Electric Load Forecasting Using LSTM-Based Deep Learning Models

Authors: Sarah Kim, Ben Pilnick, Celena Kim, Emily Yoo
STAT 362 Final Project

I. Introduction

Understanding how electricity is used in a household is important, not just for lowering energy bills, but also for making smarter decisions about consumption and promoting sustainability. However, electricity usage is rarely straightforward because it fluctuates with daily routines, seasons, and appliance behavior, making it nonlinear and hard to predict with traditional time series methods. Deep learning, and especially models like LSTMs, is better suited for this kind of data because it can capture long-term patterns and dependencies in how power usage evolves over time.

In this project, we are focusing on forecasting household Global Active Power using the Individual Household Electric Power Consumption dataset from the UCI Machine Learning Repository. So far, we have developed a baseline LSTM model to evaluate how well it can learn from historical electricity data. This is a starting point so we can further explore more advanced architectures such as RNNs, GRUs, and Transformers.

II. Problem Statement

The Problem: Understanding electricity consumption is essential for optimizing energy usage, reducing costs, and supporting a more efficient management of a household's power. Our goal is to predict hourly household electricity usage (Global_active_power) using past consumption patterns and related electrical measurements.

Evaluation Plan: We will measure success quantitatively using RMSE and model accuracy to assess predictive performance across our models. Qualitatively, we'll visualize training and validation loss curves to observe learning behavior and convergence, and present tables comparing model performance metrics side by side to highlight which approach best balances accuracy and generalization.

Dataset:

- **Source:** UCI Machine Learning Repository
- **Duration:** Dec 2006 - Nov 2010
- **Total Records:** 2,075,259 minutes → resampled to 34,589 hourly records for easier analysis.
- **Target Variable:** Global_active_power (kilowatts)
- **Input Features:** Voltage, Global_reactive_power, Global_intensity, Sub_metering_1/2/3

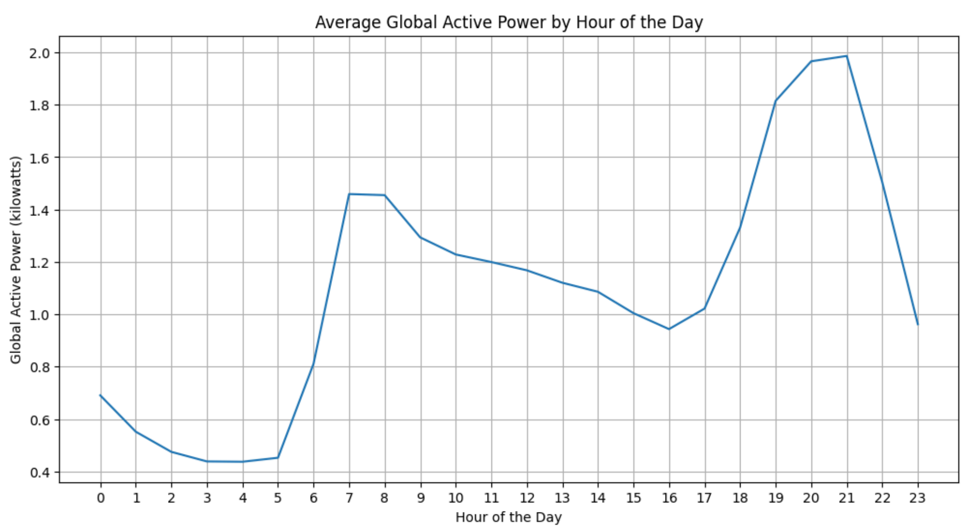
III. Technical Approach

We are using sequenced models such as RNNs, GRUs, and LSTMs. Similar to previous models we defined the training data as the first 70%, the validation data between 70% and 85%, and finally the test data as the rest. We then scaled on the train data for both the features and the output variable.

To create sliding windows we use 24 hours as the lookback period and predict 1 hour ahead. This will allow us to get predictions for every hour. For our baseline model we are using LSTM with one input layer, one LSTM layer with 100 neurons, and one output layer. We compiled the model using adam and mse.

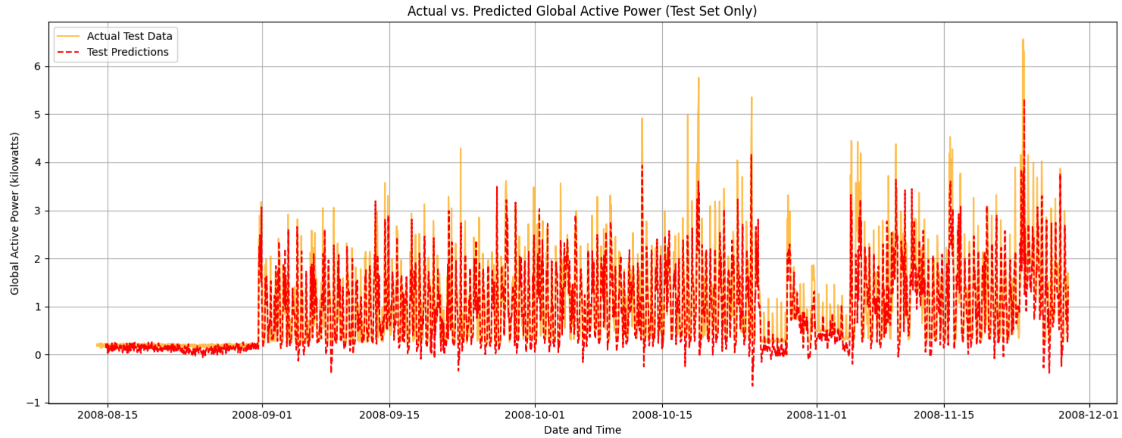
IV. Preliminary Results

Exploratory analysis was conducted first to better understand the dataset. First we looked at missing values and found 25,969 NA's from the Sub_metering_3 We filled these missing values with the forward fill feature from sklearn (i.e. replace the missing values by carrying the last known value forward). Second, we converted the dataset from every minute to every hour by taking the mean of each hour to reduce the size for deeper learning. We plotted average Global_Active_Power against each hour to look at the cyclicality of power consumption each day.



Looking at the chart we notice two peak consumption periods early morning (Hours 6-9) and late evening (Hours 18-21). Utilizing these trends and looking at the correlation between global active power between different times we identified our lookback period is 24 hours for generating window sequences and predicting 1 hour ahead.

Baseline Results: Using a LTSM deep learning model with windows of 24 hours and predicting 1 hour ahead we obtained a test RMSE of 0.551, train RMSE of 0.608, validation RMSE 0.516. We can see that the baseline model does a reasonable job at predicting actual data.



V. Next Steps

Preprocessing: Add cyclical encoding using sine and cosine functions for the following features: `hour`, `dayofweek`, `isweekend`, `month`, `hoursine`, `hourcosine`. Research has shown that using cyclical encoding for time based models like LTSM, GRUs, and RNNs can greatly improve performance to capture deep patterns within the dataset. Research

Model selection: Continue to improve our LTSM architecture. Analyzing validation loss and training loss, we observed that training loss did not decrease much after the first epoch. Thus, we have to change our model by adding more layers , dropout, or other optimization techniques. Create RNN and GRU architectures to compare to the performance of LTSM.