Electricity Price Forecasting using Weather, Load, and Hybrid Deep Learning Models

Preksha Mishra : 2022ES11849
Course Project for ESL372
Department of Energy Science and Engineering, IIT Delhi

Data and Code Availability. All datasets and code used in this study are accessible via this Google Drive link: `https://drive.google.com/drive/folders/1kUt_TejZh7MGwGAhqvFBamoww6-Ti_W8?usp=sharing`

Abstract

Electricity markets are increasingly affected by variable demand, evolving generation portfolios, and weather uncertainty. Accurate short-term forecasting of electricity price is critical for market bidding, scheduling, and grid flexibility. This report presents an end-to-end forecasting pipeline that uses energy system data and weather variables to predict day-ahead electricity price. Multiple models are compared, including XGBoost, GRU, LSTM, CNN, CNN-LSTM, LSTM-Attention, and hybrid models combining neural networks with XGBoost residual correction. The results show that models incorporating temporal memory and learned temporal importance outperform purely regression-based approaches, while hybrid architectures further improve robustness by correcting residual nonlinear patterns.

1 Introduction

Electricity price forecasting plays a key role in energy trading, demand management, and reliability planning. Due to the integration of renewable resources and weather-driven demand variability, price behavior is often nonlinear and non-stationary. In this study, the forecasting problem is formulated as a supervised time-series regression task. The goal is to predict actual market price using historical price behavior, load patterns, generation mix, and relevant meteorological variables.

2 Data Description and Preprocessing

2.1 Data Sources

The analysis is based on the *Hourly Energy Demand, Generation, and Weather in Spain* dataset, obtained from Kaggle [1]. This dataset provides synchronized hourly observations of both power system states and meteorological variables, enabling the study of how market dynamics interact with weather conditions.

Electricity demand exhibits strong daily and weekly cycles, which are visible when examining the total system load over a typical two-week period (Figure 1). This temporal structure motivates the need for forecasting models capable of learning sequential patterns.

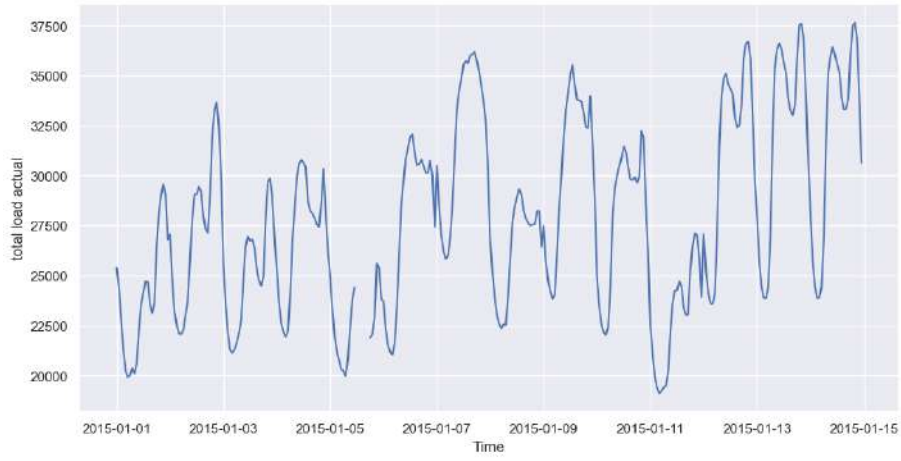

Figure 1: Two-week segment of the total actual system load. Clear daily oscillations justify the use of time-series forecasting.

Energy System Data

Weather Data

The complete list of variables for both datasets is provided in Appendix 6.

2.2 Data Cleaning

The raw datasets required several preprocessing steps to ensure consistency and physical plausibility.

- **Timestamp standardization and alignment:** All records were converted to hourly `datetime` index, and duplicate entries at identical (`time`, `city`) coordinates were removed.

- **Removal of non-informative fields:** Columns consisting of only zeros or redundant descriptive weather text were dropped.
- **Missing value handling:** Linear interpolation along the time axis was applied to preserve continuity.
- **Outlier correction (weather):** Implausible values such as `pressure` outside 870–1080 hPa and `wind_speed` values above 113 m/s were replaced and re-interpolated. The distributional effect of this cleaning is illustrated in Figures 3 and 4.
- **Fossil generation consolidation:** Related coal-based generation fields were combined to reduce redundancy. This decision is supported by strong feature correlations observed in Figure 2.

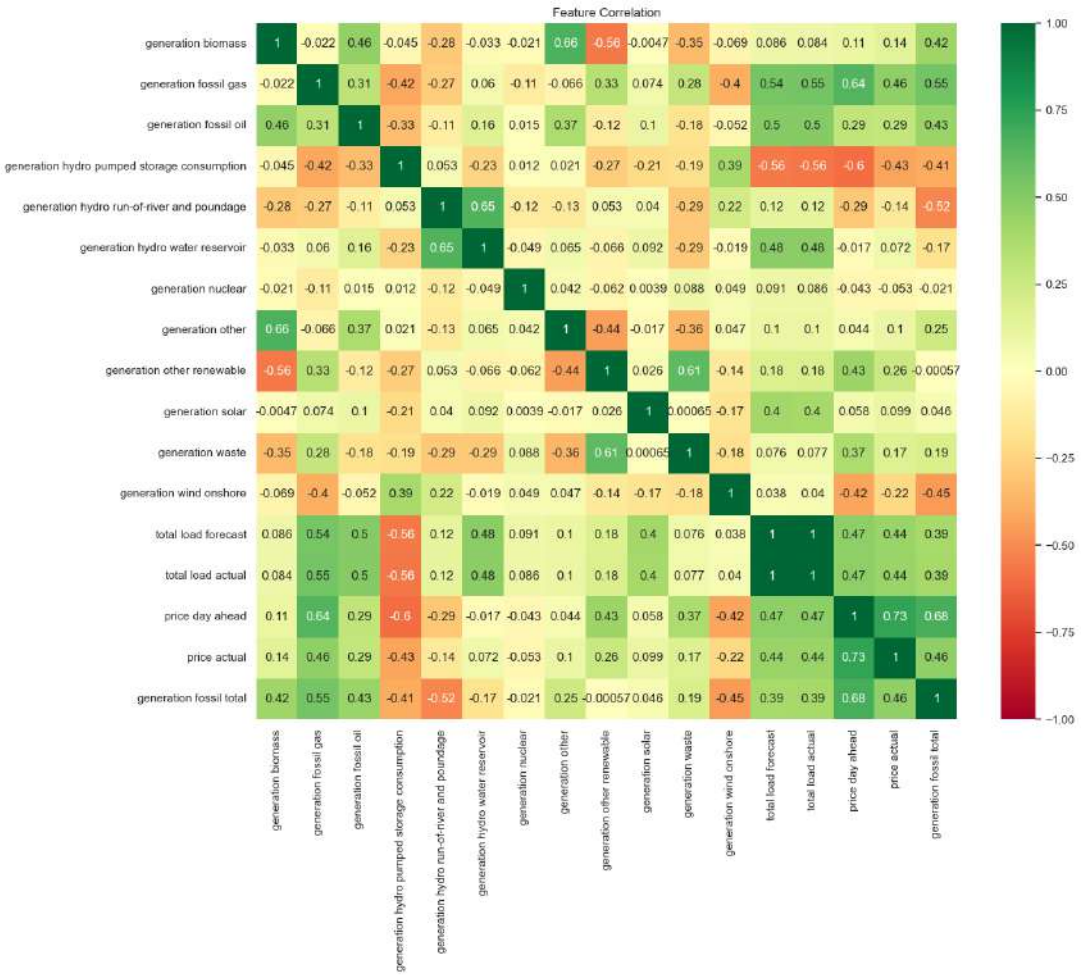

Figure 2: Correlation heatmap of the energy dataset after initial cleaning. Strong redundancy among fossil categories supports feature consolidation.

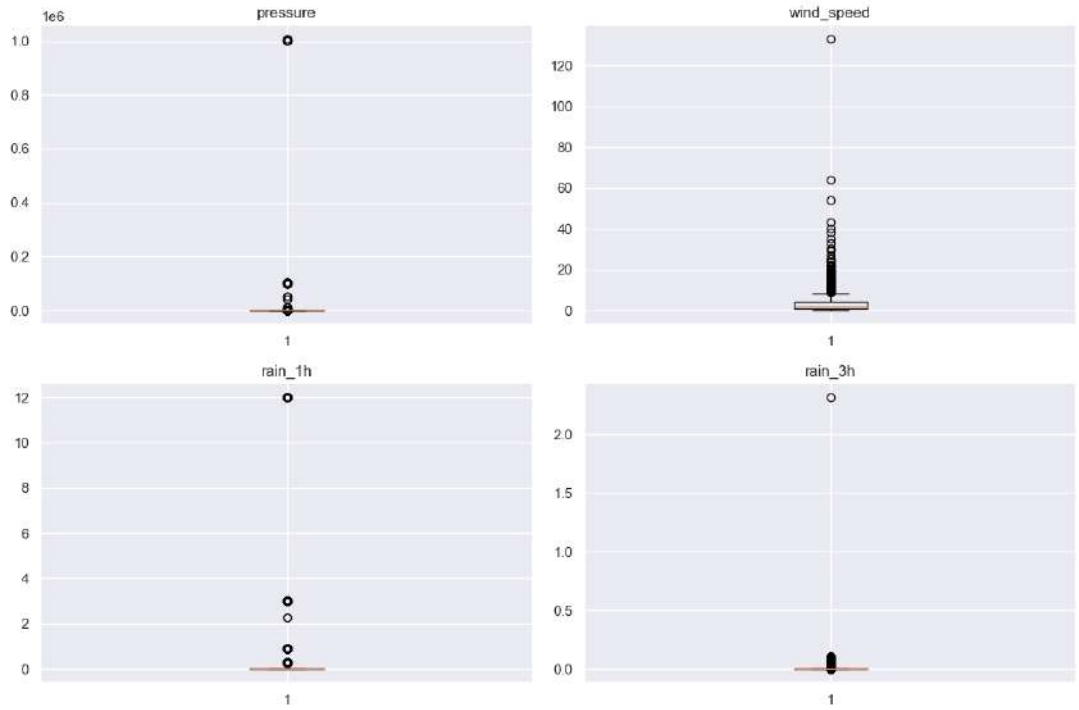

Figure 3: Weather feature distributions before outlier correction. Extreme pressure and wind values are visible.

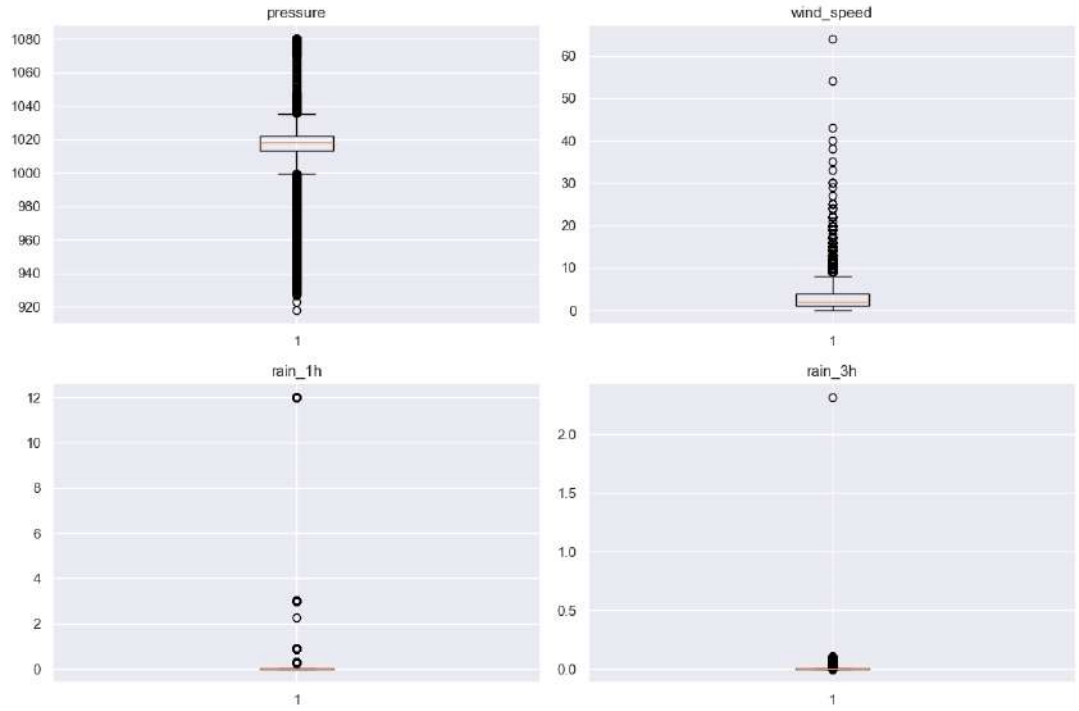

Figure 4: Weather feature distributions after outlier correction and interpolation. The cleaned distributions appear physically reasonable.

2.3 Feature Engineering

Additional signals were constructed to better model temporal and meteorological drivers of electricity prices.

- **Temporal decomposition:** Extracted `hour`, `weekday`, `month`, and `year`.
- **Cyclical time encoding:** Applied sine/cosine encoding to preserve periodicity.
- **Short-term weather dynamics:** Included temperature change rate `temp_gradient`.
- **Multi-city representation:** Weather variables were retained for all cities using city-specific suffixes.

The relationship structure among numeric weather variables is shown in Figure 5, supporting the selection of compact, non-redundant atmospheric predictors.

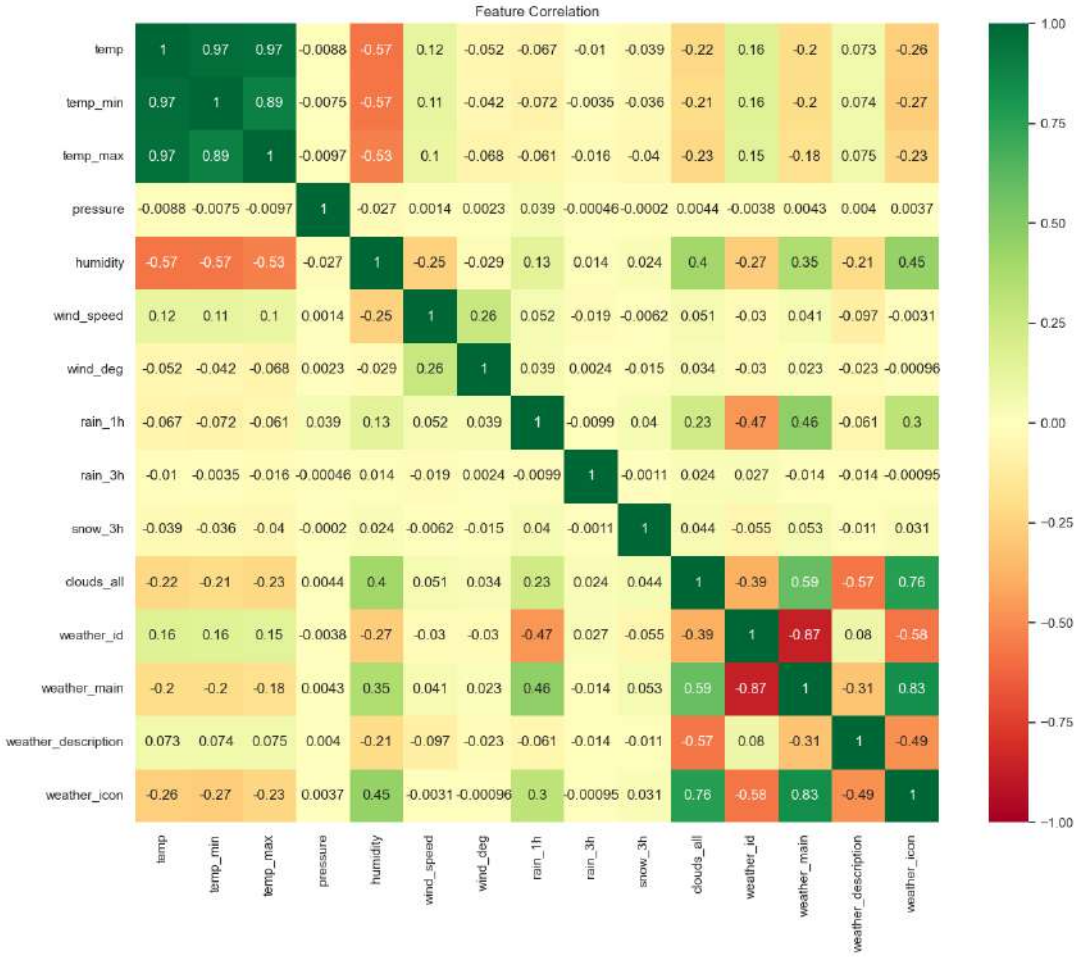

Figure 5: Correlation heatmap of numerical weather variables after cleaning.

2.4 Train–Validation–Test Split

A chronological split was used to prevent information leakage:

$$\texttt{train:val:test} = 80\% : 10\% : 10\%.$$

The split boundaries were visually verified using the `price actual` time series to ensure the test set lies strictly in the future relative to training.

3 Exploratory Data Analysis and Time-Series Characteristics

Before model development, exploratory data analysis was conducted to understand the temporal dynamics of electricity prices and their relationship to demand and weather conditions.

3.1 Price Trends and Temporal Structure

Figure 6 shows the actual market price together with 24-hour (daily) and 168-hour (weekly) rolling averages. The clear oscillatory structure reflects strong daily demand cycles, while slower undulations indicate seasonal effects across weeks. Such multi-scale variation suggests that forecasting models must be capable of capturing both short-term and long-term temporal dependencies.

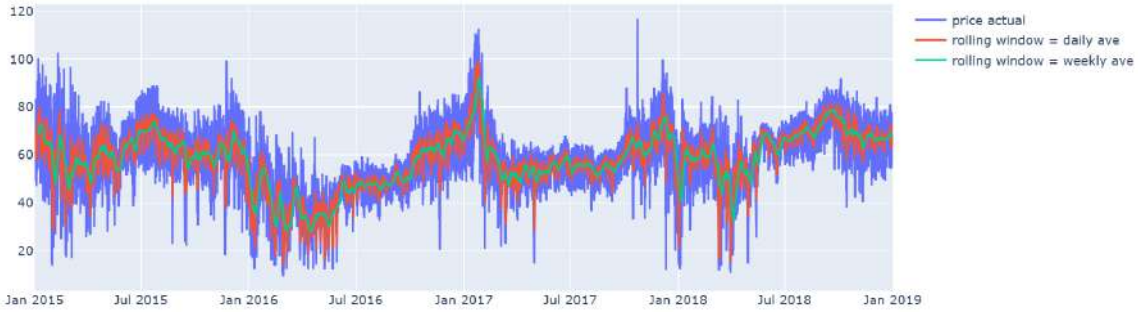

Figure 6: Actual electricity price with daily and weekly rolling means.

To further examine systematic variations, aggregated prices were compared across months and weekdays (Figure 7). Prices are higher in certain months and exhibit a recognizable weekday pattern, indicating that seasonal and behavioral factors influence market outcomes.

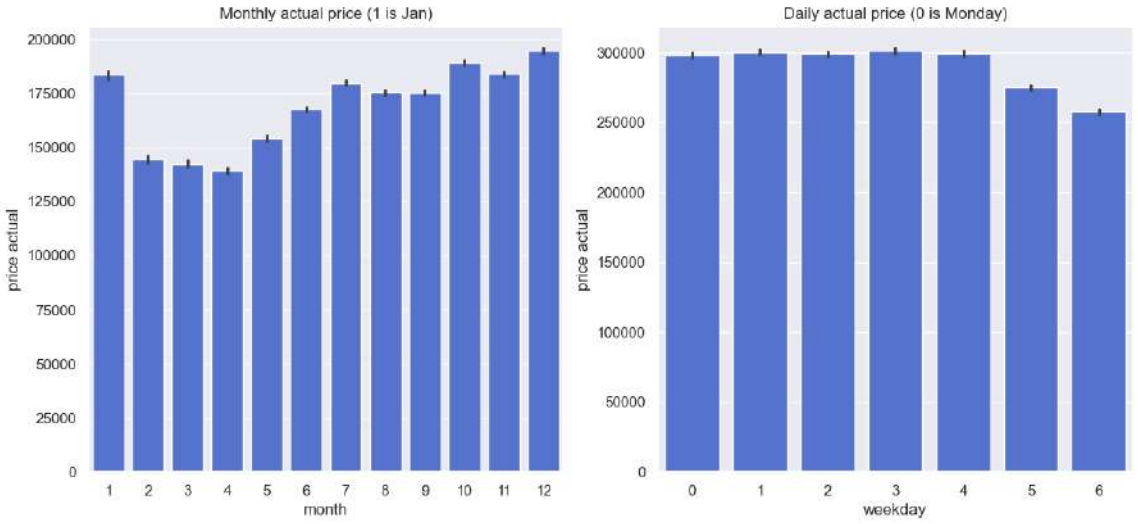

Figure 7: Monthly and weekday aggregation of actual electricity prices.

3.2 Forecast vs. Actual Distributions

To evaluate how closely operator forecasts reflect real system behavior, the distributions of the actual and day-ahead electricity prices were compared (Figure 8). After Min–Max normalization on the training range, the mean absolute error between the two series was approximately 0.07, indicating strong predictive alignment. This supports using the day-ahead price as a reference baseline when evaluating forecasting models.

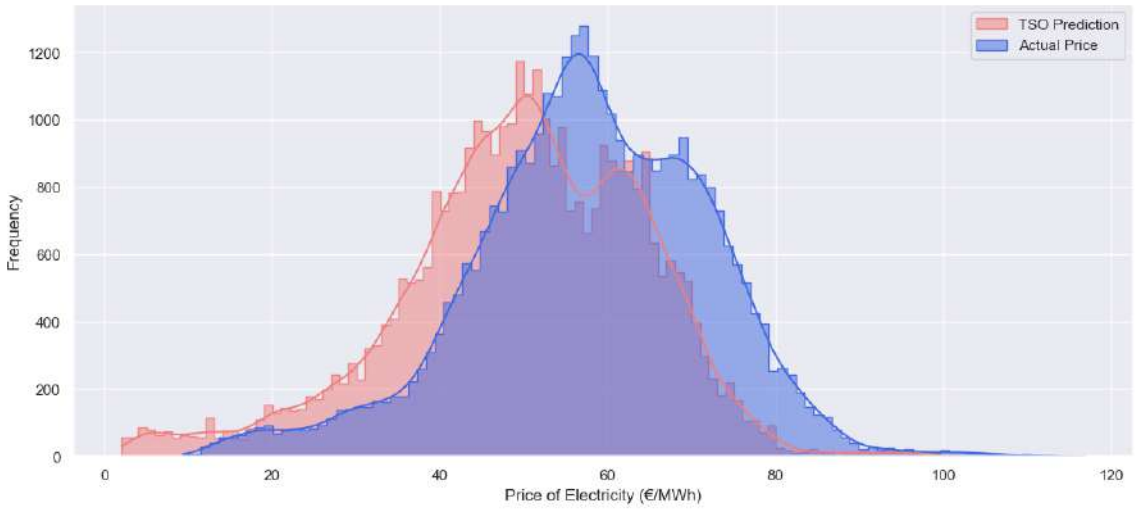

Figure 8: Distributions of actual electricity price and the system operator's day-ahead price forecast.

A similar comparison was performed for system load (Figure 9). In contrast to the price case, the forecasted load distribution was nearly indistinguishable from the actual load distribution and did not provide additional predictive variance relevant to price formation. Therefore, the feature `total_load_forecast` was excluded from subsequent

modeling.

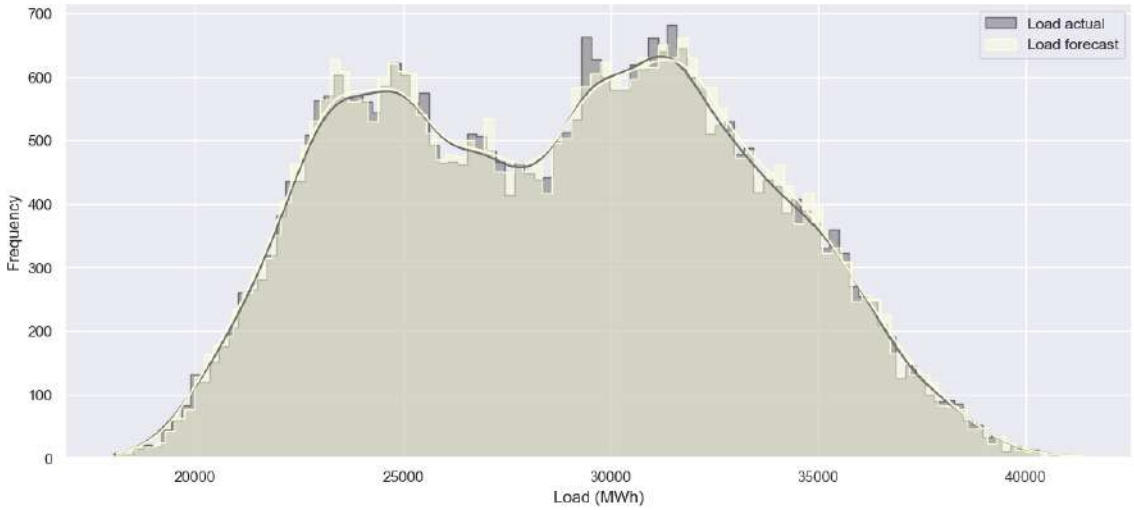

Figure 9: Distributions of actual system load and forecasted system load.

3.3 Seasonal Decomposition

The actual price time series was decomposed into trend, seasonal, and residual components, as shown in Figure 10. The seasonal component displays short-period fluctuations, while the trend reflects broader market shifts. The presence of a non-flat residual indicates remaining variability that learning models must capture.

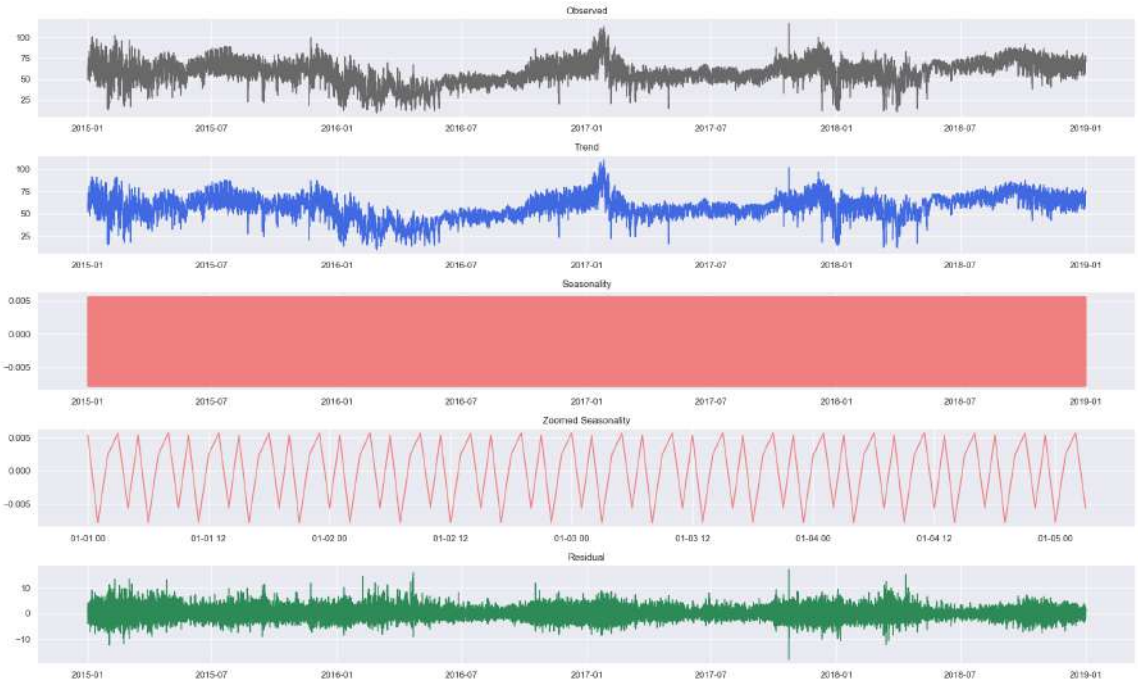

Figure 10: Additive decomposition of the electricity price into trend, seasonal, and residual components.

3.4 Stationarity and Autocorrelation

The Augmented Dickey–Fuller test indicated that the price series is non-stationary (p-value > 0.05). The Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots (Figure 11) show strong temporal dependence across multiple lags. This confirms that price evolution is history-dependent and justifies the use of sequence models such as LSTM, GRU, and hybrid architectures.

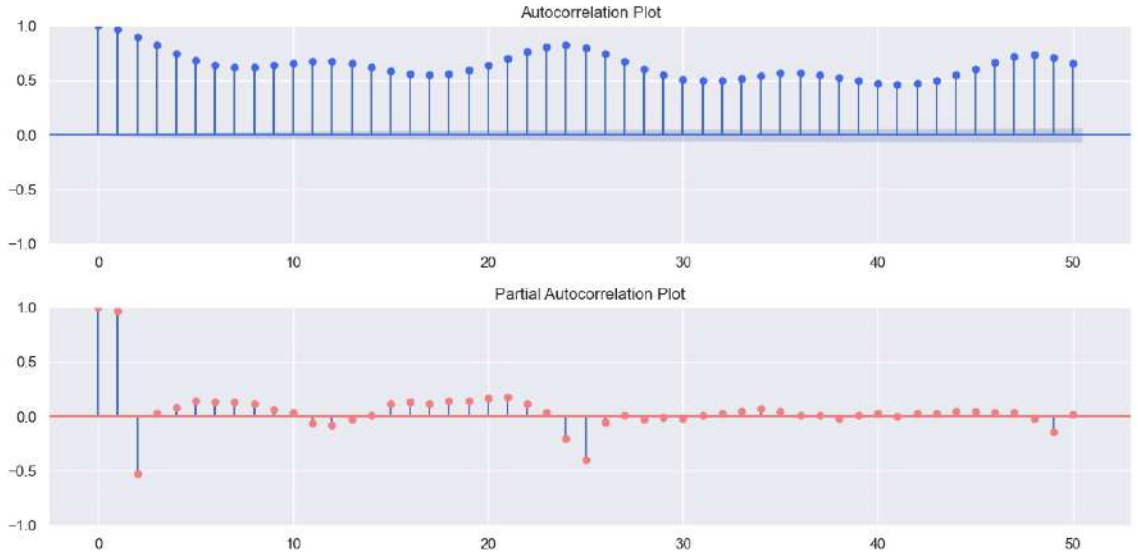

Figure 11: Autocorrelation (ACF) and partial autocorrelation (PACF) of the actual electricity price.

3.5 Dimensionality Reduction and Sequence Construction

Principal Component Analysis (PCA) retained components explaining 80% of variance, substantially reducing feature dimensionality while preserving information. The time series was then reshaped into sliding windows of 24 hours, enabling models to learn price evolution from preceding daily behavior. The final train–validation–test segmentation is illustrated in Figure 12.

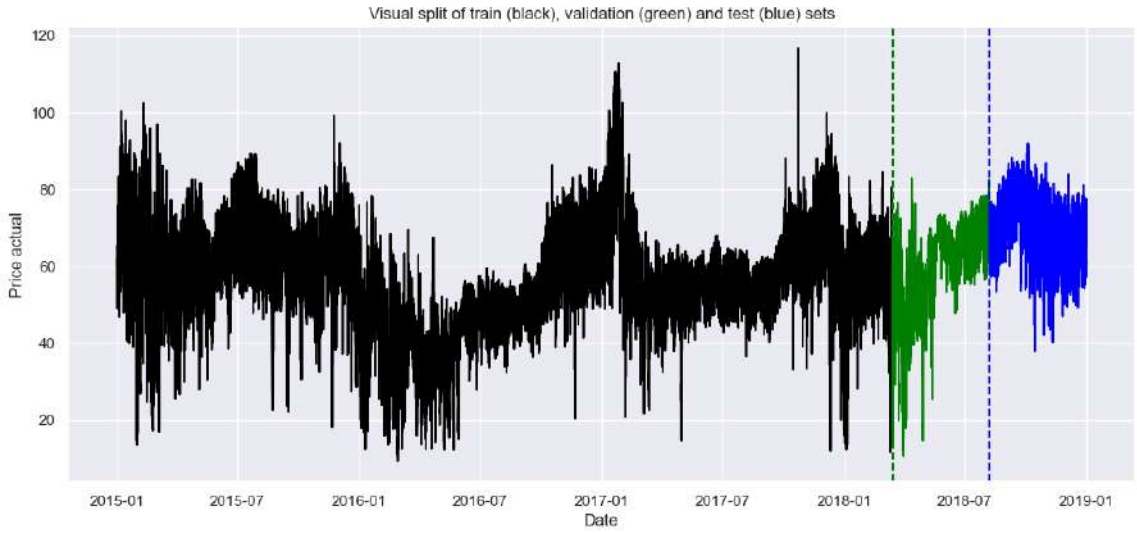

Figure 12: Train–validation–test time segmentation of the actual price series.

4 Forecasting Models and Training Strategy

Electricity price forecasting is a temporally structured regression problem, where the future price depends on recent market behavior, system load dynamics, and weather-driven renewable generation conditions. To capture these dependencies, the models are presented in increasing order of representational capacity, beginning from non-sequential decision tree ensembles and progressing to recurrent, attention-based, and hybrid architectures that explicitly combine sequential modeling with nonlinear residual learning.

All models are trained on 24-hour sliding windows of input features, predicting the price of the next hour. Training uses MAE loss, chosen for its robustness to sharp price spikes relative to MSE. Model performance is evaluated on the held-out test set, which reflects unseen future system behavior.

4.1 XGBoost (Non-Sequential Baseline Model)

XGBoost is used as the baseline model because it is highly effective at learning complex relationships between many input variables. Instead of learning from the sequence directly, it learns how different weather, load, and generation conditions relate to price by combining many small decision trees. Each new tree focuses on correcting the mistakes made by the previous ones, so the model gradually improves its understanding of the data.

However, since XGBoost does not have a memory of past time steps, the 24-hour history window is flattened into a single feature vector before training. This allows the model to see recent conditions, but it cannot explicitly learn how patterns evolve over time.

This makes XGBoost a meaningful baseline: if a more complex sequential model does not outperform it, then the added complexity is not justified.

Performance: The model achieves an MAE of 0.016, which is a large improvement over the day-ahead market forecast (MAE = 0.078). This shows that price can be predicted more accurately when weather and system variables are accounted for, but the model still struggles with sudden price spikes due to its lack of temporal memory.

Figure 13 shows the model predictions versus actual prices, and the training and validation error curves.

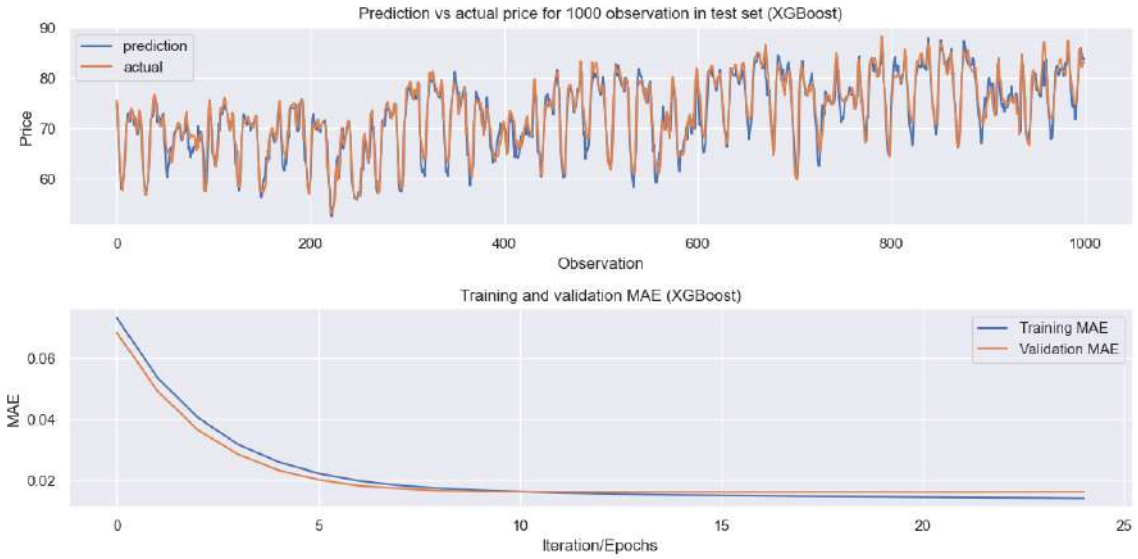

Figure 13: Prediction vs Actual and Training vs Validation Loss Curves for XGBoost

4.2 GRU (Recurrent Model for Temporal Memory)

The GRU model is introduced to directly learn patterns that unfold over time, something XGBoost cannot do. Instead of treating each 24-hour window as a fixed block of numbers, GRU processes the sequence one hour at a time and learns which past hours are important to remember for predicting the next price. This makes GRU better suited to capturing daily usage cycles, ramp-up periods, and lingering effects of weather changes.

A bidirectional GRU is used so that the model learns both the lead-up to a price movement and the fading influence of older conditions. To train it effectively, a cyclical learning rate is applied, letting the optimizer periodically explore and refine different regions of the loss surface rather than settling too early (Figure 14).

Compared to XGBoost, GRU shows a clearer ability to track short-term fluctuations in price, especially during periods of rapid change (see prediction plot in Figure 15). However, because the model is relatively small, it does not fully capture all seasonal and longer-term effects present in the data.

Performance: MAE = 0.026. While GRU does learn meaningful temporal patterns,

its performance is weaker than XGBoost in this configuration, suggesting that the model would benefit from either additional capacity or architectural support (which motivates the LSTM, attention, and hybrid models that follow).

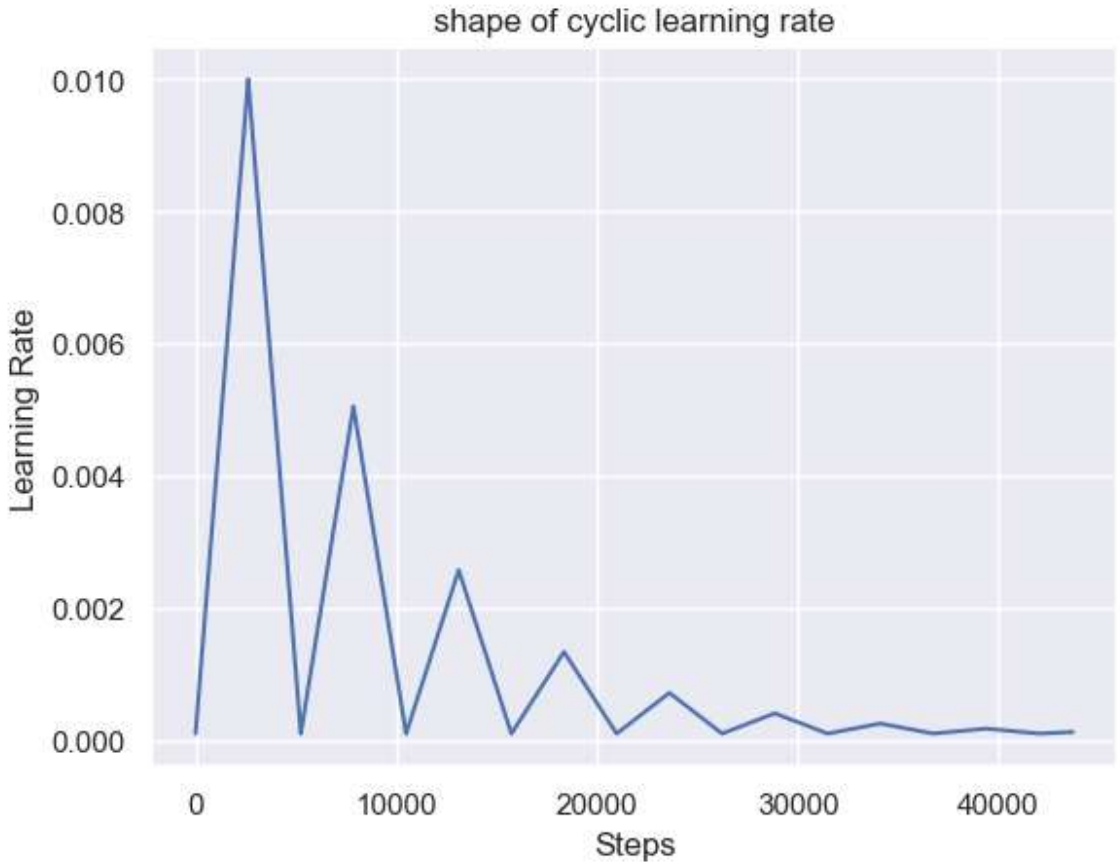

Figure 14: Cyclical Learning Rate Schedule for GRU

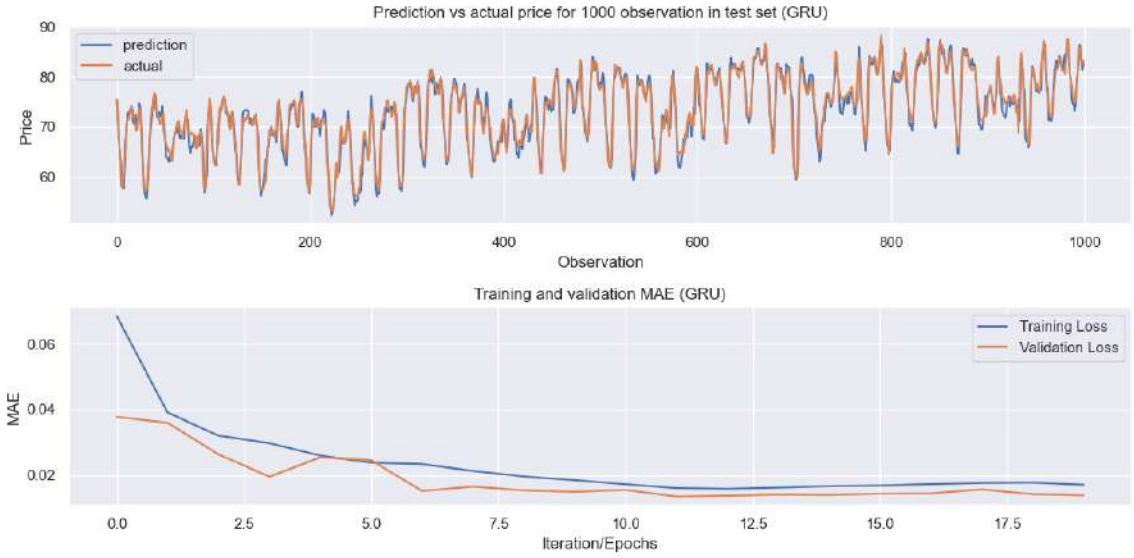

Figure 15: Prediction vs Actual and Training vs Validation Loss Curves for GRU

4.3 LSTM (Enhanced Temporal Memory)

The LSTM model extends the idea of the GRU by providing more control over how information is stored and forgotten across time. This allows it to retain relevant patterns that unfold over many hours or days, while discarding noise and short-lived fluctuations. In the context of electricity markets, this is useful because price movements are often influenced by gradual shifts in load, renewable availability, and market expectations—not just the immediate past.

Compared to the GRU, the LSTM shows a clearer ability to follow medium-range temporal trends in the price signal (Figure 16). This leads to smoother and more accurate predictions, especially during gradual increases or decreases in price. The training and validation curves (Figure 16) indicate stable convergence without overfitting, helped by early stopping and cyclical learning rate scheduling.

Performance: MAE = 0.014, the best among the purely neural models without hybrid correction. This suggests that electricity price dynamics include longer-range dependencies that the LSTM can represent more effectively than the GRU.

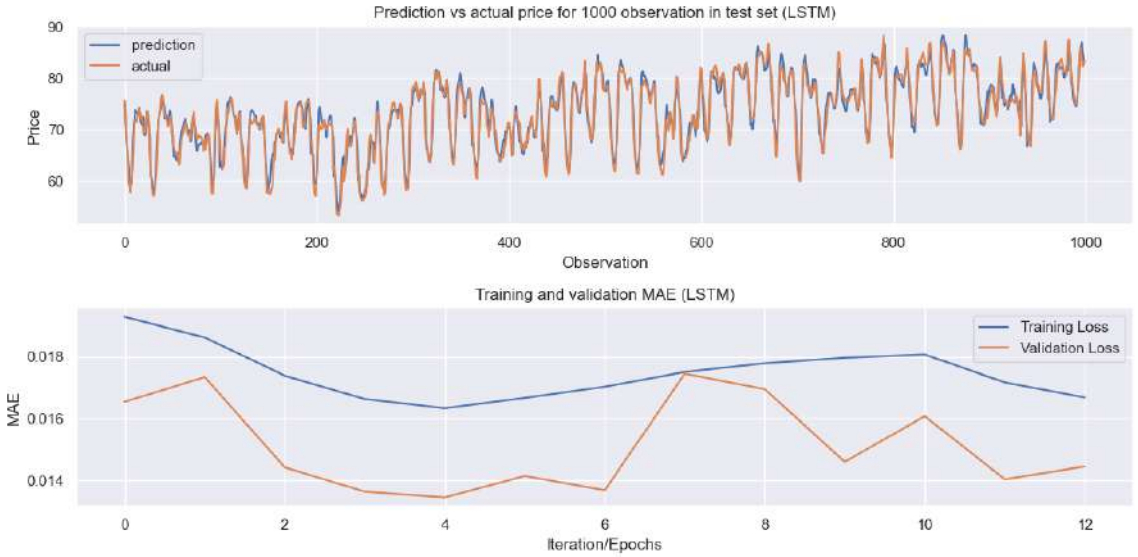

Figure 16: Prediction vs Actual and Training vs Validation Loss Curves for LSTM

4.4 CNN (Local Temporal Pattern Extractor)

The CNN model is included to evaluate how much of the electricity price signal can be explained by short-term local patterns alone. Instead of processing the sequence hour by hour like an RNN, the CNN slides small filters across the time window to detect repeated shapes such as rapid spikes, ramps, or sudden drops. This makes CNNs good at identifying short-term fluctuations that occur over a few hours.

However, CNNs do not have an internal memory of what came before or after these patterns. As a result, the model can recognize local price movements but cannot under-

stand broader temporal context—such as daily demand cycles or weather-driven trends that unfold gradually. Consequently, the CNN captures some meaningful short-term structure but struggles with sustained shifts in price levels.

Performance: MAE = 0.032, which is weaker than the recurrent models. This indicates that short-range patterns alone are not sufficient and reinforces the need for architectures that explicitly model temporal dependence.

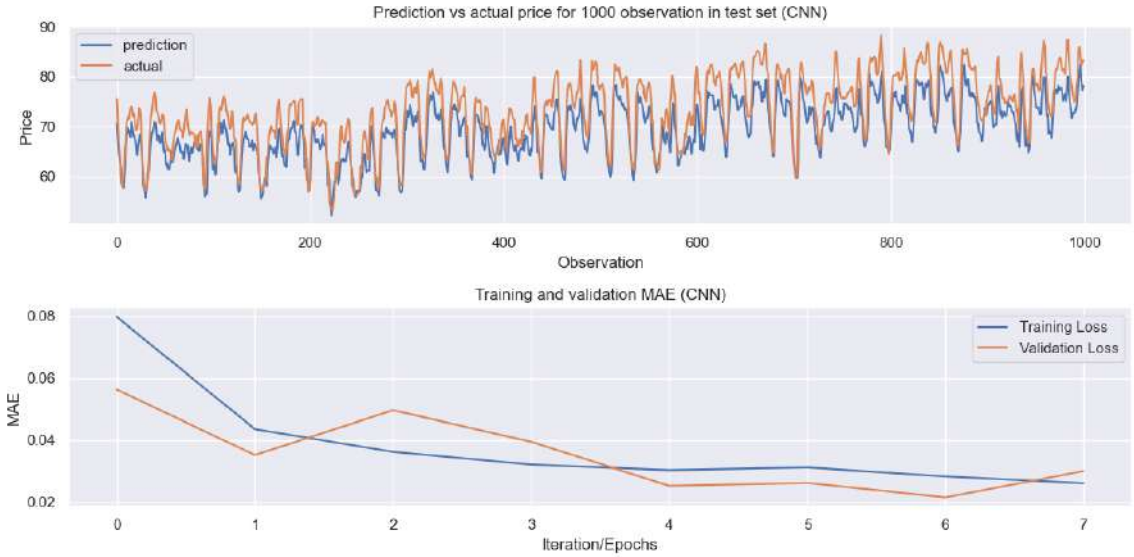

Figure 17: Prediction vs Actual and Training vs Validation Loss Curves for CNN

4.5 CNN–LSTM (Local Pattern Extraction + Long-Range Memory)

The CNN–LSTM combines the strengths of both convolutional and recurrent models. The CNN layer first scans the input window to extract short-term patterns, such as rapid ramps or fluctuations that occur over a few hours. The resulting feature map is then passed to an LSTM layer, which learns how these short-term patterns evolve over time and how they relate to longer-range trends in price behavior.

This two-stage structure allows the model to represent both local shape features and broader temporal dependencies—something neither CNN nor LSTM captures fully on its own. As a result, the CNN–LSTM shows improved responsiveness to changes compared to pure LSTM and better trend stability than pure CNN. The prediction curves (Figure 18) illustrate smoother tracking of price levels while still reacting to shorter fluctuations.

Performance: MAE = 0.022. This is better than CNN alone but not as strong as standalone LSTM. This suggests that while short-term patterns matter, the dominant drivers of electricity price variation still depend on temporal context that benefits from deeper recurrent memory.

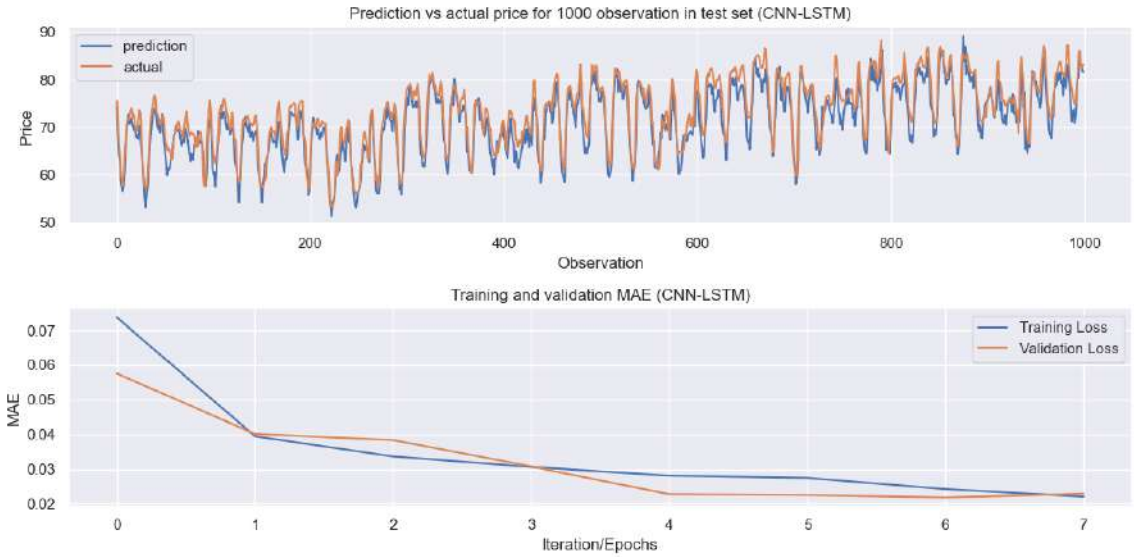

Figure 18: Prediction vs Actual and Training vs Validation Loss Curves for CNN-LSTM

4.6 LSTM with Attention (Learned Temporal Importance Weighting)

The LSTM with Attention extends the standard LSTM by learning which hours in the recent past are most relevant for predicting the next price. Instead of treating all time steps in the 24-hour window equally, the attention mechanism assigns a weight to each hour based on how informative it is for the current prediction. This allows the model to amplify key moments—such as sudden load changes or wind generation shifts—while down-weighting quieter, less influential periods.

In practice, this results in more accurate handling of sharp price movements and sudden ramps (Figure 19). The training curves (Figure 19) show stable convergence, indicating that attention improves learning efficiency by directing the model's focus.

Performance: MAE = 0.016, matching XGBoost while improving the interpretation of short-term fluctuations. This suggests that selectively weighting past time steps is more valuable than simply increasing model complexity.

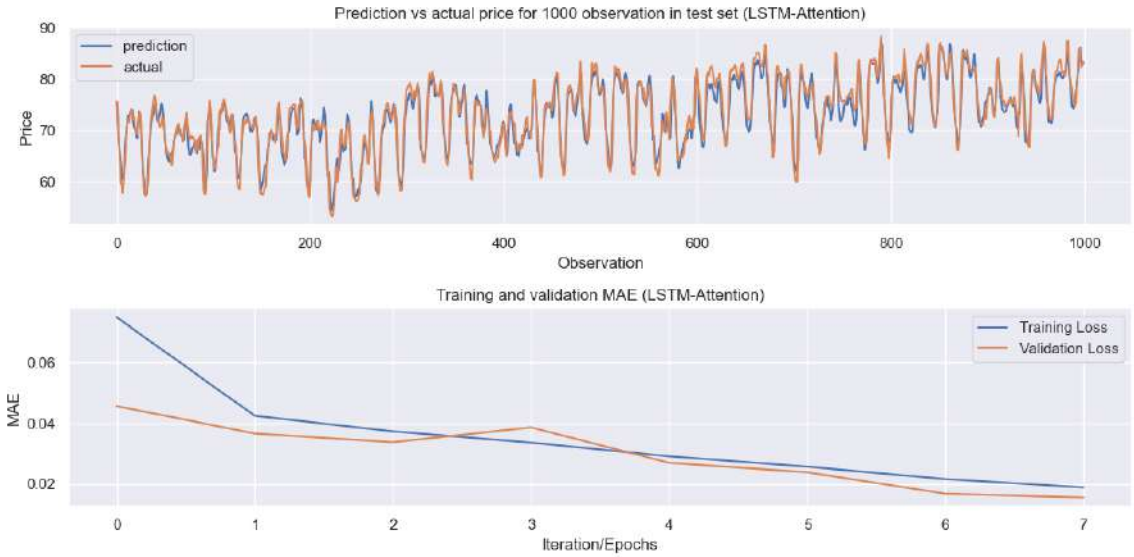

Figure 19: Prediction vs Actual and Training vs Validation Loss Curves for LSTM-Attention

4.7 Hybrid GRU–XGBoost (Temporal Memory + Residual Error Correction)

The hybrid GRU–XGBoost model combines sequence learning with nonlinear residual correction. First, the GRU captures the underlying temporal patterns in electricity prices by learning how recent history influences the next price step. However, neural recurrent models can struggle with sudden price spikes caused by short-lived market conditions (e.g., abrupt renewable output changes or balancing costs).

To address this, XGBoost is trained on the *residual errors* of the GRU (i.e., the difference between the GRU forecast and the actual price). In doing so, it learns the irregular, nonlinear components the GRU did not capture. The final prediction is the sum of:

$$\hat{y}_{\text{hybrid}} = \hat{y}_{\text{GRU}} + \hat{r}_{\text{XGBoost}}.$$

This structure allows:

- The GRU to model smooth temporal evolution,
- XGBoost to correct the remaining spikes and nonlinear effects.

The hybrid prediction plot (Figure 20) shows improved responsiveness to rapid price changes compared to the standalone GRU. Because the hybrid model is assembled in two independent stages, it does *not* have a single unified loss curve; therefore, the "training curve" panel in Figure 20 is intentionally empty. Each component is trained separately, so no single training loss applies to the combined model.

Performance: MAE = 0.015, a significant improvement over the standalone GRU

(0.026), demonstrating that residual correction effectively recovers predictive information missed by sequence-only models.

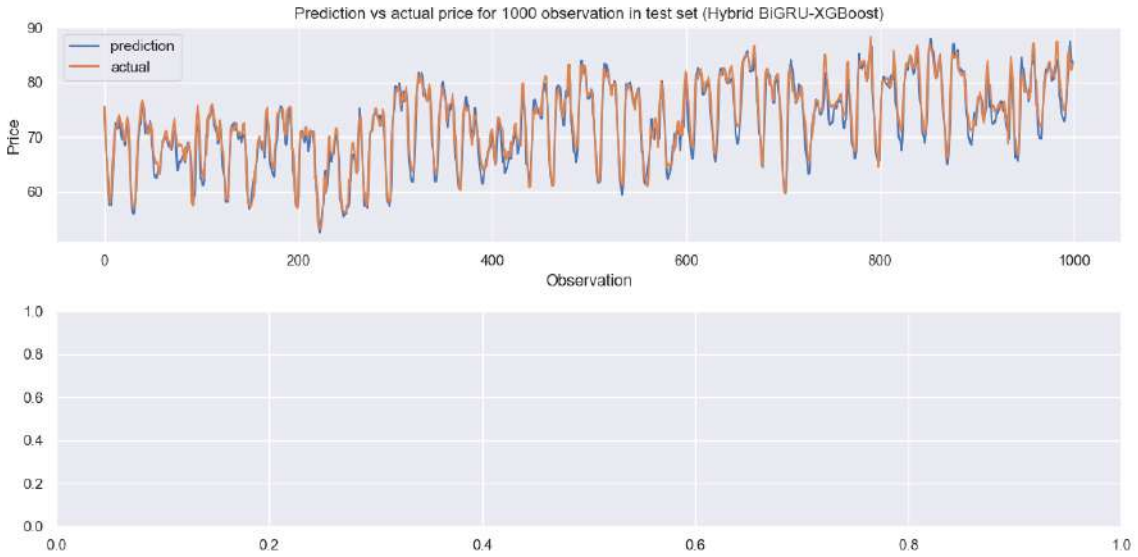

Figure 20: Prediction vs Actual and Training vs Validation Loss Curves for Hybrid GRU-XGBoost

4.8 Hybrid LSTM–Attention–XGBoost (Context-Aware Memory + Residual Learning)

The hybrid LSTM–Attention–XGBoost model extends the logic of the previous hybrid approach by beginning with a stronger base forecaster. The LSTM learns medium-range temporal dependencies, while the attention mechanism highlights the specific past hours most relevant for the next price value (e.g., recent volatility, ramp-up periods, or renewable output shifts). This yields a more context-aware sequence forecast compared to LSTM or GRU alone.

However, even LSTM with attention can miss sharp price jumps driven by short-lived system imbalances or market interventions. To capture these remaining non-smooth effects, XGBoost is trained on the *residual errors* of the LSTM–Attention model. The final prediction is:

$$\hat{y}_{\text{hybrid}} = \hat{y}_{\text{LSTM-Attention}} + \hat{r}_{\text{XGBoost}}.$$

This hybrid structure allows:

- The LSTM–Attention model to learn structured temporal patterns and emphasize the most relevant past timesteps.
- XGBoost to model irregular residual deviations that do not follow smooth temporal dynamics.

The resulting prediction plot (Figure 21) shows improved tracking of short-lived spikes compared to the standalone LSTM–Attention model. As with the GRU hybrid, the training and correction occur in two separate phases; therefore, there is no single combined training loss curve. For this reason, the second panel in Figure 21 is intentionally blank.

Performance: MAE = 0.014, matching the best non-hybrid model (plain LSTM) while providing sharper responsiveness to price fluctuations. This reflects that attention-driven temporal encoding already captures most nonlinear structure, and the residual correction primarily adds refinement.

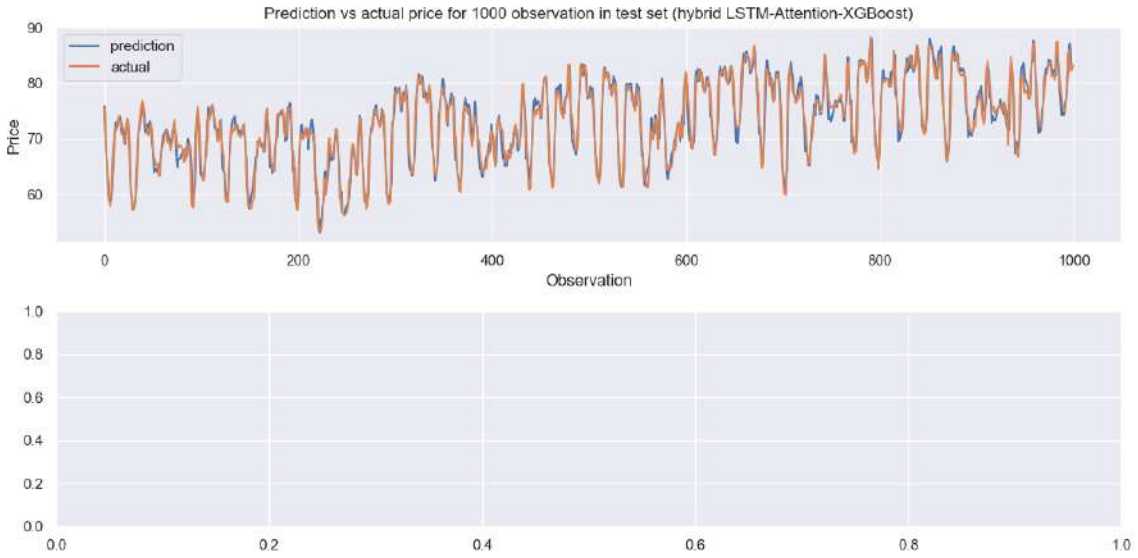

Figure 21: Prediction vs Actual and Training vs Validation Loss Curves for Hybrid LSTM-Attention-XGBoost

5 Performance Summary

Several architectures achieved strong performance, with the **LSTM** and **Hybrid LSTM–Attention–XGBoost** models both obtaining the lowest Mean Absolute Error (MAE) of **0.014**. While the LSTM alone is already highly effective at modeling temporal dependencies, the hybrid model provides improved robustness during price spikes and reduces residual nonlinear errors, offering better stability despite matching the same average error score.

Model	MAE	Strength	Limitation
TSO Forecast	0.078	Real-world baseline reference	Misses short-run fluctuations and price spikes
XGBoost	0.016	Learns strong static feature interactions	No temporal memory (sequence flattened)
GRU	0.026	Captures short-term sequential structure	Limited long-horizon dependency modeling
LSTM	**0.014**	Strong long-term temporal memory; stable forecasting	Can still under-react to sharp price spikes
CNN-LSTM	0.022	Learns both local and long-range temporal patterns	Does not explicitly emphasize important time steps
LSTM-Attention	0.016	Learns which past time steps matter most	Some nonlinear residual structure remains
Hybrid GRU-XGBoost	0.015	Residual correction improves GRU fit	Base GRU constraints remain
Hybrid LSTM-Attention-XGBoost	**0.014**	Most robust under volatile price regimes; best spike-handling	Higher computational cost due to two-stage training

Table 1: Forecasting model performance and qualitative behavior.

Although the LSTM and Hybrid LSTM–Attention–XGBoost models achieve the same MAE, the hybrid model demonstrates **better stability under high-volatility periods**. The attention mechanism highlights the most relevant past hours, while XGBoost models remaining nonlinear residual patterns. This results in more reliable forecasting during sudden load or generation shifts, even when the average error remains equal.

6 Conclusion

This study evaluated a spectrum of forecasting models ranging from tree-based regressors to recurrent and hybrid deep learning architectures for short-term electricity price prediction. While XGBoost demonstrated strong performance by capturing static relationships between weather, demand, and generation variables, its inability to model temporal structure limited its responsiveness to rapidly changing conditions. Recurrent models, particularly the LSTM, substantially improved predictive accuracy by learning medium- and long-range temporal dependencies inherent in electricity price dynamics.

The inclusion of an attention mechanism enabled the model to assign greater weight to the most informative recent hours, improving interpretability and responsiveness to short-term fluctuations. However, even attention-enhanced LSTMs left behind small, systematic nonlinear residuals. To address this, hybrid residual-learning models were introduced. The Hybrid LSTM–Attention–XGBoost model did not reduce the MAE beyond that of the standalone LSTM (both achieving **0.014**), but it provided **greater robustness under volatile periods** by correcting residual nonlinear effects the neural sequence model did not fully capture. In other words, the hybrid model improves *stability and spike sensitivity*, even when average error remains the same.

Overall, the findings show that the most effective forecasting strategies integrate:

1. **Temporal memory** (LSTM),
2. **Learned importance across past timesteps** (Attention),
3. **Residual nonlinear correction** (XGBoost).

Future extensions may explore probabilistic forecasting for uncertainty quantification, multi-horizon prediction frameworks suitable for real-time trading, and explainability techniques (e.g., SHAP for residual learning and attention-weight visualization) to support operational decision-making in electricity markets.

Appendix

A. Energy System Dataset Fields

Category	Fields
Time Index	time
Electricity Prices	price_day_ahead, price_actual
System Load	total_load_forecast, total_load_actual
Fossil Fuel Generation	generation_fossil_gas, generation_fossil_hard_coal, generation_fossil_brown_coal/lignite, generation_fossil_coal-derived_gas, generation_fossil_oil, generation_fossil_oil_shale, generation_fossil_peat
Renewable Generation	generation_wind_onshore, generation_wind_offshore, generation_solar, generation_hydro_run-of-river_and_poundage, generation_hydro_water_reservoir, generation_other_renewable
Other Generation Technologies	generation_nuclear, generation_biomass, generation_waste, generation_geothermal, generation_marine, generation_other
Hydro Pumped Storage	generation_hydro_pumped_storage_aggregated, generation_hydro_pumped_storage_consumption
Renewable Forecasts (Day-Ahead)	forecast_solar_day_ahead, forecast_wind_onshore_day_ahead, forecast_wind_offshore_day_ahead

Table 2: Grouped fields from the energy system dataset.

B. Weather Dataset Fields

Category	Fields
Time and Location	dt_iso, city_name
Temperature and Humidity	temp, temp_min, temp_max, humidity
Atmospheric Conditions	pressure, clouds_all
Wind Conditions	wind_speed, wind_deg
Precipitation Indicators	rain_1h, rain_3h, snow_3h
Weather Descriptors (Categorical)	weather_id, weather_main, weather_description, weather_icon

Table 3: Grouped fields from the weather dataset.

References

[1] N. J. Hana, "Hourly energy demand, generation, and weather in Spain," Kaggle, 2018. https://www.kaggle.com/datasets/nicholasjhana/energy-consumption-generation-prices-and-weather