

10. Future Scope

- Per-appliance disaggregation (NILM) for exact fault attribution
- Multi-meter / feeder-level anomaly detection
- Adaptive thresholds that track seasonal drift
- SMS/push alert integration for real deployments