Design and Implementation of an Educational Electricity Load Forecasting System: A Modular and Adaptive Approach

David Santiago Téllez Melo (20242020107),
Ana Karina Roa Mora (20232020118),
Daniela Bustamante Guerra (20241020131),
Andrés Felipe Correa Méndez (20221020141)
Faculty of Engineering, Systems Engineering Program
Universidad Distrital Francisco José de Caldas
Bogotá, Colombia
Emails: {dstellezm, akroam, dabustamanteg, afcorream}@correo.udistrital.edu.co

***Abstract*—Electric load forecasting plays a crucial role in optimizing power grid operations and ensuring energy reliability. This paper presents a modular, reproducible forecasting framework based on PyTorch that integrates two complementary models: a Multilayer Perceptron (MLP) for tabular, feature-engineered data and a Long Short-Term Memory (LSTM) network for sequential modeling. The system design follows a multi-layer architecture, clearly separating data preprocessing, feature engineering, modeling, validation, and evaluation stages. This ensures flexibility, interpretability, and ease of debugging.**

The proposed approach incorporates calendar, temperature, and historical consumption data, along with derived indicators such as Heating and Cooling Degree Days (HDD/CDD), to enhance temporal feature richness. Models are trained and validated using a rolling-origin cross-validation scheme, which closely replicates real-world forecasting conditions.

Preliminary results demonstrate that this architecture achieves consistent accuracy and supports model interchangeability without modifying the pipeline structure. The proposed solution establishes a professional, reproducible workflow suitable for industrial deployment and academic evaluation.

I. Introduction

Electric load forecasting is a fundamental task for energy providers, enabling efficient grid management, cost optimization, and reliability in energy distribution. With the increasing integration of renewable energy sources and the variability of consumption patterns, accurate forecasting has become more challenging and crucial. Recent advances in machine learning and deep learning have significantly improved predictive capabilities, allowing models to capture nonlinear patterns and temporal dependencies in large datasets.

Traditional forecasting methods, such as ARIMA and exponential smoothing, often struggle to adapt to complex feature interactions and external variables such as temperature, holidays, or socioeconomic events. In contrast, neural networks — particularly Multilayer Perceptrons (MLP) and Long Short-Term Memory (LSTM) architectures — offer flexibility and robustness for modeling nonlinear and sequential relationships. However, implementing these models in a reproducible and maintainable way remains a challenge, especially when dealing with heterogeneous data sources and long-term deployment.

This work proposes a modular forecasting framework that separates each stage of the pipeline into clearly defined layers: data ingestion, feature engineering, model training, forecasting, and evaluation. Each layer is independently testable, reusable, and replaceable, which facilitates experimentation and scalability. The implementation relies on **PyTorch**, ensuring modern deep learning capabilities while maintaining transparency and reproducibility.

The system integrates various data sources, including historical load data, weather information, and calendar events, to create a rich set of engineered features. Using these, two models — an MLP and an LSTM — are trained and validated using rolling-origin cross-validation, a methodology that mimics real-world forecasting conditions by continuously shifting the training and validation windows.

Figure 1 illustrates the overall architecture of the proposed system, showing the data flow and modular organization of its components.

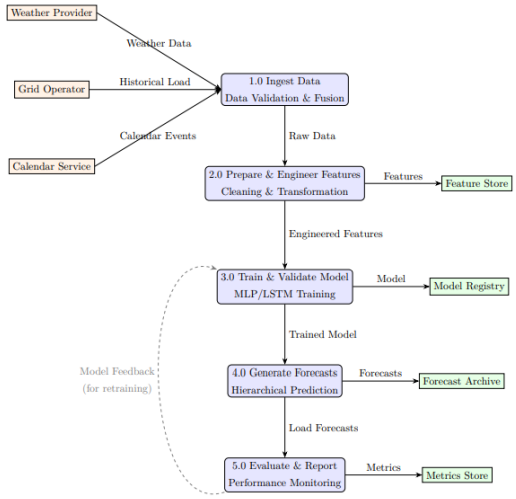

Figure 1: General architecture of the proposed load forecasting system. Each module handles a specific layer of the workflow, ensuring modularity and reproducibility.

II. Methods and Materials

This section describes the technical design of the proposed forecasting system, focusing on the data architecture, model implementation, and sensitivity management strategies. The approach was guided by systems engineering principles and the requirements identified in Workshop 1.

A. System Architecture Overview

The system is designed as a modular pipeline that transforms multi-source data into hierarchical electricity load forecasts. Each stage performs a well-defined function, allowing independent testing and development.

The architecture consists of five main stages:

- **Data Ingestion:** Collects and validates heterogeneous data sources, including weather feeds, historical load records, and calendar events.
- **Feature Engineering:** Generates predictive features such as temperature lags, calendar variables, and holiday patterns.
- **Model Training:** Trains MLP and LSTM architectures with hierarchical consistency constraints and temporal cross-validation.
- **Forecast Generation:** Produces zone-level and aggregated forecasts, including uncertainty quantification.
- **Evaluation and Monitoring:** Continuously measures model performance and triggers retraining when drift is detected.

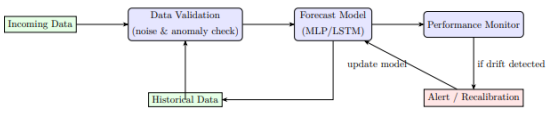

Figure 2: Integrated data ingestion, feature engineering, and forecasting workflow with feedback loop.

B. Clean ML Implementation

The system follows a *Clean ML Architecture*, ensuring separation of concerns and testability. The layers are organized as follows:

- **Data Layer:** Reads, cleans, and standardizes input datasets.
- **Feature Layer:** Adds weather, lagged, and calendar-based features.
- **Model Layer:** Implements PyTorch MLP/LSTM models with dropout regularization.
- **Validation Layer:** Performs rolling-origin cross-validation and calculates metrics such as RMSE and MAPE.
- **Application Layer:** Coordinates training, inference, and report generation.

This modular structure allows models to be replaced or updated without affecting the rest of the pipeline, facilitating experimentation and maintenance.

C. Modeling Approach

Two types of neural architectures were implemented:

1) **MLP (Multilayer Perceptron):** Used for tabular features. It applies Batch Normalization, ReLU activation, and Dropout to avoid overfitting.
2) **LSTM (Long Short-Term Memory):** Designed for temporal sequences, enabling the model to learn dependencies over extended time horizons.

Both models are evaluated using rolling-origin cross-validation to simulate real forecasting conditions, ensuring robustness and reliability.

D. Handling Sensitivity and Chaos

Electricity demand forecasting is inherently sensitive to small changes in temperature, calendar events, and data completeness. To address this, several stability mechanisms were incorporated:

- Continuous data validation and automatic smoothing of anomalies.
- Dropout regularization and ensemble averaging to mitigate overfitting.
- Feedback-based retraining triggered when model error exceeds thresholds.

Layer	Responsibility	Example Component
Data Layer	Read, clean, and standardize raw data.	`data_repo.py`
Feature Layer	Add calendar, weather, and lagged features.	`features/`
Model Layer	Define and train PyTorch models (MLP or LSTM).	`torch_models.py`, `torch_train.py`
Validation Layer	Handle rolling-origin cross-validation and metrics.	`cv.py`, `metrics.py`
Application Layer	Coordinate the workflow and generate the submission file.	`train.py`, `infer.py`

Figure 3: Sensitivity and chaos management loop for adaptive recalibration.

E. Technical Stack

The implementation uses a Minimal Clean ML stack centered on PyTorch, complemented by lightweight libraries:

- **pandas, NumPy:** Data manipulation and feature engineering.
- **scikit-learn:** Cross-validation and baseline comparisons.
- **matplotlib:** Visualization and diagnostic analysis.
- **holidays:** Automatic feature generation for national holidays.

III. Results & Discussion

IV. Conclusions

References

[1] A. Paszke, S. Gross, F. Massa *et al.*, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems*, 2019.

[2] G. Liu and D. Papalexopoulos, "Advanced methods for electricity load forecasting: A review," *Electric Power Systems Research*, vol. 189, 2020.

[3] T. Chawla, "Clean Machine Learning Architecture for Scalable Model Deployment," *arXiv preprint arXiv:2104.04541*, 2021.