Interpretable Deep Learning for Short-Term Electricity Load Forecasting with Calibrated Prediction Intervals

HUSSEIN AHMAD AHMAD[1, *], TAHA BENARBIA[2], SEYYED KASRA MORTAZAVI[1], MOHAMED EL BAHNASAWI[1], MAHMOUD HAMED[1], KYANDOGHERE KYAMAKYA[1]

[1]Institute for Smart Systems Technologies, Universitaet Klagenfurt, Klagenfurt,
AUSTRIA

[2]Industrial Engineering Department, University of Oran 2,
Oran,
ALGERIA

[*]*Corresponding author*

Abstract: Accurate electricity load forecasting is vital for power grid management, market operations, and renewable energy integration. This paper presents a comprehensive study on multivariate short-term load forecasting using deep learning models—specifically Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks with attention variants. We introduce a data-driven pipeline encompassing extensive exploratory analysis, advanced feature engineering, and systematic model refinement guided by permutation feature importance. Furthermore, we investigate uncertainty quantification (UQ) via Deep Ensembles and Monte Carlo (MC) Dropout to derive reliable prediction intervals. The refined multivariate LSTM demonstrates consistent gains over baseline implementations and robust predictive capability across short-term horizons, including day-ahead ($h=24$). UQ analysis indicates superior calibration and coverage with MC Dropout relative to Deep Ensembles, offering actionable insight for risk-aware decision making in modern energy systems.

Key-Words: short-term load forecasting, multivariate time series, deep learning, LSTM, GRU, uncertainty quantification, calibrated prediction intervals, permutation feature importance

Received: March 29, 2025. Revised: July 27, 2025. Accepted: August 29, 2025. Published: December 2, 2025.

1 Introduction

This work proposes a holistic, reproducible, and interpretable forecasting framework for multivariate short-term electricity load prediction. The framework integrates (i) exploratory analysis and domain-informed feature engineering, (ii) permutation-importance–guided model refinement, (iii) deep sequence models based on LSTM and GRU with attention variants, and (iv) uncertainty quantification via Deep Ensembles and Monte Carlo (MC) Dropout to provide calibrated prediction intervals for risk-aware operations.

Accurate short-term electricity load forecasting is a cornerstone of modern power system operations, directly impacting grid reliability, economic efficiency, and renewable energy integration [1], [2]. Growing penetration of distributed generation and demand-side flexibility further elevates the need for forecasting tools that deliver high accuracy together with reliable uncertainty estimates for operational decision-making.

Classical statistical models (e.g., ARIMA/SARIMA) offer transparency and strong theoretical foundations but rely on linear structure, which limits their ability to capture nonlinear demand dynamics and to scale with diverse exogenous inputs [3], [4]. Machine learning methods (e.g., support vector regression, random forests, gradient boosting) increase flexibility, yet often struggle with long temporal dependencies that characterize electricity demand. Deep learning models address these challenges by learning hierarchical representations directly from data: recurrent neural networks (RNNs), especially Long Short-Term Memory (LSTM) [5] and Gated Recurrent Unit (GRU) [6] architectures, mitigate vanishing gradients [7] and have become widely adopted for load forecasting. Attention mechanisms [8] and Transformer architectures [9] provide powerful alternatives, though in many operational settings LSTM/GRU models remain competitive due to efficiency and robustness at moderate sequence lengths.

Beyond point accuracy, there is increasing recognition that probabilistic forecasting and uncertainty quantification (UQ) are essential for risk-aware grid operations [10]. Two practical UQ approaches are particularly attractive in deep learning pipelines: MC Dropout [11], which retains stochastic dropout masks at inference to approximate predictive

distributions, and Deep Ensembles [12], which average independently trained models to improve robustness and capture epistemic uncertainty. Each method embodies a trade-off between coverage and sharpness; understanding these trade-offs in a unified experimental setting is crucial for deployment.

Despite substantial progress, the literature still exhibits gaps: few studies integrate feature refinement, attention-augmented recurrent models, and deep UQ within a single, reproducible pipeline suitable for practice. Moreover, the role of permutation feature importance (PFI) in shaping deep model inputs for both accuracy and calibration remains underexplored [13]. These gaps motivate our contribution.

Contributions. This work addresses these gaps through the following contributions:

- We develop an end-to-end, reproducible short-term load forecasting framework that integrates feature refinement, attention-augmented recurrent models, and uncertainty quantification in a single operational pipeline.
- We empirically evaluate attention-augmented LSTM and GRU architectures against their base counterparts, clarifying when attention helps under realistic sequence lengths and tuning budgets.
- We conduct a systematic comparison of MC Dropout [11] and Deep Ensembles [12] within the same experimental protocol, highlighting their calibration–sharpness trade-offs and operational implications.
- We demonstrate how interpretability via permutation feature importance and the use of predictive intervals can be jointly leveraged to support risk-aware decisions in electricity system operations.
- We include a high-level flowchart (Figure 1) that summarizes the integration of feature refinement, deep models, and UQ for transparent re-use and extension.

By unifying feature refinement, deep sequence modeling, and UQ within a transparent and reproducible framework (Figure 1), this study advances the practical deployment of deep learning for real-time electricity load forecasting.

Paper organization. Section 2 reviews related work on classical baselines, deep sequence models, and UQ methods. Section 3 details data preprocessing, model architectures, hyperparameter optimization, feature importance analysis, and UQ implementation.

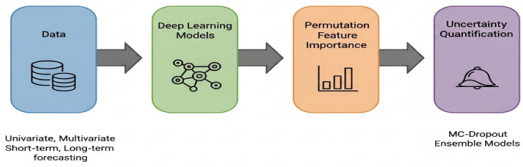


Figure 1: High-level flowchart of the proposed forecasting framework, integrating feature refinement, deep sequence models, and uncertainty quantification.

Source: Created by the authors.

Section 4 presents results and discussion across horizons, including accuracy, calibration, and operational interpretation. Section 5 concludes and outlines avenues for future research.

2 Background and Related Works

Electricity load forecasting is a mature field that has steadily evolved as data availability, market complexity, and computational resources have increased. This section reviews (i) classical statistical baselines, (ii) deep sequence models (RNN, LSTM, GRU, Attention), and (iii) Uncertainty Quantification (UQ) approaches relevant to robust short-term load forecasting (STLF).

2.1 Traditional Time Series Models

Classical models, such as the Autoregressive Integrated Moving Average (ARIMA) model, remain foundational benchmarks for univariate load forecasting because they explicitly model linear temporal dependencies, trends, and seasonality [3]. An ARIMA(p, d, q) process comprises autoregressive order p, differencing order d (to induce stationarity), and moving-average order q. For pronounced seasonal structure, the Seasonal ARIMA extension SARIMA$(p, d, q)(P, D, Q)_s$ augments with seasonal autoregressive P, seasonal differencing D, and seasonal moving-average Q, over period s (e.g., s=24 for hourly intra-day seasonality or s=168 for weekly cycles). These models offer transparency via their parametric link to the autocorrelation and partial autocorrelation functions. Their limitations include (a) linearity assumptions, (b) difficulty scaling to high-dimensional exogenous feature sets, and (c) reduced expressiveness for nonlinear, regime-switching, and interaction effects prevalent in

modern grids with demand response and renewable penetration [4]. While our work uses a basic ARIMA model as a simple baseline, its limitations in handling seasonality and complex data-driven features are noted, and a more comprehensive comparison with advanced statistical benchmarks like SARIMA is reserved for future work.

2.2 Deep Learning for Time Series Forecasting

Deep learning methods address nonlinearities and complex cross-feature interactions while enabling hierarchical feature abstraction. Key architectures are summarized below.

Recurrent Neural Networks (RNNs). Vanilla RNNs process a sequence $(x_1, \ldots, x_T)$ by updating a hidden state $h_t = \phi(W_h h_{t-1} + W_x x_t + b)$ that propagates temporal information. Training via backpropagation through time often suffers from vanishing/exploding gradients, impairing long-range dependency learning [7].

Long Short-Term Memory (LSTM). LSTMs [5] introduce gated pathways that regulate information flow, mitigating gradient decay. Given input x_t, previous hidden state h_{t-1}, and cell state C_{t-1}, the computations are:

$$\textbf{Forget gate:} \quad f_t = \sigma\big(W_f[h_{t-1}, x_t] + b_f\big) \tag{1}$$

$$\textbf{Input gate:} \quad i_t = \sigma\big(W_i[h_{t-1}, x_t] + b_i\big) \tag{2}$$

$$\textbf{Candidate cell:} \quad \tilde{C}_t = \tanh\big(W_C[h_{t-1}, x_t] + b_C\big) \tag{3}$$

$$\textbf{Cell update:} \quad C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{4}$$

$$\textbf{Output gate:} \quad o_t = \sigma\big(W_o[h_{t-1}, x_t] + b_o\big) \tag{5}$$

$$\textbf{Hidden state:} \quad h_t = o_t \odot \tanh(C_t) \tag{6}$$

Here, σ is the logistic sigmoid, $\odot$ denotes element-wise multiplication, and $[\cdot, \cdot]$ denotes concatenation. The architecture is illustrated in Figure 2.

Gated Recurrent Unit (GRU). GRUs [6] reduce parameter count by merging certain gates and states. The architecture is shown in Figure 3:

$$\textbf{Update gate:} \quad z_t = \sigma\big(W_z[h_{t-1}, x_t] + b_z\big) \tag{7}$$

$$\textbf{Reset gate:} \quad r_t = \sigma\big(W_r[h_{t-1}, x_t] + b_r\big) \tag{8}$$

$$\textbf{Candidate state:} \quad \tilde{h}_t = \tanh\big(W_h[r_t \odot h_{t-1}, x_t] + b_h\big) \tag{9}$$

$$\textbf{Hidden state:} \quad h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{10}$$

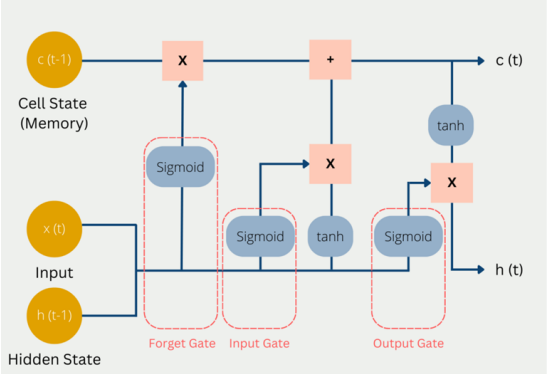


Figure 2: Long Short-Term Memory (LSTM) cell architecture showing forget (f_t), input (i_t), candidate ($\tilde{C}_t$), and output (o_t) gates that regulate the cell state (C_t) and hidden state (h_t).

Source: Based on [5].

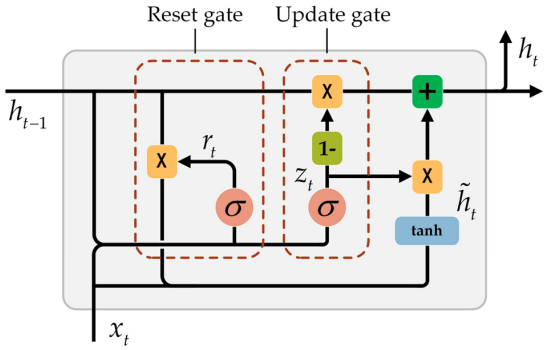


Figure 3: Gated Recurrent Unit (GRU) architecture showing update (z_t) and reset (r_t) gates, the candidate state ($\tilde{h}_t$), and the resulting hidden state (h_t).

Source: Based on [6].

Often achieving performance comparable to LSTMs with fewer parameters, GRUs can yield faster training and a reduced risk of overfitting in scenarios with limited data. Here, σ is the logistic sigmoid, $\odot$ denotes element-wise multiplication, $[\cdot, \cdot]$ denotes concatenation, and 1 is the all-ones vector (broadcast as needed).

Attention Mechanisms. Attention layers [8], [9] enhance sequence models by learning a relevance weighting over past hidden representations, enabling the model to focus on salient temporal positions (e.g., the same hour of the previous day/week). A schematic is given in Figure 4. For decoder time step t with decoder state h_t and encoder states h_i

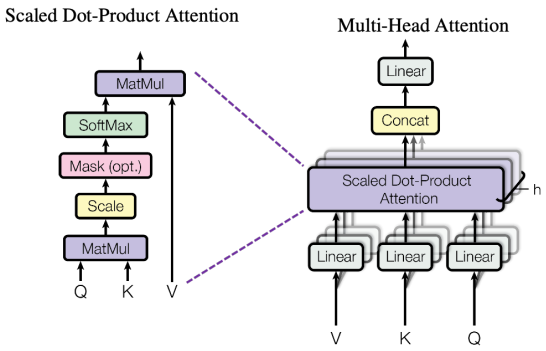


Figure 4: Multi-head attention illustration showing score computation ($e_{t,i}$), softmax weights ($\alpha_{t,i}$), and context aggregation into c_t.

Source: After [8], [9].

(i=1 . . . T):

$$e_{t,i} = \text{score}(h_t, h_i) \tag{11}$$

$$\alpha_{t,i} = \frac{\exp(e_{t,i})}{\sum_{k=1}^{T} \exp(e_{t,k})} \tag{12}$$

$$c_t = \sum_{i=1}^{T} \alpha_{t,i} h_i \tag{13}$$

The context vector c_t is concatenated with h_t and projected to the forecast output.

2.3 Uncertainty Quantification (UQ) in Deep Learning

Providing calibrated predictive intervals (PIs) alongside point forecasts is essential for operational risk management (e.g., reserve scheduling). We focus on two practical, widely adopted epistemic UQ approaches.

Monte Carlo Dropout (MC Dropout). MC Dropout [11] retains dropout masks during inference, performing T stochastic forward passes with parameters implicitly sampled via random neuron omission:

$$p(y \mid x, \mathcal{D}) \approx \int p(y \mid x, \omega) p(\omega \mid \mathcal{D})\, d\omega \tag{14}$$

$$\approx \frac{1}{T} \sum_{t=1}^{T} p(y \mid x, \omega_t) \tag{15}$$

with ω_t denoting the t-th dropout-mask instantiation. Empirical variance or quantile statistics across samples form prediction intervals. MC Dropout is computationally lightweight relative to training multiple full models.

Deep Ensembles. Deep Ensembles [12] train M independently initialized networks (potentially with data-order shuffling or bootstrapping). For input x, the predictive distribution is approximated by averaging:

$$p(y \mid x, \mathcal{D}) \approx \frac{1}{M} \sum_{m=1}^{M} p(y \mid x, \theta_m) \tag{16}$$

Variance (or quantiles) across members estimates epistemic uncertainty; aggregation typically improves robustness and calibration.

3 Methodology

Our approach to electricity load forecasting follows a rigorous multi-stage pipeline designed to (i) maximize point forecast accuracy and (ii) provide reliable uncertainty quantification.

3.1 Data Description and Preprocessing

The dataset comprises hourly records of electricity load, generation, and market prices from 2015 to 2018 [14]. The target variable is the actual total system load. Preprocessing ensures temporal integrity and feature richness. Initial exploratory analysis, as depicted in the correlation matrix (Figure 5), revealed strong linear relationships between key variables such as `total_load_actual` and `total_load_forecast`. This informed the initial feature selection and confirmed the presence of strong predictive signals within the dataset. Other variables, like `price_day_ahead` and `generation_nuclear`, also showed notable correlations with the target, highlighting their importance as potential predictors.

Consistent with recent STLF practice, we include exogenous drivers (e.g., temperature, rainfall, and day-ahead prices) alongside calendar encodings and multi-seasonal lags; recent work demonstrates gains from such covariates using a stacking model with a Lasso meta-learner on datasets from Australia and Spain [15].

1) Feature Engineering. Rich contextual signals were derived:

- **Temporal features:** `hour`, `day_of_week`, `month`, `day_of_year` encode daily, weekly, and annual seasonality.
- **Lagged features:** `load_lag_24h` (previous day same hour), `load_lag_168h` (previous week same hour) capture strong autoregressive structure.

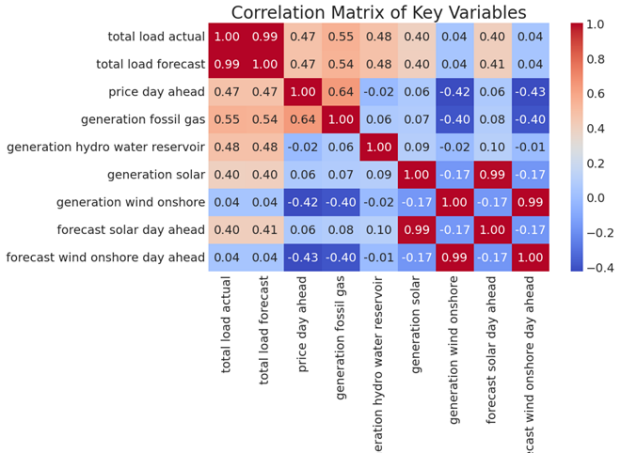


Figure 5: Correlation matrix of key variables in the dataset.

Source: Created by the authors.

- **Exogenous variables:** Supply and market context such as `total_load_forecast`, `forecast_solar_day_ahead`, `forecast_wind_onshore_day_ahead`, `price_day_ahead`, `price_actual`, `generation_fossil_hard_coal`, `generation_nuclear`, `generation_hydro_water_reservoir`, `generation_fossil_gas`.

2) Missing Values. Occasional gaps are imputed using time-series–aware methods (forward fill followed by linear interpolation for residual gaps) to preserve continuity without introducing look-ahead bias.

3) Scaling. All continuous predictors and the target are Min–Max scaled to $[0, 1]$ using statistics computed on the *training* split only; the fitted transform is then applied to validation and test to avoid leakage:

$$x' = \frac{x - \min_{\text{train}}(X)}{\max_{\text{train}}(X) - \min_{\text{train}}(X)}. \tag{17}$$

4) Chronological Split. Temporal ordering is preserved:

- Training: 17,522 hours (parameter learning),
- Validation: 8,760 hours (hyperparameter tuning / early stopping),
- Test: 8,782 hours (final generalization evaluation).

3.2 Model Architectures

All deep learning models were implemented in PyTorch [16] with data handling via `pandas` [17] and preprocessing utilities from `scikit-learn` [18]; visualization employed Matplotlib [19] and Seaborn [20]. We evaluate:

- **LSTM:** One or more stacked LSTM layers (Section 2) followed by a linear projection.
- **GRU:** Parameter-efficient recurrent alternative leveraging update/reset gates (Section 2).
- **LSTM+Attention / GRU+Attention:** Additive Bahdanau-style attention [8] applied over encoder hidden states to form a context vector prior to output projection (cf. Section 2). A detailed analysis of the attention mechanisms, including hyperparameter sensitivity and attention-weight visualizations, is reserved for future work.

3.3 Hyperparameter Optimization

Automated search used Optuna [21] (TPE sampler with pruning). Each trial executes a full train/validation cycle; validation Mean Absolute Error (MAE) guides selection.

Search Space and Optima (RNN Class).

- `hidden_size` $\in \{64, 128, 256\}$; optimum: 128,
- `num_layers` $\in \{1, 2, 3, 4\}$; optimum: 3,
- `dropout_rate` $\sim \mathcal{U}(0.1, 0.5)$; optimum: 0.295,
- `learning_rate` $\sim$ log-uniform $[10^{-5}, 10^{-2}]$; optimum: 5.84×10^{-3},
- Epochs: 30 with early stopping (patience 5) on validation MAE.

Training Objective. Mean Squared Error (MSE) minimized during training:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2. \tag{18}$$

Adam optimizer (default β parameters) is used with a fixed random seed for reproducibility.

3.4 Feature Importance and Refinement

Permutation Feature Importance (PFI) [13], [22] is computed on the best initial multivariate GRU by independently shuffling each feature over the validation window and measuring the increase in MAE; larger degradation implies higher importance.

Top 8 retained features after refinement:

- `total_load_forecast`
- `day_of_week`
- `hour`
- `load_lag_24h`
- `forecast_solar_day_ahead`
- `generation_hydro_water_reservoir`
- `day_of_year`
- `load_lag_168h`

Excluded low/negative-contribution variables (e.g., `generation_fossil_gas`, `generation_nuclear`) reduced input dimensionality, aiding generalization. The impact of refinement on interval calibration is left for future work.

3.5 Uncertainty Quantification Methods

We assess two epistemic UQ strategies on the refined models.

Monte Carlo Dropout (MC Dropout) [11]. Retain dropout during inference; run T stochastic forward passes producing $\{\hat{y}^{(t)}\}_{t=1}^{T}$. Mean and quantiles approximate the predictive distribution, capturing epistemic effects via mask variability.

Deep Ensembles [12]. Train $M{=}5$ independently initialized replicas. For input x, ensemble predictive mean and empirical variance arise from member outputs $\{\hat{y}^{(m)}\}_{m=1}^{M}$:

$$\bar{y} = \frac{1}{M}\sum_{m=1}^{M} \hat{y}^{(m)}. \tag{19}$$

Prediction intervals (e.g., 95%) use empirical quantiles of the member distribution.

Interval Construction. For both methods, 95% PIs are formed from the 2.5th and 97.5th percentiles (non-parametric). A more explicit decomposition of aleatoric vs. epistemic components is left to future work.

3.6 Evaluation Metrics

Model performance is assessed using a suite of point-forecast and uncertainty-quantification metrics. Let y_i be the actual value, $\hat{y}_i$ the point forecast, and N the number of observations in the test set. For uncertainty metrics, let L_i and U_i be the lower and upper bounds of the prediction interval, respectively.

Point Forecast Metrics We evaluate point forecasts using the following metrics. Let $\{(y_i, \hat{y}_i)\}_{i=1}^{N}$ denote the actual and predicted values over the *test* set of size N, and let $\{y_t^{\text{train}}\}_{t=1}^{T}$ denote the *training* series of length T. Denote the sample mean of the actuals by $\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$.

- **Mean Absolute Error (MAE):** Measures the average magnitude of errors (scale–dependent).

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}_i|\,. \tag{20}$$

- **Root Mean Squared Error (RMSE):** A quadratic scoring rule that penalizes larger errors more strongly.

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \big(y_i - \hat{y}_i\big)^2}\,. \tag{21}$$

- **Mean Absolute Percentage Error (MAPE):** Expresses accuracy as a percentage of the actual value.

$$\text{MAPE} = \frac{100\%}{N}\sum_{i=1}^{N} \left|\frac{y_i - \hat{y}_i}{y_i}\right|. \tag{22}$$

Note: Undefined when any $y_i = 0$. Common fixes include excluding zero targets or using $\left|\frac{y_i - \hat{y}_i}{\max(|y_i|, \varepsilon)}\right|$ with a small $\varepsilon > 0$.

- **Mean Absolute Scaled Error (MASE):** A scale–independent measure comparing the test MAE to the in-sample one-step naïve forecast error computed on the *training* partition [23].

$$\text{MASE} = \frac{\frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}_i|}{\frac{1}{T-1}\sum_{t=2}^{T} \left|y_t^{\text{train}} - y_{t-1}^{\text{train}}\right|}\,. \tag{23}$$

For seasonal data with period m, replace the denominator by

$$\frac{1}{T-m}\sum_{t=m+1}^{T} \left|y_t^{\text{train}} - y_{t-m}^{\text{train}}\right|. \tag{24}$$

- **Coefficient of Determination (R^2):** Proportion of variance in the target explained by the predictions (can be negative if the model underperforms the mean baseline; most interpretable when a model includes an intercept).

$$R^2 = 1 - \frac{\sum_{i=1}^{N}\big(y_i - \hat{y}_i\big)^2}{\sum_{i=1}^{N}\big(y_i - \bar{y}\big)^2}\,. \tag{25}$$

Table 1: MAE (MW) of Univariate Models

Model	h1	h6	h12	h24
ARIMA	5804.888	6586.747	6653.147	6657.045
LSTM	434.110	972.137	1316.426	1627.001
GRU	508.785	1014.924	1301.697	1669.757

Source: Created by the authors.

Uncertainty Quantification Metrics We assess interval quality over a test set of size N, where $[L_i, U_i]$ denotes a nominal $(1-\alpha)$ prediction interval for y_i.

- **Prediction Interval Coverage Probability (PICP):** Percentage of actual values that fall within their prediction intervals. For a nominal $(1-\alpha)$ interval (e.g., 95%), the target is PICP $\approx 1-\alpha$.

$$\text{PICP} = \frac{1}{N}\sum_{i=1}^{N} I\big(L_i \le y_i \le U_i\big). \tag{26}$$

- **Mean Prediction Interval Width (MPIW):** Average width of the prediction intervals. Lower is better *only when* coverage is close to the nominal level.

$$\text{MPIW} = \frac{1}{N}\sum_{i=1}^{N} \big(U_i - L_i\big). \tag{27}$$

Here, $I(\cdot)$ is the indicator function.

4 Results and Discussion

Our evaluation provides a comprehensive assessment of model performance—both point forecasts and uncertainty characteristics—across four representative forecast horizons: h=1 (1 hour ahead), h=6 (6 hours), h=12 (12 hours), and h=24 (24 hours).

4.1 Univariate Baselines

A univariate setup, where models only used historical values of `total_load_actual`, served as a fundamental baseline. We compared a basic ARIMA$(1,1,1)$ model with standalone LSTM and GRU networks (no exogenous inputs).

Table 1 shows ARIMA performs very poorly, failing to capture strong multi-seasonal structure (daily/weekly) without explicit seasonal terms (e.g., SARIMA). In contrast, univariate deep models leverage nonlinear temporal patterns. Notably, the univariate LSTM attained an exceptionally low h=1 MAE (434.110 MW), even lower than the refined multivariate LSTM's h=1 MAE (826.900 MW; see Section 4.4). This apparent discrepancy typically reflects differences in:

Table 2: MAE (MW) of Original Multivariate Models

Model	h1	h6	h12	h24
LSTM	1327.569	1851.344	2088.193	2073.183
GRU	1203.837	1525.013	1717.220	1919.911
LSTM+Attn	2094.392	2272.460	2177.144	2137.599
GRU+Attn	1223.798	1522.962	1796.910	1964.553

Source: Created by the authors.

- scaling/normalization contexts between experimental tracks;
- train/validation/test chronological segmentation (slight distribution shifts);
- inclusion of additional (possibly noisy) exogenous features in the multivariate scenario prior to refinement.

In operational settings, multivariate models are generally preferred because external drivers (forecasts of load, weather, market prices) materially influence demand and improve robustness for medium horizons.

Note on baselines. Here, ARIMA$(1,1,1)$ serves as a minimal performance floor for linear models; a fuller classical comparison (e.g., SARIMA with seasonal terms) is left to future work.

4.2 Initial Multivariate Performance

Initial deep models were trained on the full 15-feature multivariate set. We report MAE (MW); other metrics (RMSE, MAPE, R^2, MASE) followed similar relative rankings and are omitted for brevity.

Table 2 summarizes the initial multivariate MAE by horizon. From Table 2, GRU yields the lowest MAE at all horizons, indicating strong capacity to capture nonlinear autoregressive structure with the unreduced feature set. Attention-augmented variants did not consistently outperform their base counterparts; *LSTM+Attention* underperformed markedly, suggesting: (i) added attention parameters were not optimally tuned, (ii) potential overfitting to noisy or redundant inputs, or (iii) insufficient sequence-length context to exploit attention benefits. These outcomes are consistent with broader empirical comparisons of sequence-modeling architectures, where convolutional and recurrent networks exhibit complementary strengths across tasks [24]. A plausible contributing factor is that strong engineered lags (e.g., `load_lag_24h`, `load_lag_168h`) already encode the most salient temporal dependencies, reducing the marginal benefit of learned attention under the current sequence length and tuning budget.

4.3 Permutation Feature Importance

Permutation Feature Importance (PFI) was computed for the best initial model (GRU) on h=24 to guide

Table 3: Permutation Feature Importance for GRU (h=24): Increase in MAE (MW)

Feature	ΔMAE
total_load_forecast	1516.967
day_of_week	522.915
hour	272.817
load_lag_24h	251.579
forecast_solar_day_ahead	171.593
generation_hydro_water_reservoir	151.867
day_of_year	84.088
load_lag_168h	65.749
price_day_ahead	57.431
month	26.136
generation_fossil_hard_coal	24.199
forecast_wind_onshore_day_ahead	18.900
price_actual	14.007
generation_fossil_gas	-18.178
generation_nuclear	-23.289

Source: Created by the authors.

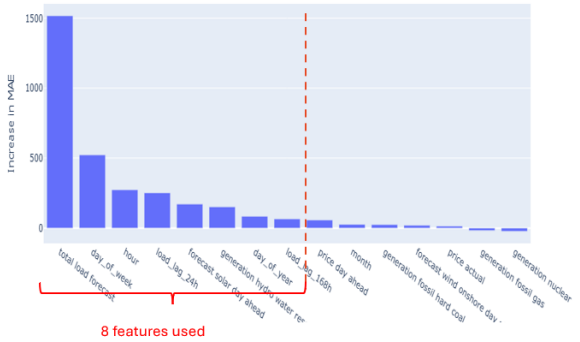


Figure 6: Permutation feature importance for GRU at h=24. Higher ΔMAE indicates greater importance.

Source: Created by the authors.

feature refinement.

Results (Table 3) and the corresponding bar plot (Figure 6) show `total_load_forecast` dominates predictive signal. Temporal (`day_of_week`, `hour`, `day_of_year`, `month`) and lagged autoregressive features (`load_lag_24h`, `load_lag_168h`) are also critical, reflecting multi-seasonality. Negative importance values (e.g., `generation_fossil_gas`, `generation_nuclear`) imply that random permutation improved validation MAE—an indication of redundancy or noise injection relative to stronger correlated predictors (notably `total_load_forecast`). These were candidates for removal.

4.4 Refined Feature Set Performance

All models were retrained using the top 8 features: `total_load_forecast`, `day_of_week`, `hour`, `day_of_year`, `load_lag_24h`, `load_lag_168h`, `forecast_solar_day_ahead`, `generation_hydro_water_reservoir`.

Table 4: MAE (MW) of Refined Multivariate Models

Model	h1	h6	h12	h24
LSTM (Refined)	826.900	1249.350	1494.019	1674.688
GRU (Refined)	979.392	1385.315	1550.875	1735.391
LSTM+Attn (Ref.)	1574.835	1689.637	1884.772	1932.895
GRU+Attn (Ref.)	906.056	1343.569	1576.559	1741.856

Source: Created by the authors.

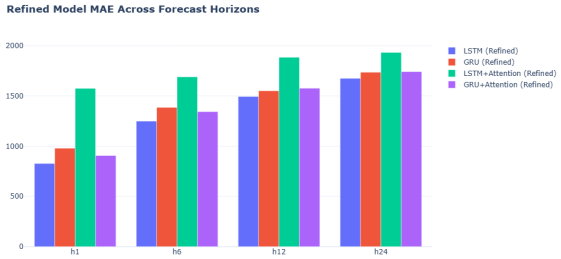


Figure 7: Comparison of MAE across forecast horizons for original and refined multivariate models.

Source: Created by the authors.

Table 5: Quantified Performance Improvements for h=1

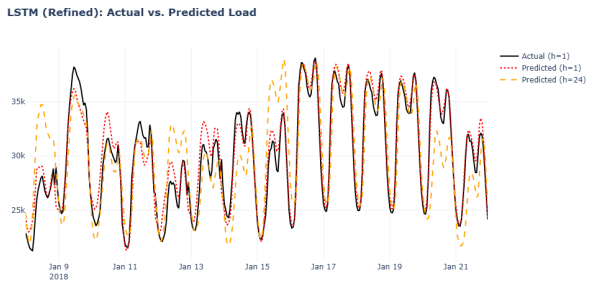
Metric	Orig. GRU	Ref. LSTM	Abs. Gain	% Improve	Type
MAE	1203.837	826.900	376.937	31.31	↓
RMSE	1647.313	1189.787	457.526	27.77	↓
MAPE	4.011%	2.842%	1.169 p.p.	29.14	↓
MASE	1.128	0.775	0.353	31.30	↓
R^2	0.872	0.933	0.061	7.00	↑

Source: Created by the authors.

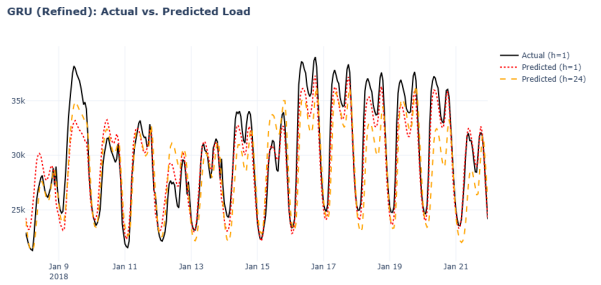
Table 4 indicates substantial MAE reductions across horizons relative to Table 2. The refined LSTM emerges as best overall, achieving the lowest MAE at all horizons, with GRU+Attention (Ref.) marginally competitive at h=6 and h=12. Figure 7 visually confirms these trends, illustrating the substantial MAE reduction achieved by the refined models across all horizons.

The quantified improvements for the best refined model (LSTM) over the best original model (GRU) for the 1-hour horizon (h=1) further highlight the success of our refinement process (Table 5). Absolute and percentage gains are reported for principal accuracy metrics. The refined LSTM reduces MAE by 376.937 MW (31.31%), RMSE by 457.526 MW (27.77%), and MAPE by 1.169 percentage points (29.14%), while improving MASE and R^2 by 31.30% and 7.00%, respectively.

Refined models, as shown in Figure 8, track peaks and troughs more tightly and exhibit a much



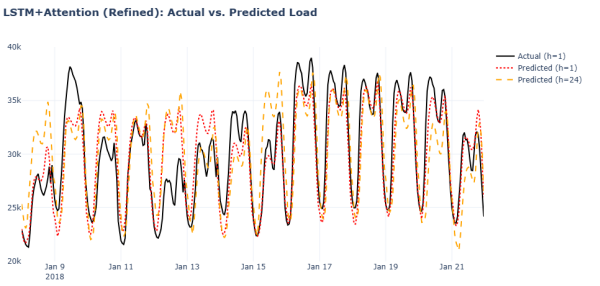
(a) Refined LSTM



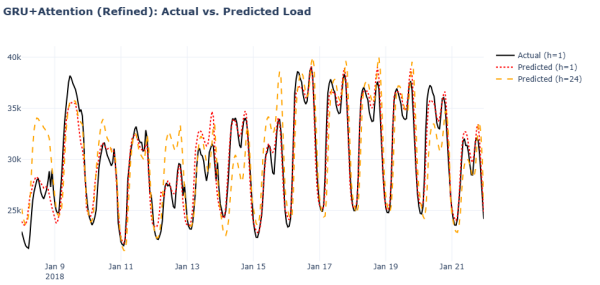
(b) Refined GRU

Figure 8: Refined recurrent model predictions vs. actual load (sample week): **(a)** LSTM and **(b)** GRU after feature reduction.

Source: Created by the authors.



(a) Refined LSTM+Attention



(b) Refined GRU+Attention

Figure 9: Refined attention-augmented model predictions vs. actual load (sample week): **(a)** LSTM+Attention and **(b)** GRU+Attention after feature reduction.

Source: Created by the authors.

closer visual fit to the actual load. Panels (a) and (b) correspond to the refined LSTM and refined GRU, respectively, demonstrating better capture of the data's dynamic behavior and reducing systematic underestimation, consistent with the quantitative improvements.

The attention-augmented refined variants are depicted in Figure 9(a) (LSTM+Attention) and Figure 9(b) (GRU+Attention), showing modest competitiveness for *GRU+Attention* at intermediate horizons.

4.5 Uncertainty Quantification Results

The quality of uncertainty estimates is crucial for risk management. We evaluate interval quality using PICP (Table 6) and MPIW (Table 7) across all horizons. The results show a clear trade-off between coverage (PICP) and sharpness (MPIW), consistent with the calibration–sharpness paradigm of strictly proper scoring rules [25]. MC Dropout models (both GRU and LSTM) consistently achieved higher PICP values across all horizons compared to Deep Ensembles. For instance, GRU–MC reached a PICP of 0.927 at h=1, close to the ideal 95% nominal coverage. Conversely, Deep Ensemble models yielded much narrower MPIW, but this came at the cost of under-coverage and poorer calibration.

Table 6: Prediction Interval Coverage Probability (PICP) across horizons (target 0.95; higher is better)

Model	h1	h6	h12	h24
GRU–Ens	0.629	0.499	0.442	0.362
GRU–MC	0.927	0.768	0.638	0.553
GRU+A–Ens	0.590	0.566	0.488	0.363
GRU+A–MC	0.830	0.695	0.588	0.493
LSTM–Ens	0.554	0.479	0.410	0.373
LSTM–MC	0.914	0.764	0.680	0.530
LSTM+A–Ens	0.606	0.555	0.467	0.399
LSTM+A–MC	0.856	0.644	0.608	0.544

Source: Created by the authors.

Figure 10 summarizes PICP vs. MPIW behavior across horizons. The coverage–sharpness scatter in Figure 11 summarizes this trade-off: MC Dropout models cluster towards the top-right (higher PICP, wider MPIW), while Deep Ensembles group towards the bottom-left (lower PICP, narrower MPIW). Furthermore, calibration curves (not shown) indicate better calibration for MC Dropout (curves closer to $y=x$) than for Deep Ensembles.

Figure 10: UQ metric comparison across horizons: PICP (coverage) vs. MPIW (interval width) for Deep Ensemble and MC Dropout variants (LSTM / GRU). MC Dropout attains higher coverage with moderate width expansion.

Source: Created by the authors.

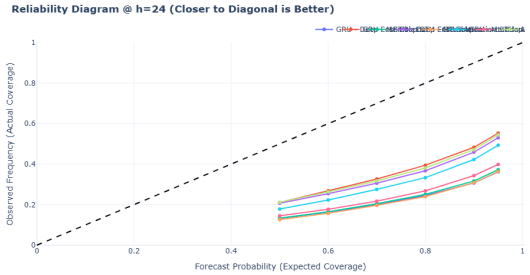


Figure 11: Coverage–sharpness trade-off scatter: MC Dropout models cluster nearer target coverage (95%) with acceptable width; Deep Ensembles produce sharper but under-covering intervals.

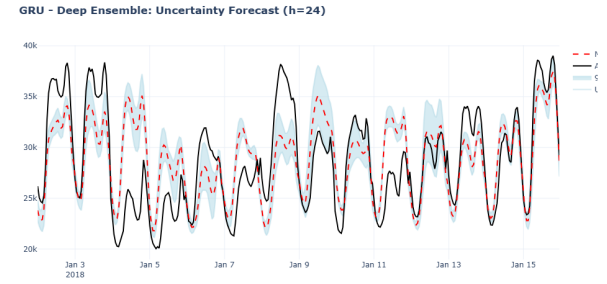
Source: Created by the authors.

Table 7: Mean Prediction Interval Width (MPIW) across horizons (lower is better; units: MW)

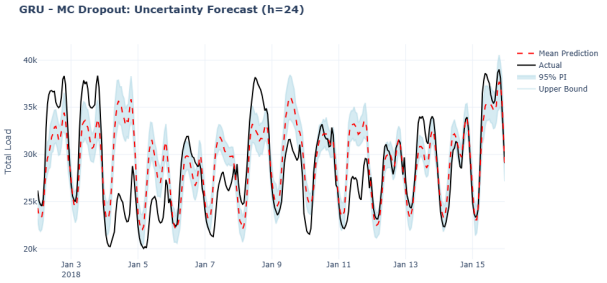
Model	h1	h6	h12	h24
GRU–Ens	1198.810	1579.293	1731.931	1913.764
GRU–MC	2992.200	3063.555	2978.452	2914.356
GRU+A–Ens	1279.209	1956.668	2034.294	1982.618
GRU+A–MC	2969.720	3072.988	3071.915	2913.472
LSTM–Ens	1035.183	1491.849	1670.195	1983.171
LSTM–MC	2998.307	3117.371	3132.415	2917.098
LSTM+A–Ens	1414.458	1882.396	1949.839	2173.803
LSTM+A–MC	2937.407	2982.504	2965.459	2879.422

Source: Created by the authors.

The tabulated results (Table 6 and Table 7) quantify the trade-off: Deep Ensembles attain markedly narrower mean interval widths (higher sharpness) but at the expense of under-coverage relative to the 0.95 target. MC Dropout intervals are broader yet substantially closer to nominal coverage, yielding superior reliability for risk-sensitive operations. Representative 95% interval examples at h=24 are shown in Figure 12, Figure 13, Figure 14, and Figure 15, confirming that MC Dropout captures peak volatility more reliably. *Deep Ensemble examples* (e.g., Figure 12a and



(a) GRU DeepEns



(b) GRU MC Dropout

Figure 12: GRU models: 95% uncertainty forecast (h=24): **(a)** Deep Ensemble; **(b)** MC Dropout.

Source: Created by the authors.

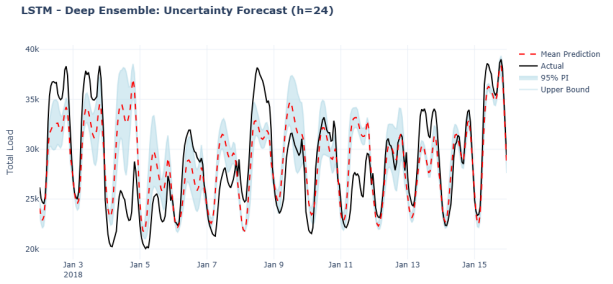
Figure 13a) often produce narrower intervals that miss excursions, indicating overconfidence, whereas *MC Dropout examples* (e.g., Figure 12b and Figure 13b) show wider prediction intervals that more frequently encapsulate the actual load, especially during high-volatility periods. Additional attention-augmented uncertainty examples are displayed in Figure 14, where panel (a) shows the GRU+Attention Deep Ensemble and panel (b) the corresponding MC Dropout configuration, further illustrating how attention mechanisms influence predictive variance.

Operational implication. While Deep Ensembles produce sharper (narrower) intervals, their systematic under-coverage represents an operational risk for grid management (e.g., reserve shortfalls when demand is underestimated). In contrast, MC Dropout's broader intervals yield more conservative and operationally safer forecasts, making MC Dropout the preferred choice for deployment in mission-critical settings.

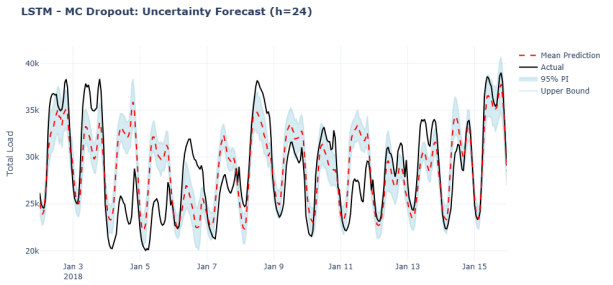
4.6 Discussion

Key observations:

- **Feature dominance:** `total_load_forecast` acts as a high-fidelity exogenous proxy, dwarfing



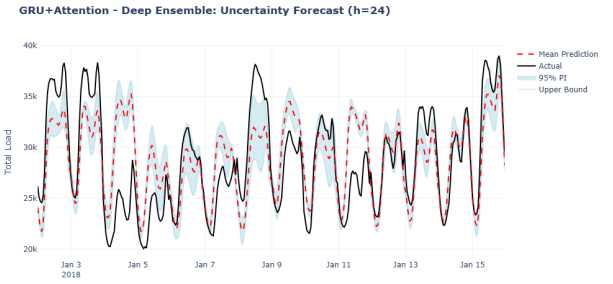
(a) LSTM DeepEns



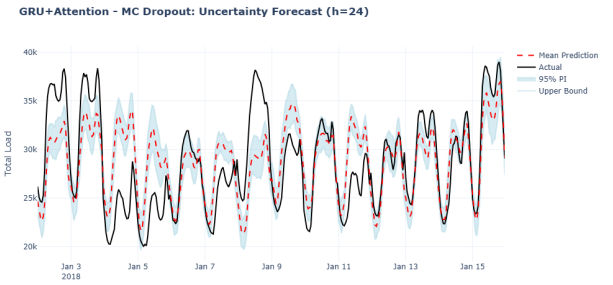
(b) LSTM MC Dropout

Figure 13: LSTM models: 95% uncertainty forecast (h=24): **(a)** Deep Ensemble; **(b)** MC Dropout.

Source: Created by the authors.



(a) GRU+Attn DeepEns



(b) GRU+Attn MC Dropout

Figure 14: GRU+Attention models: 95% uncertainty forecast (h=24): **(a)** Deep Ensemble; **(b)** MC Dropout.

Source: Created by the authors.

other predictors.

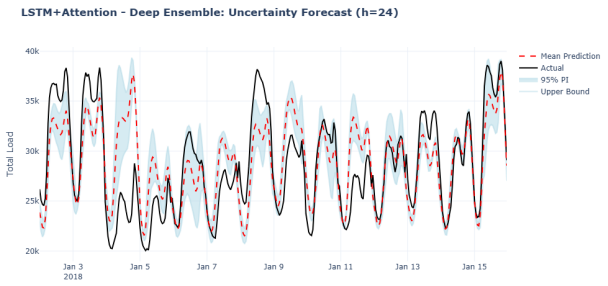
- **Model selection:** GRU favored in unreduced space; LSTM excels after denoising, suggesting LSTM sensitivity to redundant inputs.
- **Attention utility:** Attention layers did not add value without deeper tuning (context length, head configuration, regularization); strong engineered lag features may already encode critical temporal dependencies, reducing the marginal benefit of attention under current settings.
- **Parsimony gains:** Removing low/negative-contribution variables reduced variance and improved generalization (lower MAE, better calibration trends), highlighting the joint benefit of interpretable feature selection and accuracy.
- **UQ approach:** MC Dropout delivered better coverage; ensembles offered sharper but miscalibrated intervals—a trade-off with direct operational implications.

Overall, systematic feature selection coupled with modest architectural complexity delivered the best accuracy–calibration balance.

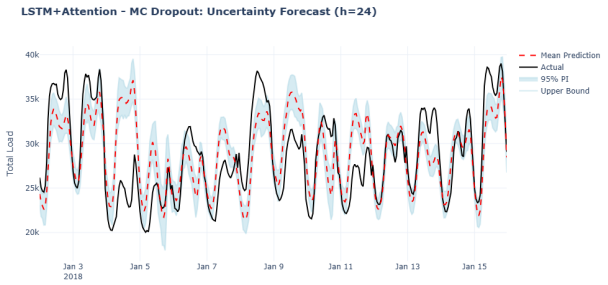
5 Conclusion

This research developed and rigorously evaluated deep learning models for multivariate short-term electricity load forecasting, achieving high accuracy through systematic data preparation, informed feature engineering, and hyperparameter optimization. After refinement to an 8-feature subset guided by permutation importance, the LSTM model emerged as the top point forecaster, reducing $h1$ MAE by over 31% relative to the best initial (unrefined) GRU and attaining a MASE of 0.775 (demonstrating improvement over a naive seasonal baseline). These gains underscore the value of targeted feature parsimony in mitigating overfitting while preserving essential seasonal and exogenous drivers.

Comprehensive Uncertainty Quantification (UQ) analysis revealed that MC Dropout consistently produced better-calibrated and higher-coverage prediction intervals than Deep Ensembles. While ensembles yielded sharper (narrower) intervals, they systematically under-covered true outcomes, indicating overconfidence. MC Dropout offered a more favorable balance for operational risk contexts where dependable interval coverage is paramount.



(a) LSTM+Attn DeepEns



(b) LSTM+Attn MC Dropout

Figure 15: LSTM+Attention models: 95% uncertainty forecast (h=24): **(a)** Deep Ensemble; **(b)** MC Dropout.

Source: Created by the authors.

Overall, the findings reaffirm the effectiveness of deep recurrent architectures, particularly when combined with principled feature selection, automated hyperparameter tuning, and robust UQ. Integrating uncertainty estimates alongside point forecasts meaningfully enhances decision support for grid operators and market participants facing variability and structural complexity in modern power systems.

In summary, the proposed integrated framework unifies feature refinement, deep architectures, and uncertainty quantification into a single operational pipeline. The refined models deliver consistent accuracy gains across short-term horizons while preserving interpretability, and the UQ analysis indicates that MC Dropout achieves superior calibration and coverage relative to Deep Ensembles—an advantage for risk-aware grid operations and deployment.

Future Work

Future extensions and deployment-oriented improvements include:

- **Broader Model Benchmarking:** A more comprehensive comparative analysis against advanced statistical benchmarks like Seasonal ARIMA (SARIMA) and state-of-the-art machine learning models (e.g., LightGBM, Prophet) would provide a richer context for the deep learning models' performance.
- **Deeper Attention Analysis:** A more thorough investigation into the attention mechanism is warranted. This includes a systematic hyperparameter search for attention-augmented models and an analysis of the learned attention weights to gain insights into model interpretability and determine their specific benefits for electricity load forecasting.
- **Impact of Feature Selection on UQ:** Evaluate how the refined feature set impacts the quality and calibration of the uncertainty intervals produced by both Deep Ensembles and MC Dropout, and whether the removal of noisy features also tightens the bounds.
- **Disentangling Uncertainty:** Explore advanced Uncertainty Quantification methods, such as variational Bayesian neural networks, to explicitly separate and quantify aleatoric (inherent data noise) from epistemic (model-related) uncertainty. This would provide a more nuanced understanding of forecast reliability.
- **External Weather Data Integration:** Incorporate higher-resolution meteorological features (temperature, humidity, solar radiation, wind speed) to better capture non-periodic demand fluctuations.
- **Advanced Temporal Encoding:** Employ cyclical (sine/cosine) encodings and interaction features (e.g., temperature $\times$ hour) to enrich temporal semantics.
- **Architectural Exploration:** Incorporate patch-token Transformer backbones tailored for long-horizon forecasting—specifically PatchTST and a BERT-augmented variant—and evaluate them within our calibrated-UQ pipeline (MC Dropout / Deep Ensembles), with emphasis on longer horizons and parallelism benefits [26], [27].
- **Enhanced UQ Methods:** Investigate refined dropout rate tuning, and direct quantile regression (e.g., QRNN) for sharper yet reliable intervals.
- **Extended Horizons:** Expand forecasting windows (48 h, 7 day) to support medium-term planning; may necessitate hierarchical or multi-scale modeling.

- **Anomaly-Aware Strategies:** Introduce holiday/event indicators, robust loss functions, and anomaly-aware preprocessing to better handle extreme deviations.
- **Operational Deployment:** Build a real-time inference and monitoring pipeline for drift detection, automatic retraining triggers, and continuous calibration tracking.

These directions aim to further tighten accuracy–calibration trade-offs and enhance practical resilience of forecasting solutions in evolving energy system contexts.

References:

[1] T. Hong, S. Fan, Probabilistic electric load forecasting: A tutorial review, *Int. J. Forecasting*, 32(3), 714–728, 2016.

[2] M. H. Al-Sharif, J. F. Al-Amri, A comprehensive review of electric load forecasting models, *Applied Energy*, 278, 115712, 2020.

[3] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, G. M. Ljung, *Time Series Analysis: Forecasting and Control*, Wiley, 2019.

[4] J. W. Taylor, R. Buizza, A comparison of univariate and multivariate models for electricity demand forecasting, *J. R. Stat. Soc. C*, 52(2), 209–223, 2003.

[5] S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural Comput.*, 9(8), 1735–1780, 1997.

[6] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, arXiv:1406.1078, 2014.

[7] Y. Bengio, P. Simard, P. Frasconi, Learning long-term dependencies with gradient descent is difficult, *IEEE Trans. Neural Netw.*, 5(2), 157–166, 1994.

[8] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, arXiv:1409.0473, 2014.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: *Adv. Neural Inf. Process. Syst.* 30, 2017.

[10] Q. Wang, A. Tuohy, M. Ortega-Vazquez, M. Bello, E. Ela, D. Kirk-Davidoff, W. B. Hobbs, D. J. Ault, R. Philbrick, Quantifying the value of probabilistic forecasting for power system operation planning, Appl. Energy, 343, 121254, 2023.

[11] Y. Gal, Z. Ghahramani, Dropout as a Bayesian approximation: Representing model uncertainty in deep learning, in: Proc. ICML, 1050–1059, 2016.

[12] B. Lakshminarayanan, A. Pritzel, C. Blundell, Simple and scalable predictive uncertainty estimation using deep ensembles, in: *Adv. Neural Inf. Process. Syst.* 30, 2017.

[13] A. Fisher, C. Rudin, F. Dominici, All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously, *J. Mach. Learn. Res.*, 20(177), 1–81, 2019.

[14] N. J. Jhana, "Energy consumption, generation, prices, and weather," *Kaggle*, 2019. [Online]. Available: `https://www.kaggle.com/datasets/nicholasjhana/energy-consumption-generation-prices-and-weather` [Accessed: 31 Jul. 2025].

[15] F. Guo, H. Mo, J. Wu, L. Pan, H. Zhou, Z. Zhang, L. Li, F. Huang, A Hybrid Stacking Model for Enhanced Short-Term Load Forecasting, *Electronics*, 13(14), Art. 2719, 2024. doi:10.3390/electronics13142719.

[16] A. Paszke et al., PyTorch: An imperative style, high-performance deep learning library, in: *Adv. Neural Inf. Process. Syst.* 32, 8024–8035, 2019.

[17] W. McKinney, Data structures for statistical computing in Python, in: Proc. 9th Python Sci. Conf., 51–56, 2010.

[18] F. Pedregosa et al., Scikit-learn: Machine learning in Python, *J. Mach. Learn. Res.*, 12, 2825–2830, 2011.

[19] J. D. Hunter, Matplotlib: A 2D graphics environment, *Comput. Sci. Eng.*, 9(3), 90–95, 2007.

[20] M. L. Waskom, Seaborn: Statistical data visualization, *J. Open Source Softw.*, 6(60), 3021, 2021.

[21] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: A next-generation hyperparameter optimization framework, in: Proc. KDD, 2623–2633, 2019.

[22] L. Breiman, Random forests, *Mach. Learn.*, 45(1), 5–32, 2001.

[23] R. J. Hyndman, A. B. Koehler, Another look at measures of forecast accuracy, *Int. J. Forecasting*, 22(4), 679–688, 2006. doi:10.1016/j.ijforecast.2006.03.001.

[24] S. Bai, J. Z. Kolter, V. Koltun, An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, arXiv:1803.01271, 2018.

[25] T. Gneiting, A. E. Raftery, Strictly proper scoring rules, prediction, and estimation, *J. Amer. Stat. Assoc.*, 102(477), 359–378, 2007.

[26] Y. Nie, N. H. Nguyen, P. Sinthong, J. Kalagnanam, A Time Series is Worth 64 Words: Long-term Forecasting with Transformers, in: Proc. ICLR, 2023. arXiv:2211.14730.

[27] H. Ahmad, S. K. Mortazavi, M. El Bahnasawi, F. Al Machot, W. V. Kambale, K. Kyamakya, Enhanced Time Series Forecasting: Integrating PatchTST with BERT Layers, in: Proc. IEEE ICAMCS, 2024. doi:10.1109/ICAMCS62774.2024.00014.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.