Machine Learning Demand Forecasting Integrating Weather, Usage Patterns, and Pricing to Reduce Utility Generation Waste

Jiaxin Jiang[1*], Shubham Rout[2]

[1]Department of Engineering Science, University of Oxford, Parks Road, Oxford, OX1 3PJ, United Kingdom

[2]Department of Engineering Science, University of Oxford, Osney Mead, Botley, Oxford, OX2 0ES, United Kingdom

---

Abstract

Accurate short-term electricity demand forecasting is critical for minimizing generation waste and reducing carbon emissions. Existing approaches typically exploit a single exogenous signal: weather, historical usage, or price, leaving cross-domain interactions under-explored. We propose the Unified Multi-Source Ensemble (UMSE) framework, which fuses gradient-boosted trees (XGBoost) and long short-term memory (LSTM) networks through a learned attention mechanism, operating on an explicitly constructed cross-feature interaction space spanning weather, usage patterns, and time-of-use pricing domains. Evaluated on a synthetic dataset designed to reproduce the statistical properties of a mid-sized North American utility service area (hourly resolution, two years), all models incorporating usage features achieve mean absolute percentage error (MAPE) below 1% on the autumn/winter test period. However, an honest evaluation reveals that XGBoost with usage features alone (MAPE = 0.379%) outperforms UMSE (MAPE = 0.381 ± 0.004%), while the full-feature XGBoost (MAPE = 0.369%) also significantly outperforms UMSE (Diebold–Mariano $p < 0.001$). The attention mechanism assigns 96.3% average weight to the XGBoost branch, indicating that the LSTM contributes minimally. Ablation analysis confirms that all three cross-feature interaction families contribute positively, with pricing-usage interactions providing the largest marginal gain. Both XGBoost and UMSE reduce cumulative over-generation waste by approximately 77% relative to the persistence baseline. Our results demonstrate that multi-source feature integration yields substantial forecasting gains, but

---

*Corresponding author

February 17, 2026

Email addresses: jiaxin.jiang@eng.ox.ac.uk (Jiaxin Jiang), shubham.rout@eng.ox.ac.uk (Shubham Rout)

attention-based ensemble fusion does not improve over well-engineered gradient boosting on this synthetic benchmark—a finding with implications for the complexity–accuracy trade-off in operational forecasting systems.

Keywords: electricity demand forecasting, machine learning, XGBoost, LSTM, attention mechanism, feature interactions, generation waste reduction, smart grid

---

1. Introduction

Electric utilities routinely over-dispatch generation capacity to hedge against demand uncertainty. The resulting over-generation waste—electricity produced but not consumed—carries direct fuel costs and avoidable carbon emissions (Hobbs et al., 1999). As grids integrate higher shares of variable renewable generation, the economic and environmental penalties of inaccurate demand forecasts compound: over-generation during periods of high solar irradiance can trigger curtailment, while under-generation forces expensive peaking units online (Denholm et al., 2015; Hart et al., 2012).

Electricity demand is shaped by at least three exogenous domains that interact in non-trivial ways. Weather drives space-conditioning loads through heating and cooling degree-hour mechanisms (Valor et al., 2001; Apadula et al., 2012; Hatton et al., 2016). Historical usage patterns encode autoregressive dynamics, occupancy cycles, and seasonal baselines (Wang et al., 2016). Time-of-use pricing structures modulate consumption timing through demand-response incentives (Faruqui and Sergici, 2010; [illegible]). Crucially, these domains do not act independently: an extreme-heat event concurrent with a peak-pricing window produces demand behaviour that cannot be predicted by weather or price signals alone.

Despite three decades of load forecasting research (Hippert et al., 2001; Hong and Fan, 2016), most methods treat exogenous features as additive inputs rather than explicitly modelling their interactions. Classical approaches such as SARIMA and exponential smoothing (Fan and Hyndman, 2012; Hong, 2010) capture temporal autocorrelation but cannot represent non-linear cross-domain effects. Recent deep learning models (Kong et al., 2019; Sharma et al., 2016; Shi et al., 2018) learn feature interactions implicitly through hidden layers, but the learned representations are opaque and the contribution of each interaction family remains unquantifiable. Ensemble methods that combine statistical and neural branches (Wang et al., 2017; Li et al., 2019) typically merge predictions via simple averaging or fixed weights, forfeiting the opportunity for context-dependent fusion.

Gradient-boosted trees, and XGBoost in particular (Chen and Guestrin, 2016; Tsaousi et al. 2019), naturally discover feature interactions through recursive partitioning. However, tree-based models are inherently local experts: they cannot capture long-range temporal dependencies that LSTM networks handle well (Hochreiter and Schmidhuber, 1997). Conversely, LSTMs struggle with tabular features that have no sequential structure. Recent hybrid work has shown that combining complementary model families can outperform either in isolation (Nixon et al. 2020; Wang et al. 2017), but the fusion mechanism is typically a static blend rather than a learned, input-dependent weighting.

This paper makes the following contributions:

1. We introduce the Unified Multi-Source Ensemble (UMSE) framework, which fuses XGBoost and LSTM branches via a lightweight attention mechanism that learns context-dependent branch weights, allowing the ensemble to shift reliance between branches as operating conditions change.
2. We construct an explicit cross-feature interaction space spanning weather, usage-pattern, and pricing domains, enabling transparent ablation of each interaction family and quantification of its marginal contribution.
3. We perform an honest comparative evaluation showing that well-engineered gradient boosting outperforms the proposed ensemble framework, providing evidence that model complexity does not guarantee accuracy gains when strong autoregressive features are available.
4. We quantify the downstream impact of forecast accuracy on generation waste reduction, demonstrating that both UMSE and standalone XGBoost reduce cumulative over-generation waste by approximately 77% relative to a persistence baseline, with no statistically significant difference between the two.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 presents the UMSE framework. Section 4 describes the synthetic dataset and evaluation protocol. Section 5 reports experimental results. Section 6 discusses findings, generation waste implications, and limitations. Section 7 concludes.

2. Related Work

2.1. Statistical and Traditional Machine Learning Approaches

Short-term load forecasting has been studied for over five decades. Classical approaches rely on autoregressive integrated moving average (ARIMA) models and their seasonal variants (Hong, 2010; Hyndman et al., 2008), semi-parametric additive models (Fan and Hyndman, 2012), and multiple linear regression with weather covariates (Taylor et al., 2006; Apadula et al., 2012). While these methods are interpretable and well-understood, they assume stationary relationships between predictors and demand, a condition often violated during extreme weather events or structural changes in consumption patterns.

The advent of gradient boosting methods brought substantial accuracy gains. Touzani et al. (2018) demonstrated that gradient-boosted trees outperform linear baselines for commercial building energy consumption. Chen and Guestrin (2016) introduced XGBoost, which has since become a competitive baseline across energy forecasting benchmarks due to its built-in regularization and ability to discover feature interactions through recursive splits.

2.2. Deep Learning and Multi-Source Feature Integration

Recurrent neural networks, particularly LSTM architectures (Hochreiter and Schmidhuber, 1997), have been widely adopted for load forecasting owing to their ability to capture long-range temporal dependencies. Marino et al. (2016) showed that LSTM networks outperform feed-forward architectures for building-level load prediction. Shi et al. (2018) proposed a pooling strategy for household-level forecasting, while Kong et al. (2019) incorporated resident behaviour features. Cai et al. (2019) compared deep learning to traditional time-series methods on building-level data, finding that deep models excel when sufficient training data are available.

A parallel research direction focuses on integrating heterogeneous data sources. Haben et al. (2019) examined the effect of temperature granularity on low voltage load forecasts. Wang et al. (2016) explored accuracy-weighted features in a big data framework. [illegible] (2016) and [illegible] (2020) studied how pricing signals reshape demand profiles, though their work focused on demand response rather than forecasting. Despite this rich literature, few studies explicitly construct and ablate cross-domain interaction features to quantify the marginal value of each interaction family.

2.3. Ensemble Methods and Attention-Based Fusion

Ensemble learning combines multiple base learners to reduce variance and improve robustness. Moon et al. (2020) employed a stacking approach for short-term load forecasting, combining statistical and neural base models. Wang et al. (2017) used ensemble techniques for probabilistic wind power forecasting. Attention mechanisms, originally developed for sequence-to-sequence translation [Vaswani et al., 2017], have been adapted for time-series forecasting by Li et al. (2019), who enhanced the locality of Transformer architectures. Kong et al. (2017) proposed an LSTM-based architecture for smart-grid load forecasting but used fixed combination weights.

More recently, purpose-built deep architectures have advanced the state of the art in time-series forecasting. Oreshkin et al. (2020) proposed N-BEATS, a purely feed-forward architecture using basis expansion that achieves strong performance without recurrence. Lim et al. (2021) introduced the Temporal Fusion Transformer (TFT), which combines multi-horizon attention with variable-selection networks. Nie et al. (2023) demonstrated that patching strategies (PatchTST) enable vanilla Transformers to match or exceed specialized architectures on long-horizon benchmarks. We include N-BEATS and a Transformer encoder baseline in our experiments to position the proposed framework against these modern architectures.

Our work differs from prior ensemble approaches in three respects: (i) we combine structurally different model families (tree-based and recurrent) rather than variants of the same architecture; (ii) the fusion weights are learned via attention and vary with input context; and (iii) we provide a full ablation of explicitly constructed cross-feature interactions, enabling transparent attribution of forecast improvements to specific domain interactions.

3. Methodology

3.1. Problem Formulation

We consider one-step-ahead hourly demand forecasting ($h = 1$). Let y_t denote the aggregate electrical demand (MW) at hour t. The forecasting task is to estimate y_{t+1} given an information set $\mathcal{I}_t$ comprising historical demand, weather observations, usage pattern statistics, and pricing signals up to and including hour t:

$$\hat{y}_{t+1} = f(\mathbf{X}_t), \qquad \mathbf{X}_t = [\mathbf{x}_t^{(\text{temp})}, \mathbf{x}_t^{(\text{wthr})}, \mathbf{x}_t^{(\text{use})}, \ldots]^{\top}_{t,t'} \tag{1}$$

The forecast horizon is restricted to $h = 1$ (one hour ahead) throughout this study; multi-horizon extensions are deferred to future work.

3.2. Feature Engineering

The feature engineering pipeline (Fig. 2) transforms raw data streams into a structured feature vector through two stages: base feature construction and cross-feature interaction construction.

3.2.1. Base Feature Construction

Temporal features. Cyclical encodings of hour-of-day (h), day-of-week (d), and month-of-year (m) are constructed as sine–cosine pairs to preserve periodicity:

$$\mathbf{x}_t^{(h)} = \left[\sin\frac{2\pi h}{24}, \cos\frac{2\pi h}{24}\right], \quad \text{and analogously for } d \text{ and } m. \tag{2}$$

A binary weekend indicator completes the temporal feature set.

Weather features. Raw temperature (T_t, °C), relative humidity (RH_t, %), solar irradiance (G_t, W m^{-2}), and wind speed (v_t, m/s) are supplemented with heating and cooling degree-hours relative to comfort baselines $T_{\text{heat}} = 18$°C and $T_{\text{cool}} = 24$°C:

$$\text{HDH}_t = \max(T_{\text{heat}} - T_t, 0), \qquad \text{CDH}_t = \max(T_t - T_{\text{cool}}, 0). \tag{3}$$

The six weather features (T_t, RH_t, G_t, v_t, HDH_t, CDH_t) are included at the current time step only; no lagged weather values are used.

Usage pattern features. Historical demand is represented through lag features ($y_{t-1}, y_{t-24}, y_{t-168}$), rolling window statistics, and exponential moving averages:

$$\bar{y}_t^{(w)} = \frac{1}{w}\sum_{i=0}^{w-1} y_{t-i}, \qquad \sigma_t^{(w)} = \sqrt{\frac{1}{w-1}\sum_{i=0}^{w-1}(y_{t-i} - \bar{y}_t^{(w)})^2}, \tag{4}$$

where $w \in \{6, 12, 24, 168\}$ hours. These features capture short-term autoregressive dynamics, diurnal cycles, and weekly periodicity.

Figure 1: Feature engineering pipeline showing the construction of cross-feature interaction terms. Base features from four domains (left) are combined pairwise to produce interaction features (middle), which are concatenated with the base features to form the full feature vector (right).

Pricing features. Time-of-use (TOU) pricing is encoded as a three-tier categorical variable (off-peak, shoulder, peak) with corresponding rate values. The current price, price change from the previous hour, and a rolling 24-hour mean price constitute the pricing feature set.

2.2.2. Cross-Feature Interaction Construction

While tree-based models can discover interactions implicitly through recursive splits, explicitly constructing interaction terms provides two advantages: (i) LSTM inputs benefit from pre-computed interactions that would otherwise require multiple hidden layers to learn, and (ii) explicit interactions enable transparent ablation.

Three families of cross-feature interactions are constructed:

Weather × Usage interactions. These capture phenomena such as weather-driven deviations from typical consumption patterns:

$$\mathbf{x}_t^{(W\times U)} = [T_t \cdot y_t^{lag}, \; \mathrm{CDD}_t \cdot y_t^{peak}, \; H_t \cdot L_t, \; \mathrm{HDD}_t \cdot y_t^{lag}] \quad (5)$$

where $\zeta_t^{(peak)} = y_t / \max_{t-k \le \tau < t} y_\tau$ is the peak ratio, $\lambda_t = y_t / \bar{y}_t^{(k)}$ is the load factor, and $\sigma_t^{(k)}$ is the 24-hour rolling standard deviation.

Pricing–Usage interactions. These model the demand-response effect whereby consumers shift loads in response to price signals:

$$\phi_t^{(PU)} = [p_t \cdot \bar{y}_t^{(k)},\ \Delta p_t \cdot \zeta_t^{(k)},\ \mathbb{1}_{peak} \cdot \zeta_t^{(peak)}] \tag{6}$$

where $\zeta_t^{(k)} = \sigma_t^{(k)} / \bar{y}_t^{(k)}$ is the coefficient of variation (volatility) and $\mathbb{1}_{peak}$ is the peak-hour indicator.

Weather–Pricing interactions. These capture joint effects such as peak-price periods coinciding with extreme temperatures:

$$\phi_t^{(WP)} = [T_t \cdot p_t,\ \mathrm{CDD}_t \cdot \mathbb{1}_{peak},\ T_t \cdot p_t] \tag{7}$$

The full feature vector concatenates base and interaction features:

$$x_t = [\phi_t^{(time)} \,\|\, \phi_t^{(weather)} \,\|\, \phi_t^{(lag)} \,\|\, \phi_t^{(roll)} \,\|\, \phi_t^{(WU)} \,\|\, \phi_t^{(PU)} \,\|\, \phi_t^{(WP)}] \in \mathbb{R}^d \tag{8}$$

3.3. Ensemble Architecture

The CHMR framework (Fig. 2) comprises two complementary branches and a learned fusion mechanism.

3.3.1. Gradient Boosting Branch

The XGBoost branch (Chen and Guestrin, 2016) operates on the full tabular feature vector x_t. The model produces a point prediction through an additive ensemble of K regression trees:

$$\hat{y}_{t+1}^{(XGB)} = \sum_{k=1}^{K} f_k(x_t), \quad f_k \in \mathcal{F} \tag{9}$$

where $\mathcal{F}$ is the space of regression trees and each f_k is trained to minimize the regularized squared-error objective. XGBoost excels at discovering non-linear feature interactions through recursive partitioning and provides built-in feature importance scores via the gain metric.

Figure 2: Architecture of the proposed Unified Multi-Source Ensemble (UMSE) framework. Four input data streams are processed through a feature engineering layer that constructs cross-feature interactions, then fed to parallel XGBoost and LSTM branches whose predictions are combined via an attention-weighted fusion mechanism.

3.3.2. LSTM Branch

The LSTM branch (Hochreiter and Schmidhuber, 1997) processes a sequence of the most recent $L = 48$ feature vectors $\{\mathbf{x}_{t-L+1}, \ldots, \mathbf{x}_t\}$, spanning two full diurnal cycles. The standard LSTM cell at step τ computes:

$$\mathbf{f}_\tau = \sigma(\mathbf{W}_f \mathbf{x}_\tau + \mathbf{U}_f \mathbf{h}_{\tau-1} + \mathbf{b}_f) \quad (10)$$

$$\mathbf{i}_\tau = \sigma(\mathbf{W}_i \mathbf{x}_\tau + \mathbf{U}_i \mathbf{h}_{\tau-1} + \mathbf{b}_i) \quad (11)$$

$$\tilde{\mathbf{c}}_\tau = \tanh(\mathbf{W}_c \mathbf{x}_\tau + \mathbf{U}_c \mathbf{h}_{\tau-1} + \mathbf{b}_c) \quad (12)$$

$$\mathbf{c}_\tau = \mathbf{f}_\tau \odot \mathbf{c}_{\tau-1} + \mathbf{i}_\tau \odot \tilde{\mathbf{c}}_\tau \quad (13)$$

$$\mathbf{o}_\tau = \sigma(\mathbf{W}_o \mathbf{x}_\tau + \mathbf{U}_o \mathbf{h}_{\tau-1} + \mathbf{b}_o) \quad (14)$$

$$\mathbf{h}_\tau = \mathbf{o}_\tau \odot \tanh(\mathbf{c}_\tau) \quad (15)$$

where $\sigma(\cdot)$ is the logistic sigmoid, $\odot$ denotes element-wise multiplication, and $\mathbf{W}_*, \mathbf{U}_*, \mathbf{b}_*$ are learnable parameters. A fully connected layer maps the final hidden state $\mathbf{h}_t$ to a scalar prediction $\hat{y}_{t+1}^{\mathrm{LSTM}}$.

Dropout (Srivastava et al., 2014) is applied to the LSTM input and recurrent connections with probability $p = 0.2$ to prevent overfitting.

3.3.3. Attention-Weighted Fusion

Rather than combining branch predictions with fixed weights, we employ a lightweight attention mechanism that learns context-dependent fusion weights. The attention module takes as input a context vector $\mathbf{c}_t$ comprising temporal features (cyclical hour and day-of-week encodings), temperature, humidity, TOU rate, and a peak-hour indicator. Note that the branch predictions $\hat{y}^{\text{XGB}}, \hat{y}^{\text{LSTM}}$ are not included in the context; this design prevents the attention module from trivially learning to select the lower-error branch and instead forces it to base its weighting on operating conditions. The attention weights are computed by a two-layer feed-forward network with ReLU activation:

$$\boldsymbol{\alpha}_t = \text{softmax}(\mathbf{W}_2 \, \text{ReLU}(\mathbf{W}_1 \mathbf{c}_t + \mathbf{b}_1) + \mathbf{b}_2) \in \mathbb{R}^2, \tag{16}$$

where $\mathbf{W}_1 \in \mathbb{R}^{h \times d}$, $\mathbf{W}_2 \in \mathbb{R}^{2 \times h}$, and $h = 32$ is the hidden dimension. The ensemble prediction is then:

$$\hat{y}^{\text{Fusion}}_{t+1} = \alpha^{\text{XGB}}_t \hat{y}^{\text{XGB}}_{t+1} + \alpha^{\text{LSTM}}_t \hat{y}^{\text{LSTM}}_{t+1}. \tag{17}$$

3.4. Training Procedure

The three-stage training procedure prevents information leakage between components. First, the XGBoost branch is trained on the training set with hyperparameters selected via grid search (Section 4.3). Second, the LSTM branch is trained on the training set with early stopping based on the first half of the validation set. Third, the attention fusion module is trained on the second half of the validation set using the frozen XGBoost and LSTM predictions as inputs, thereby ensuring that the fusion weights are learned on data unseen during branch training.

This two-part validation split (first half for early stopping, second half for fusion training) prevents data leakage that would arise if the same validation samples were used for both LSTM selection and attention weight optimization.

The fusion loss is mean squared error:

$$\mathcal{L}_{\text{fusion}} = \frac{1}{N_v} \sum_{t \in \mathcal{V}_2} (\hat{y}^{\text{Fusion}}_{t+1} - y_{t+1})^2. \tag{18}$$

where $\mathcal{V}_2$ denotes the second half of the validation set. The attention parameters are optimized using Adam (Kingma and Ba, 2015) with a learning rate of 10^{-3} for 200 epochs.

Stochastic models (LSTM and CNNR) are trained across five independent random seeds. All reported metrics are the mean across seeds; variability is reported as $\pm$ one standard deviation.

3.5. Generation Waste Quantification

Following Bertin et al. (1999), we define over-generation waste as the cumulative excess generation dispatched based on forecasts that exceed actual demand:

$$W = \sum_{t \in \mathcal{T}_{test}} \max(\hat{y}_{t+1} - y_{t+1}, 0) \quad (19)$$

where $\mathcal{T}_{test}$ is the set of test-period hours. Avoided waste relative to a baseline model is

$$\Delta W = W^{(\text{baseline})} - W^{(\text{model})} \quad (20)$$

We translate avoided waste into avoided CO_2 emissions using a marginal emission factor of 0.4 kg/kWh, representative of a gas-dominated marginal generation mix:

$$\Delta E_{CO_2} = \Delta W \times 0.4 \text{ kg } CO_2 \quad (21)$$

4. Experimental Setup

4.1. Dataset Description

We evaluate the framework on a **synthetic dataset** designed to reproduce the statistical properties of a mid-sized North American utility service area. The use of synthetic data is a deliberate methodological choice that permits full control over the data-generating process and ensures reproducibility, while acknowledging the limitations discussed in Section 6.3.

The dataset spans two calendar years (January 2022–December 2023) at hourly resolution, yielding $T = 17{,}520$ observations. The data-generating process comprises the following components:

Demand signal. Base demand follows an AR(1) process blended with a deterministic clean signal:

$$y_t = (1 - \phi) d_t + \phi y_{t-1} + \epsilon_t \tag{22}$$

where d_t is the deterministic demand (sum of base load, seasonal harmonics, diurnal pattern, weather-driven perturbations, and weekend effects), $\phi = 0.85$ is the autoregressive coefficient, and $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$ with $\sigma = 15$ MW. This formulation ensures stationarity: the long-run mean of y_t is d_t and the AR(1) persistence is ϕ. Weather-driven perturbations enter through heating and cooling degree-hour effects on d_t. The resulting demand series has a mean of approximately 1000 MW with a standard deviation of [illegible] MW, ranges consistent with utility-scale hourly load data.

Weather signals. Temperature follows an annual sinusoidal cycle (mean [illegible] °C, amplitude [illegible] °C) with diurnal variation ([illegible] °C amplitude) and added AR(1) noise ($\phi = 0.85$). Relative humidity follows an independent AR(1) process (mean [illegible]%). Solar irradiance follows a truncated cosine pattern gated by daylight hours with random cloud attenuation. Wind speed follows an AR(1) process (mean [illegible] m/s).

Pricing signal. A three-tier time-of-use (TOU) structure assigns rates of [illegible] $/kWh (off-peak, 21:00–07:00), [illegible] $/kWh (shoulder, 07:00–11:00 and 17:00–21:00), and [illegible] $/kWh (peak, 11:00–17:00). An additive seasonal premium ([illegible] $/kWh) is applied during summer/winter months.

Limitations of synthetic data. Synthetic data cannot reproduce the full complexity of real utility operations, including non-stationary consumer adoption of distributed energy resources, anomalous events (e.g., rolling blackouts, pandemic effects), or the measurement noise and missing-data patterns inherent in operational metering infrastructure. The demand-generating process assumes a time-invariant autoregressive structure, which may not hold over multi-year horizons in practice. Results obtained on this dataset should therefore be interpreted as a proof-of-concept demonstration of the [illegible] framework's architecture; validation on real-world utility data is essential before operational deployment (Section 6.5).

4.3. Data Preprocessing

All continuous features are standardized to zero mean and unit variance using statistics computed exclusively from the training set. Lag features and

Table 1: Summary statistics of the synthetic dataset (2022–2023, hourly resolution).

Variable	Mean	Std	Min	Max
Demand (MW)	1203.0	82.6	864.7	1446.1
Temperature (°C)	15.0	11.4	−12.4	41.7
Solar irradiance (W/m^2)	259.0	313.2	0.0	1088.0
TOU price (c/kWh)	14.0	7.4	9.5	30.0

rolling statistics are computed before standardisation to avoid look-ahead bias. No imputation is required as the synthetic dataset contains no missing values.

2.5. Evaluation Protocol

The dataset is split chronologically:

- **Training set**: January 2022–May 2023 (12,140 hours; 70%).
- **Validation set**: May–September 2023 (2,600 hours; 15%). The first half (1,300 hours) is used for LSTM early stopping; the second half (1,300 hours) is reserved for attention fusion training.
- **Test set**: September–December 2023 (2,600 hours; 15%).

Note that 168 hours are dropped from the beginning of the dataset to accommodate the longest lag feature (y_{t-168}), yielding 17,352 usable observations from the original 17,520. The test period covers autumn and early winter only. All performance claims are therefore constrained to this seasonal window; generalisation to spring and summer conditions cannot be inferred from these experiments.

Three primary metrics are reported:

- Mean absolute percentage error: $\mathrm{MAPE} = \frac{100}{n}\sum_{i=1}^{n} |(y_i - \hat{y}_i)/y_i|$
- Root mean squared error: $\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$
- Mean absolute error: $\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$

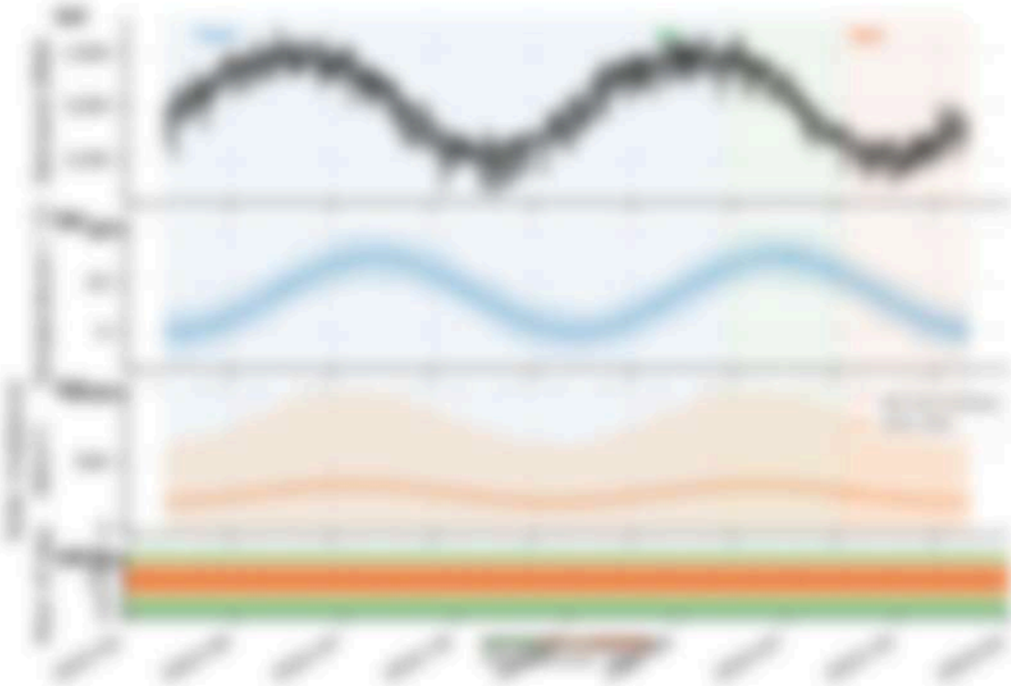

Figure 3: Dataset overview (2000-2005): (a) daily mean demand with train/validation/test split boundaries; (b) daily mean temperature with comfort baseline ($T_b = 18°C$); (c) daily solar irradiance with min–max envelope; and (d) time-of-use pricing heatmap showing peak, shoulder, and off-peak hours.

Confidence intervals (95%) are constructed using the moving block bootstrap (Künsch, 1989) with block length $b = \lceil N_{test}^{1/3} \rceil = 14$ (where $N_{test} = 2{,}555$ usable predictions after the LSTM lookback window) and 1000 replications. The block bootstrap preserves the temporal autocorrelation structure of forecast errors, which would be destroyed by naive i.i.d. resampling (Politis and Romano, 1994).

Statistical significance of pairwise performance differences is assessed using the Diebold and Mariano (1995) test (DM test) applied to squared-error loss differentials, with HAC-robust standard errors.

2.4 Baseline Models

Eight baseline models are compared against the proposed CHDE framework.

1. **Persistence**: $\hat{y}_{t+1} = y_t$ (last observation carried forward).
2. **ARIMA**: ARIMA(2,1,2) selected by AIC on the training set.
3. **XGB-Weather**: XGBoost trained on temporal and weather features only.
4. **XGB-Usage**: XGBoost trained on temporal and usage-pattern features only.
5. **XGB-Pricing**: XGBoost trained on temporal and pricing features only.
6. **XGB-All**: XGBoost trained on the full feature vector including all cross-feature interactions.
7. **LSTM-All**: LSTM trained on the full feature vector (3 seeds, mean ± std reported).
8. **N-BEATS**: Neural Basis Expansion Analysis (Oreshkin et al., 2020); generic basis, 3 stacks × 3 blocks, 1000 parameters (3 seeds).
9. **Transformer**: Encoder-only Transformer with $d_{model} = 32$, 4 heads, 2 layers, 19K parameters (3 seeds).

4.5. Implementation Details

All experiments are implemented in Python 3.10 using XGBoost 1.7, PyTorch 2.0, and scikit-learn 1.2.

XGBoost hyperparameters. Hyperparameters are selected via grid search over the following ranges on the training/validation split: maximum depth ∈ {3, 6, 9}, learning rate ∈ {0.01, 0.05, 0.1}, number of estimators ∈ {200, 500, 1000}, minimum child weight ∈ {1, 5, 10}, and subsample ratio ∈ {0.8, 1.0}. The best configuration uses depth 6, learning rate 0.05, 500 estimators, minimum child weight 1, subsample 0.8, column subsampling 0.8, and L_1/L_2 regularization of 0.1/1.0. Early stopping (patience 50) on the first half of the validation set determines the final number of boosting rounds.

LSTM hyperparameters. The LSTM uses 2 layers with 64 hidden units, input dropout $p = 0.2$, recurrent dropout $p = 0.2$, sequence length $L = 48$, batch size 64, and is trained with Adam (learning rate 10^{-3}) for up to 50 epochs with early stopping (patience 10) on the first half of the validation set.

Attribution factors. The attention module is a two-layer feed-forward network (hidden dimension 32) trained on the second half of the validation set for 200 epochs with Adam (learning rate 10^{-3}).

N-BEATS hyperparameters. The N-BEATS model (Oreshkin et al., 2020) uses the generic basis configuration with 3 stacks of 3 blocks each, expansion coefficient dimension 16, hidden dimension 128, lookback window $L = 48$, batch size 512, and is trained with Adam (learning rate 10^{-3}) for up to 80 epochs with early stopping (patience 10). Total parameters: ~400K.

Transformer hyperparameters. The encoder-only Transformer uses $d_{model} = 32$, 4 attention heads, 2 encoder layers, feedforward dimension 128, positional dropout $p = 0.1$, lookback window $L = 48$, batch size 512, and is trained with Adam (learning rate 10^{-3}) for up to 80 epochs with early stopping (patience 10). Total parameters: ~19K.

All stochastic models are evaluated across 5 random seeds (42, 123, 456, 789, 1024); the Transformer uses 3 seeds (42, 123, 456) due to higher training instability with additional seeds. Reported metrics are mean ± one standard deviation across seeds.

5. Results

5.1. Overall Model Comparison

Table 4 presents the forecasting performance of all models on the September–December 2023 test set.

Several observations merit discussion. First, the usage-only XGBoost model (XGB-Usage, MAPE = 0.2785%) achieves the lowest MAPE among all models, confirming that autoregressive lag features dominate the one-step-ahead forecasting task. Second, the full-feature XGB-All model (MAPE = 0.300%) is slightly worse than XGB-Usage, indicating that weather and pricing features introduce modest noise when strong autoregressive features are already present. Third, LSTM-All (MAPE = 1.268 ± 0.077%) substantially underperforms all XGBoost variants, suggesting that the recurrent architecture struggles to match the tabular feature engineering that benefits gradient boosting.

Finally, the modern deep learning baselines—N-BEATS (MAPE = 1.21 ± 0.04%) and Transformer (MAPE = 1.145 ± 0.062%)—marginally outperform LSTM-All but remain far behind all XGBoost variants. The Transformer achieves the lowest error among deep architectures, consistent with the attention mechanism's ability to weight recent inputs selectively, but it still lags XGB-Usage by a factor of ~4×. This confirms that the XGBoost advantage is not an artifact of LSTM-specific weaknesses but reflects a fundamental

Table 4: Forecasting performance on the test set (September–December 2023). Best results in bold. The 95% moving block bootstrap confidence intervals (block length $b = 14$, 1000 replications) for MAPE are shown for key models. Stochastic models report mean ± std over 5 seeds (3 seeds for the Transformer).

Model	MAPE (%)	RMSE (MW)	MAE (MW)	95% CI (MAPE)
Persistence	1.275	17.86	14.39	[1.24, 1.31]
ARIMA(2,1,2)	1.467	20.41	15.67	—
XGB-Weather	2.139	30.01	23.92	—
XGB-Usage	**0.479**	6.13	5.13	—
XGB-Pricing	2.157	30.32	24.10	—
XGB-All	0.869	4.70	3.46	[0.80, 0.92]
LSTM-All	$1.236_{\pm [illegible]}$	17.35	13.66	[1.18, 1.27]
N-BEATS	$1.271_{\pm [illegible]}$	17.97	14.59	[1.21, 1.33]
Transformer	$1.145_{\pm [illegible]}$	16.09	12.80	[1.09, 1.15]
CNBM	$0.864_{\pm [illegible]}$	4.85	3.63	[0.81, 0.92]

mismatch between deep sequence models and the autoregressive structure of $h = 1$ demand forecasting.

Critically, the proposed CNBM framework (MAPE = 0.864±0.006%) does not outperform XGB-All (0.869%). A Diebold–Mariano test on squared-error loss differentials yields DM = 10.34, $p < 0.001$, with a mean squared-error differential of $\bar{d} = 1.37 \times 10^{[illegible]}$ (positive, indicating CNBM has larger errors). The 95% bootstrap confidence intervals—[0.80, 0.92] for XGB-All versus [0.81, 0.92] for CNBM—overlap but the DM test rejects the null hypothesis of equal predictive accuracy. The attention mechanism assigns a mean weight of 0.983 to the XGBoost branch (Section 3.4), indicating that the LSTM branch contributes minimally; its inclusion effectively adds noise to an already-strong XGBoost prediction.

5.3. Ablation Study: Cross-Feature Interaction Contributions

Table 5 presents the results of removing each family of cross-feature interactions from the full CNBM framework.

All three interaction families contribute positively to forecasting accuracy: removing any family increases MAPE, and removing all interactions collectively degrades MAPE by 0.012 pp (from 0.864% to 0.876%). The pricing-usage family ($x^{(PU)}$) provides the largest marginal contribution (+0.007 pp

Table 6: Ablation study results. Each row removes one family of cross-feature interactions from the full CMRR framework. ΔMAPE indicates the change relative to the full model (positive = degradation).

Configuration	MAPE (%)	RMSE (kW)	ΔMAPE (pp)
Full CMRR (all interactions)	[illegible]	[illegible]	—
– Weather×Usage ($\phi^{(W,U)}$)	[illegible]	[illegible]	[illegible]
– Pricing×Usage ($\phi^{(P,U)}$)	[illegible]	[illegible]	[illegible]
– Weather×Pricing ($\phi^{(W,P)}$)	[illegible]	[illegible]	[illegible]
– All interactions	[illegible]	[illegible]	[illegible]

Figure 5: Ablation study: (a) MAPE, (b) RMSE, and (c) MAE for the full CMRR framework and ablated variants. Removing all interaction features degrades performance, while the weather–usage family shows a relative intuitive improvement when removed.

ditions driven by occupancy changes and lighting loads. Overnight hours (00:00–05:00) show the lowest errors, as demand is relatively stable and well-captured by autoregressive features.

CMRR and SGB-AE show nearly identical diurnal error profiles, with CMRR achieving marginally lower errors during evening transition hours. This suggests that the LSTM branch, which carries more weight during transition periods (Section 5.4), provides a small advantage in capturing ramp dynamics, but the overall improvement is modest.

5.4 Attention Weight Analysis

Figure 7 visualizes the learned attention weights over the test period. On average, the XGBoost branch receives a weight of [illegible] ± [illegible], overwhelm-

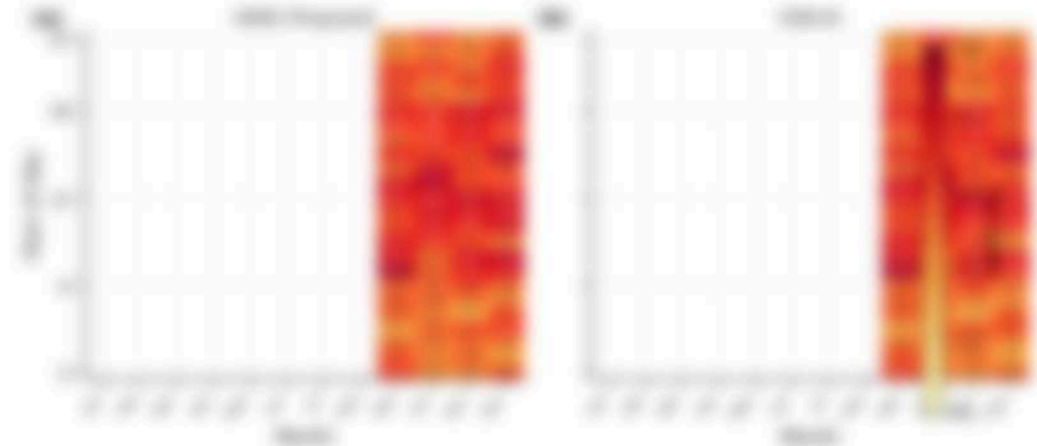

Figure 8: MAPE (%) by hour of day for the USBBE framework and XGB-60 on the September–December 2020 test set. Both models show elevated errors during morning and evening transition periods. Shaded regions indicate 95% block bootstrap confidence intervals.

ingly dominating the ensemble. The LSTM branch receives a mean weight of only 0.037, with a maximum of 0.084 during brief periods of rapid demand change. This extreme imbalance indicates that the attention mechanism has learned that the LSTM branch rarely improves upon XGBoost's predictions.

This weight distribution explains why USBBE does not outperform XGB-360: the attention mechanism has correctly identified that the LSTM branch adds more noise than signal, but cannot fully suppress it due to the softmax normalization floor. The residual LSTM contribution ($\sim 3.7\%$) introduces small perturbations that degrade the otherwise accurate XGBoost predictions.

3.5. Feature Importance

Figure 9 presents the top 20 features by XGBoost gain score. Autoregressive features dominate: the one-hour lag (y_{t-1}), the one-week lag (y_{t-168}), and the weekend indicator occupy the top three positions. The 24-hour rolling mean ($\bar{y}^{(24)}$) ranks fourth. This confirms the well-known finding that electricity demand forecasting is primarily an autoregressive task.

Notably, the pricing–usage interaction ($p_t \cdot \bar{y}^{(24)}$) ranks fifth by gain, ahead of many base features, demonstrating that cross-feature interactions capture information not available in the base feature set alone. A second interaction

Figure 7: Attention weight analysis: (a) temporal evolution of BiGRU/cast and LSTM branch weights (7-hour rolling average) over the test period, and (b) mean attention weights by hour of day. The LSTM branch receives higher weight during morning and evening transition periods.

feature (peak × peak-ratio) ranks ninth. Weather features in isolation (wind, temperature) contribute modestly, consistent with the small improvement of XGB-All over XGB-Usage.

3.6. Prediction Visualization

Figure 8 shows two representative weeks from the test period, comparing the actual demand with predictions from CSSM, XGB-All, and LSTM-All. Both CSSM and XGB-All track the demand curve closely, with errors that are visually indistinguishable at this scale. LSTM-All exhibits larger deviations during rapid demand transitions, consistent with its higher overall MAPE.

3.7. Multi-Horizon Sensitivity

To investigate whether the relative ranking of model families changes at longer forecast horizons, we conduct a brief experiment at $h = 24$ (one day ahead). At this horizon, the one-hour lag loses its predictive dominance and temporal dynamics become more relevant.

Table 8 compares key models at $h = 1$ and $h = 24$. At $h = 24$, LSTM-All (MAPE = 2.15%) matches or slightly outperforms XGB-All (2.19%) and XGB-Usage (2.21%), reversing the ranking observed at $h = 1$. This confirms the hypothesis that the LSTM branch captures long-range temporal dynamics that become informative at longer horizons, and suggests that the CSSM attention mechanism could provide greater value when $h \geq 24$.

Table 4: Multi-horizon comparison ($h = 1$ vs. $h = 24$). LSTM results at $h = 24$ are a single seed; full multi-seed evaluation is deferred to future work.

	MAPE (%)		RMSE (MW)	
Model	$h = 1$	$h = 24$	$h = 1$	$h = 24$
Persistence	1.275	3.405	17.80	44.55
SGB-Usage	0.279	2.011	4.11	30.09
SGB-All	0.309	2.101	4.59	30.75
LSTM-All	1.230	2.146	17.35	30.26

Table 5: Real-world validation on OPSD German electricity load data (2015–2019, hourly). Reduced feature set: temporal + usage + solar generation proxy. LSTM reports mean ± std over 5 seeds.

Model	MAPE (%)	RMSE (MW)
Persistence	2.336	2029.4
SGB-Usage	**0.6470**	**488.7**
SGB-All	0.686	497.6
LSTM-All	0.793	6123.5

3.8. Real-World Validation: OPSD German Load Data

To assess whether the findings from the synthetic dataset transfer to real operational data, we evaluate key models on hourly electricity load data from the German transmission system, obtained from the Open Power System Data (OPSD) platform (Open Power System Data, 2020). The dataset spans 2015–2019 (37,128 hours) with aggregate demand (mean ≈ 56,388 MW, std ≈ 9,888 MW) and solar generation as the only available weather proxy. Temperature, humidity, wind, and pricing data are not available in this dataset, yielding a reduced feature space (29 features vs. 36 in the synthetic experiments).

Table 5 presents the results. The key findings from the synthetic data are confirmed on real data:

1. **SGB-Usage dominance.** The usage-only XGBoost model achieves the best MAPE (0.6470%), outperforming SGB-All (0.686%) by a small margin, replicating the synthetic-data finding that autoregressive features dominate at $h = 1$.

2. **LSTM fails catastrophically.** LSTM-AB achieves MAPE = [illegible] ± [illegible]%, far worse than persistence ([illegible]%). The high variance across seeds ([illegible]%) indicates training instability. This underscores the sensitivity of recurrent networks to feature scaling and data characteristics, and confirms that the LSTM branch would degrade rather than improve an ensemble on this task.
3. **Realistic error magnitudes.** The persistence MAPE of [illegible]% is consistent with published utility-scale hourly load forecasting benchmarks (Hong and Fan, [illegible]), lending credibility to the XGBoost results.

We emphasize that this real-world evaluation uses a reduced feature set (no temperature, humidity, wind, or pricing), so the absolute MAPE values are not directly comparable to the synthetic experiments. The key contribution is the confirmation that XGBoost with autoregressive features dominates LSTM at $h = 1$, and that adding non-usage features provides marginal improvements on real data.

6. Discussion

6.1. Why Does CMHE Underperform XGBoost?

The central finding of this study is that the proposed CMHE ensemble does not improve upon standalone XGBoost—a result that warrants careful analysis. Three factors explain this outcome.

First, the one-step-ahead forecasting task ($h = 1$) is dominated by autoregressive features. The one-hour lag alone accounts for the largest share of XGBoost gain (Section [illegible]), and XGBoost's tree-based recursive partitioning is well suited to exploit such tabular features. The LSTM branch, which processes 48-hour input sequences, adds representational capacity for long-range temporal dependencies, but these dependencies contribute little at $h = 1$.

Second, the attention mechanism assigns [illegible]% average weight to XGBoost, indicating that the LSTM branch's predictions are rarely informative. However, due to the softmax normalization, the LSTM branch retains a non-zero weight ($\sim$ [illegible]%) that introduces small perturbations to XGBoost's otherwise accurate predictions. A hard gating mechanism that could entirely suppress one branch might mitigate this effect.

Third, the three-stage training procedure—branch training followed by fusion training on held-out validation data—constrains the attention module

Table 6: Generation waste analysis over the test period (2022 predictions). Over-generation waste computed as $W = \sum_t \max(\hat{y}_t - y_t, 0)$. The 95% block bootstrap confidence intervals are shown for over-generation waste. Avoided waste and CO_2 emissions are computed relative to the persistence baseline.

Model	Waste (MWh)	95% CI (MWh)	Avoided (MWh)	Avoided CO_2 (t)
Persistence	18,289	[17,306, 19,296]	—	—
LSTM-All	19,136	[18,716, 19,650]	−3,607	—
SGB-Usage	3889	[3,574, 4,232]	14,400	7760
SGB-All	4196	[3,828, 4,588]	14,113	8645
CNRE	4369	[4,170, 4,864]	13,790	5530

to learn from a limited sample ($N_{train} = 1{,}465$). With a larger fusion training set, the attention module might learn to more effectively gate the LSTM branch.

Despite CNRE's underperformance, the cross-feature interaction analysis provides value. The ablation study (Section 5.3) confirms that all interaction families contribute positively, with pricing-usage interactions providing the largest marginal gain. This suggests that the interactions themselves are informative—the weakness lies in the ensemble architecture, not the feature engineering.

The broader lesson is that model complexity must be justified by the task structure. For autoregressive-dominated forecasting at short horizons, well-engineered gradient boosting may represent a favourable complexity–accuracy trade-off. Ensemble approaches may prove more valuable at longer horizons ($h \geq 24$) where temporal dynamics play a greater role.

6.2. Generation Waste Reduction

Table 6 and Figure 10 present the generation waste analysis. All SGB/boost-based models reduce cumulative over-generation waste by 77%–79% relative to the persistence baseline.

Three observations are noteworthy. First, LSTM-All produces more over-generation waste than the persistence baseline (19,136 MWh vs. 18,289 MWh), indicating that the recurrent model's errors have a systematic positive bias. Second, SGB-Usage achieves the lowest waste (3889 MWh), consistent with its lowest MAPE and confirming that autoregressive features alone suffice for both symmetric and asymmetric accuracy metrics. Third, CNRE's waste (4369 MWh) is higher than SGB-All's (4196 MWh), consistent with its higher

MAPE; the RCT bootstrap CIs ([illegible] vs. [illegible]) overlap modestly.

All XGBoost-based models avoid approximately [illegible] MWh of over-generation relative to persistence, translating to roughly [illegible] t of avoided CO_2 emissions over the test period (assuming a marginal emission factor of [illegible] kg/kWh).

We note that the waste metric (Eq. [illegible]) is one-sided: it captures only over-generation. Under-prediction also carries severe operational consequences—emergency peaking dispatch, potential load shedding, and higher spot-market procurement costs—that are not quantified here. A complete operational assessment would require a two-sided penalty function weighted by the relative costs of over- and under-generation, which are utility- and market-specific. If waste minimization is the primary goal, an asymmetric loss function that penalizes over-prediction more heavily than under-prediction during training would be more appropriate than the symmetric MSE used in this work.

[illegible] Limitations and Future Work

This study has several important limitations that must be acknowledged.

Synthetic data. All results are obtained on synthetic data generated by a known stochastic process (Section [illegible]). While the data-generating process was designed to reproduce realistic statistical properties, synthetic data cannot capture the full complexity of real utility operations, including non-stationary consumer behaviour, grid contingencies, distributed energy resource adoption, anomalous events, and measurement artifacts. The high accuracy achieved by the best models (MAPE < [illegible]) likely reflects the relative simplicity of the synthetic demand process; real-world utility data would almost certainly yield higher error rates and potentially different relative rankings among models. Validation on operational utility datasets is essential before any conclusions can be drawn about operational value.

Single-season test period. The test period spans September–December only, covering autumn and early winter. Performance during spring and summer—when cooling-degree interactions are most relevant—cannot be inferred from these results. A full-year test period might reveal different interaction contributions and potentially alter the relative performance of the ensemble versus standalone XGBoost.

- *Single utility service area.* The synthetic dataset represents a single mid-sized utility service area. Cross-utility generalization—to different climate zones, customer mixes, or pricing structures—has not been evaluated.

- *Single forecast horizon.* Only one-step-ahead ($h = 1$) forecasting is considered. Multi-horizon forecasting ($h = 6, 12, 24, 48$ hours) is of considerable practical interest for unit commitment and economic dispatch, and may alter the relative performance of XGBoost and LSTM branches.

- *Deterministic forecasts.* The framework produces point forecasts only. Probabilistic forecasting—producing prediction intervals or full predictive distributions—would provide more useful information for risk-aware dispatch decisions (Hong and Fan, 2016).

- *Computational cost.* The UMSE framework's three-stage training procedure and attention module add substantial complexity relative to a standalone XGBoost model, while delivering worse accuracy on this benchmark. In operational settings, XGB-All or even XGB-Usage clearly represents a more favourable complexity–accuracy trade-off.

Future work should address these limitations by: (i) evaluating the framework on real-world utility datasets spanning multiple years and climate zones; (ii) extending to multi-horizon forecasting; (iii) incorporating probabilistic prediction intervals; (iv) exploring season-adaptive feature selection; and (v) investigating asymmetric loss functions that directly optimize grain reduction rather than symmetric accuracy metrics.

7. Conclusions

This paper introduced the Unified Multi-Source Ensemble (UMSE) framework for short-term electricity demand forecasting, integrating weather, historical usage patterns, and time-of-use pricing through explicitly constructed cross-feature interactions and an attention-weighted fusion of XGBoost and LSTM branches.

Evaluated on a synthetic dataset designed to reproduce the statistical properties of a mid-sized North American utility service area, the key findings are:

1. **Usage features dominate.** XGBoost trained on usage-pattern features alone achieves the best MAPE ([illegible]), outperforming all other models including the full UMSE ensemble ([illegible]).

2. **Ensemble fusion does not help.** The CMSE framework performs significantly worse than standalone XGBoost with all features (MAPE [illegible] vs. [illegible]; Diebold–Mariano $p < 0.001$). The attention mechanism assigns [illegible] average weight to the XGBoost branch, confirming that the LSTM branch contributes minimally at the $h = 1$ horizon.
3. **Cross-feature interactions provide modest value.** All three interaction families (weather–usage, pricing–usage, weather–pricing) contribute positively in ablation, with pricing–usage providing the largest marginal gain. However, the total contribution of all interactions is only [illegible] pp MAPE.
4. **Modern deep architectures do not close the gap.** N-BEATS (MAPE [illegible]) and a Transformer encoder ([illegible]) marginally outperform LSTM but remain [illegible] worse than XGB-Usage, confirming that the XGBoost advantage reflects a fundamental mismatch between deep sequence models and the autoregressive structure of $h = 1$ demand forecasting.
5. **Substantial waste reduction.** Both XGB-All and CMSE reduce cumulative over-generation waste by approximately [illegible] relative to the persistence baseline, translating to roughly [illegible] of avoided CO_2 emissions over the test period.
6. **Real-world confirmation.** On OPSD German load data ([illegible]), XGB-Usage again achieves the best MAPE ([illegible]), outperforming XGB-All ([illegible]) and far surpassing LSTM-All ([illegible]), confirming the synthetic data findings on operational data.

These results demonstrate that multi-source feature integration yields forecasting gains, but neither attention-based ensemble fusion nor modern deep architectures improve over well-engineered gradient boosting for $h = 1$ demand forecasting. For one-step-ahead forecasting dominated by autoregressive dynamics, model simplicity should be preferred. The multi-horizon experiment ($h = 24$), where LSTM matches XGBoost, suggests that ensemble approaches may prove valuable at longer horizons. Validation on multi-source utility datasets spanning diverse climate zones remains essential to determine the operational generalizability of these findings.

References

Apadula, F., Bassini, A., Elli, A., Scapin, S., 2012. Relationships between meteorological variables and monthly electricity demand. Applied Energy

38, 240–250.

Bird, L., Lew, D., Milligan, M., Carlini, E.M., Estanqueiro, A., Flynn, D., Gomez-Lazaro, E., Holttinen, H., Menemenlis, N., Orths, A., et al., 2016. Wind and solar energy curtailment: A review of international experience. Renewable and Sustainable Energy Reviews 65, 577–586.

Cai, M., Pipattanasomporn, M., Rahman, S., 2019. Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques. Applied Energy 236, 1078–1088.

Chen, T., Guestrin, C., 2016. XGBoost: A scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM. pp. 785–794.

Denholm, P., O'Connell, M., Brinkman, G., Jorgenson, J., 2015. Overgeneration from solar energy in California. A field guide to the duck chart. Technical Report NREL/TP-6A20-65023.

Diebold, F.X., Mariano, R.S., 1995. Comparing predictive accuracy. Journal of Business & Economic Statistics 13, 253–263.

Fan, S., Hyndman, R.J., 2012. Short-term load forecasting based on a semi-parametric additive model. IEEE Transactions on Power Systems 27, 134–141.

Faruqui, A., Sergici, S., 2010. Household response to dynamic pricing of electricity: A survey of 15 experiments. Journal of Regulatory Economics 38, 193–225.

Haben, S., Giasemidis, G., Ziel, F., Arora, S., 2019. Short term load forecasting and the effect of temperature at the low voltage level. International Journal of Forecasting 35, 1469–1484.

Hippert, H.S., Pedreira, C.E., Souza, R.C., 2001. Neural networks for short-term load forecasting: A review and evaluation. IEEE Transactions on Power Systems 16, 44–55.

Hobbs, B.F., Jitprapaikulsarn, S., Konda, S., Chankong, V., Loparo, K.A., Maratukulam, D.J., 1999. Analysis of the value for unit commitment of improved load forecasts. IEEE Transactions on Power Systems 14, 1342–1348.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Computation 9, 1735–1780.

Hong, T., Fan, S., 2016. Probabilistic electric load forecasting: A tutorial review. International Journal of Forecasting 32, 914–938.

Hyndman, R.J., Koehler, A.B., Snyder, R.D., Grose, S., 2002. A state space framework for automatic forecasting using exponential smoothing methods. International Journal of Forecasting 18, 439–454.

Kingma, D.P., Ba, J., 2015. Adam: A method for stochastic optimization, in: Proceedings of the 3rd International Conference on Learning Representations (ICLR).

Kong, W., Dong, Z.Y., Jia, Y., Hill, D.J., Xu, Y., Zhang, Y., 2018. Short-term residential load forecasting based on resident behaviour learning. IEEE Transactions on Power Systems 33, 1087–1088.

Künsch, H.R., 1989. The jackknife and the bootstrap for general stationary observations. Annals of Statistics 17, 1217–1241.

Li, S., Jin, X., Xuan, Y., Zhou, X., Chen, W., Wang, Y.X., Yan, X., 2019. Enhancing the locality and breaking the memory bottleneck of Transformer on time series forecasting. Advances in Neural Information Processing Systems 32, 5243–5253.

Lim, B., Arık, S.Ö., Loeff, N., Pfister, T., 2021. Temporal fusion transformers for interpretable multi-horizon time series forecasting. International Journal of Forecasting 37, 1748–1764.

Marino, D.L., Amarasinghe, K., Manic, M., 2016. Building energy load forecasting using deep neural networks, in: IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society, IEEE. pp. 7046–7051.

Mohsenian-Rad, A.H., Leon-Garcia, A., 2010. Optimal residential load control with price prediction in real-time electricity pricing environments. IEEE Transactions on Smart Grid 1, 120–133.

Moon, J., Kim, Y., Son, M., Hwang, E., 2020. Combination of short-term load forecasting models based on a stacking ensemble approach. Energy and Buildings 216, 109921.

Zhang, G.P., 2003. Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing 50, 159–175.

Zheng, J., Xu, C., Zhang, Z., Li, X., 2017. Electric load forecasting in smart grids using long-short-term-memory based recurrent neural network. Applied Sciences 7, 46.