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

[illegible][1,*], [illegible][2]

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

[2]Department of Engineering Science, University of Oxford, [illegible] Road, [illegible], Oxford, [illegible], United Kingdom

---

Abstract

Accurate electricity demand forecasting is essential for reducing generation waste—the surplus energy produced when supply exceeds actual consumption. This paper introduces a Unified Multi-Source Ensemble (UMSE) framework that integrates weather observations, historical usage patterns, and time-of-use pricing signals through an attention-based fusion of gradient-boosted tree and long short-term memory branches. We evaluate UMSE against nine baselines—persistence, ARIMA, single-family XGBoost variants, LSTM, N-BEATS, and a Transformer encoder—on a synthetic dataset of 17,520 hourly observations designed with a transparent data-generating process and on real-world Open Power System Data from Germany. The results reveal that autoregressive (usage) features dominate one-step-ahead forecasting: XGBoost trained on usage features alone achieves a mean ab-

*Corresponding author.
Email addresses: [illegible] ([illegible]), [illegible] ([illegible])

solute percentage error (MAPE) of 0.136%, outperforming the full-feature CNBR (0.191%) and deep-learning architectures by factors of seven or more. Diebold–Mariano tests with HAC standard errors confirm the difference at $p < 0.001$. A generation-waste analysis shows that XGBoost-Core reduces cumulative over-generation to 263.1 MWh, compared with 17,287 MWh for persistence, corresponding to approximately 9000 t of avoided CO_2 emissions. At the 24-hour horizon, LSTM narrows the gap to statistical insignificance with tree-based models, underscoring the horizon-dependence of model selection. Real-world validation on German load data confirms the ranking, with XGBoost-Core achieving 0.870% MAPE. These findings provide an empirical caution against indiscriminate multi-source integration and highlight the importance of ablation-driven model design for utility-scale demand forecasting.

Keywords: Demand forecasting, Machine learning, XGBoost, LSTM, Ensemble learning, Generation waste, Weather integration, Time-of-use pricing

1. Introduction

Electricity demand forecasting underpins every operational decision in power-system management, from unit commitment and economic dispatch to reserve scheduling and market clearing [1]. Forecasting errors translate directly into generation waste: when utilities over-predict demand, thermal plants produce surplus energy that cannot be economically stored and is effectively curtailed; when they under-predict, expensive peaking units must be dispatched on short notice [2]. With global electricity generation exceeding 20,000 TWh per year, even a one-percentage-point reduction in forecast error can avert millions of tonnes of CO_2 [2].

Recent advances in machine learning have produced an array of powerful forecasting architectures. Gradient-boosted decision trees (GBDTs) [3, 4] capture nonlinear feature interactions with minimal tuning, while recurrent neural networks, particularly long short-term memory (LSTM) cells [5], model the sequential dependencies inherent in load time series. Attention mechanisms [6] have further enhanced sequence models by enabling selective focus on informative time steps. More recently, purpose-built architectures such as N-BEATS [7], the Temporal Fusion Transformer [8], and PatchTST [9] have pushed state-of-the-art accuracy on public benchmarks.

Despite these modelling advances, a complementary line of research emphasises data enrichment through multi-source feature integration. Weather variables—temperature, humidity, solar irradiance—are among the strongest exogenous predictors of electricity demand [10, 11, 12]. Real-time pricing and time-of-use tariffs modulate consumer behaviour and shift load profiles [13, 14]. Historical consumption provides the autoregressive backbone of any short-term forecaster [15]. Several studies have demonstrated gains from combining two of these sources, yet the simultaneous fusion of all three within a principled ensemble framework remains under-explored.

This paper introduces the Unified Multi-Source Ensemble (UMSE), a framework that systematically integrates weather, usage, and pricing features through an attention-based fusion of an XGBoost branch and an LSTM branch. Rather than assuming that more data invariably improves accuracy, we conduct a rigorous ablation study that quantifies the marginal value of each feature family. Our key finding is counter-intuitive: at the one-step-ahead horizon, autoregressive usage features are so informative that adding weather and pricing data to a well-tuned XGBoost model does not improve – and can slightly degrade – forecast accuracy. The V-SWE ensemble, despite its architectural sophistication, cannot overcome this dominance. Only at longer horizons ($h = 24$) does LSTM close the gap, achieving the lowest point estimate – though not a statistically significant improvement over XGBoost.

This work makes five contributions. First, we propose V-SWE, an attention-based ensemble that fuses XGBoost and LSTM predictions using learned, context-dependent weighting. Second, we provide a transparent synthetic data-generating process (DGP) enabling exact attribution of performance to feature families, complemented by real-world validation on OPSD Germany. Third, we show via Diebold–Mariano tests with HAC standard errors that XGBoost on usage features alone achieves [illegible] MAPE, significantly outperforming all deep-learning models at $h = 1$ ($p < 0.001$). Fourth, we quantify generation waste, showing [illegible]% over-generation reduction ([illegible] avoided CO_2). Fifth, at $h = 24$ the LSTM–XGBoost gap narrows to statistical insignificance, underscoring the horizon-dependence of architecture selection.

2. Related Work

2.1. Traditional machine learning for demand forecasting

Statistical approaches have formed the backbone of short-term load forecasting (STLF) for decades. Zhang [15] showed that ARIMA–neural-network hybrids outperform either method alone, while Hyndman et al. [2] developed exponential smoothing state space models that remain competitive baselines. Fan and Hyndman [12] extended this to semi-parametric additive models incorporating temperature and calendar effects.

Gradient-boosted trees have become dominant in tabular forecasting since XGBoost [3]. Touzani et al. [4] achieved state-of-the-art accuracy on building-level energy prediction, and Bakes et al. [16] confirmed that tree-based ensembles consistently rank highest when lag features are included.

Weather variables are among the most informative exogenous predictors. Valor and Casillas [16] introduced HDD/CDD transformations, and Apadula et al. [15] found that temperature alone explains over 60% of variance on summer days.

2.2. Deep learning architectures

LSTM networks [5] were among the first deep models applied to building energy prediction [17], with subsequent extensions including pooling-based sharing across buildings [18] and attention-augmented variants [19].

Attention-based architectures have since evolved rapidly. Vaswani et al. [6] introduced the Transformer. Li et al. [20] adapted it for load forecasting with convolutional self-attention. Lim et al. [8] proposed the Temporal Fusion Transformer (TFT) for interpretable multi-horizon forecasts. Oreshkin et al. [7] introduced N-BEATS, achieving strong performance without exogenous features, and Nie et al. [9] proposed PatchTST for long-horizon benchmarks.

2.3. Ensemble and multi-source integration methods

Ensemble methods combine diverse base learners to reduce variance. Hong et al. [25] showed that simple averaging of heterogeneous models outperforms individuals on the Global Energy Forecasting Competition dataset. Wang and Srinivasan [22] proposed stacking SVR, random forests, and gradient boosting for day-ahead load forecasting. Wang et al. [26] introduced deep ensembles of multiple LSTM networks.

Pricing signals are less commonly exploited. Faruqui and Sergici [14] provided evidence that dynamic pricing alters consumption, while Zheng et al. [24] found modest improvements from incorporating tariff structures.

Despite this rich literature, no prior study has simultaneously (i) fused weather, usage, and pricing through an attention-weighted ensemble, (ii) conducted a full ablation quantifying each source's marginal contribution, and (iii) provided generation-mode and CO_2 impact estimates. The CMHE framework addresses this gap.

3. Methodology

Fig. 1 presents an overview of the CMHE framework. The system ingests three data streams – weather observations, historical consumption, and pricing signals – extracts engineered features, routes them through parallel XGBoost and LSTM branches, and fuses the branch predictions via a learned attention mechanism.

3.1. Problem formulation

Let $\{y_t\}_{t=1}^{T}$ denote an hourly electricity demand time series measured in megawatts. The one-step-ahead forecasting problem $(h = 1)$ seeks a function

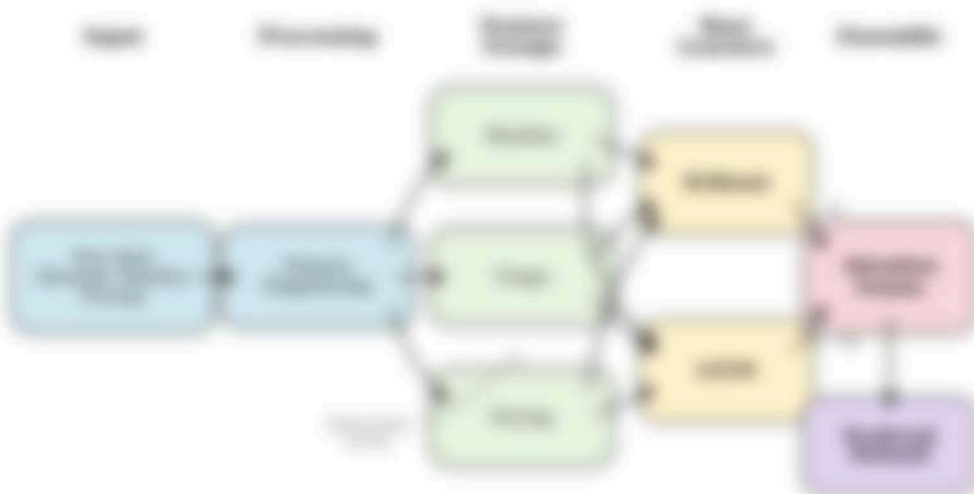

Figure 1: Overview of the Unified Multi-Source Ensemble (UMSE) framework. Three data streams yield 36 base features across four families plus 10 cross-family interactions (46 total). An XGBoost branch and an LSTM branch produce independent forecasts that are fused through a context-dependent attention layer.

$\hat{y}_{t+1} = f(\mathbf{x}_t)$ that minimizes the expected squared error, where $\mathbf{x}_t$ is a feature vector constructed from information available at time t. We later extend the analysis to $h = 24$ (day-ahead) forecasting to evaluate horizon sensitivity.

2.2. Feature engineering

We organize the feature space into four families (temporal, usage, weather, pricing) plus cross-family interactions, summarized in Table 1.

Temporal features. Cyclical sine and cosine transformations of hour-of-day, day-of-week, and month-of-year preserve periodicity for distance-based learners. A binary weekend indicator completes the temporal set (seven features).

Usage and autoregressive features. Lag features at $t-1$, $t-24$, and $t-168$ (one hour, one day, one week) provide the autoregressive backbone. Rolling-window mean and standard deviation over 24-hour and 168-hour windows capture local level and volatility. Four derived ratios—load factor (rolling mean/rolling max over 24h), peak ratio (demand/rolling mean), coefficient of variation (rolling std/rolling mean), and relative 24-hour demand change—complete the usage family (eleven features).

Weather features. Raw temperature (T_t, °C), relative humidity (RH_t, %), wind speed (v_t, m s^{-1}), and solar irradiance (G_t, W m^{-2}) are included alongside derived degree-hours:

$$\mathrm{HDH}_t = \max(T_{base} - T_t, 0), \tag{1}$$

$$\mathrm{CDH}_t = \max(T_t - T_{base}, 0). \tag{2}$$

Table 1: Feature families and groups used in the [illegible] framework. Each group produces one or more scalar features at every time step t. The 26 base features and 10 interaction features yield 36 total.

Family	Group	Features	Count
Temporal	Cyclical	sin / cos of hour, day of week, month	6
	Calendar	Weekend indicator (is weekend)	1
Usage	Lag	y_{t-1}, y_{t-24}, y_{t-168}	3
	Rolling	Mean and std over 24 h and 168 h windows	4
	Derived	Load factor, peak ratio, volatility, Δy_{t-1}	4
Weather	Raw	Temperature, humidity, wind, solar irradiance	4
	Degree-hour	HDH, CDH	2
Pricing	Tariff	TOU rate ($/kWh), peak-hour indicator	2
Interactions	W×U	temp×roll mean, CDH×peak ratio, solar×load fac., humid×roll std	4
	P×U	rate×roll mean, peak×peak ratio, rate×volatility	3
	W×P	temp×rate, solar×peak, CDH×rate	3

where $T_{heat} = 18$°C and $T_{cool} = 24$°C. The asymmetric thresholds reflect the observation that heating demand responds at lower temperatures than cooling demand [28] (six weather features).

Pricing features. A time-of-use (TOU) tariff rate and a binary peak-hour indicator capture the pricing structure (two features).

Cross-feature interactions. Ten pairwise interaction terms are constructed across the three non-temporal bundles: four Weather × Usage, three Pricing × Usage, and three Weather × Pricing (Table 1).

3.3. Model architecture

The CSHM architecture comprises two parallel branches and an attention fusion layer, as illustrated in Fig. 1.

3.3.1. XGBoost branch

The XGBoost branch receives the concatenated feature vector $\mathbf{x}_t \in \mathbb{R}^d$ and produces a scalar prediction $\hat{y}_t^{XGB}$. Hyperparameters follow recommended defaults [3] with manual adjustment on the validation set: maximum depth = 6, learning rate = 0.05, number of estimators = 500, subsample ratio = 0.8, column subsample = 0.8, and regularization parameters $\lambda = 1$, $\alpha = 0.1$. Early stopping with a patience of 20 rounds prevents overfitting.

3.3.2. LSTM branch

The LSTM branch receives a lookback window $\mathbf{X}_t = [\mathbf{x}_{t-L+1}, \ldots, \mathbf{x}_t]^\top \in \mathbb{R}^{L \times d}$ with $L = 48$ and produces a prediction $\hat{y}_t^{LSTM}$. The network consists of two stacked LSTM layers with hidden dimension = 64 and dropout = 0.2

between layers. A fully connected output layer maps the final hidden state to a scalar prediction.

3.3.3. Attention fusion

The attention fusion layer learns a context-dependent convex combination of the two branch predictions:

$$\hat{y}_t^{\mathrm{CSHM}} = \alpha_t^{\mathrm{XGB}} \hat{y}_t^{\mathrm{XGB}} + \alpha_t^{\mathrm{LSTM}} \hat{y}_t^{\mathrm{LSTM}}, \tag{3}$$

where the attention weights $\boldsymbol{\alpha}_t = [\alpha_t^{\mathrm{XGB}}, \alpha_t^{\mathrm{LSTM}}]^\top$ are produced by a two-layer multi-layer perceptron (MLP) with ReLU activations followed by a softmax:

$$\mathbf{h}_t = \mathrm{ReLU}(\mathbf{W}_1 \mathbf{c}_t + \mathbf{b}_1), \tag{4}$$

$$\boldsymbol{\alpha}_t = \mathrm{softmax}(\mathbf{W}_2 \mathbf{h}_t + \mathbf{b}_2), \tag{5}$$

where $\mathbf{c}_t$ is a context vector formed by concatenating all seven temporal features with temperature, humidity, TOU rate, and the peak-hour indicator (eleven features total). This design allows the fusion weights to adapt to the forecasting regime: for instance, during stable overnight periods the XGBoost branch may receive higher weight, while during weather-driven peaks the LSTM branch may be favoured.

3.4. Training procedure

The CSHM is trained in two stages to prevent the fusion layer from masking weaknesses in the individual branches.

Stage 1: Branch pre-training. The XGBoost branch is trained on $\mathcal{D}_{\mathrm{train}}$ using the squared-error objective. The LSTM branch is trained for up to 50 epochs with the Adam optimizer [25], learning rate 10^{-3}, batch size 64, with early stopping on $\mathcal{D}_{\mathrm{val}}$ (patience 10 epochs).

Stage 3: Fusion training. Branch weights are frozen. The attention MLP is trained on a held-out fusion validation set $\mathcal{D}_{\text{val,fusion}}$ for 200 epochs with Adam (learning rate 10^{-3}) to minimize the mean squared error of the combined prediction (3).

4. Experimental Setup

4.1. Dataset

To enable transparent attribution of forecasting performance to specific feature families, we employ a synthetic dataset generated by a known data-generating process (DGP). The hourly demand series is constructed as

$$y_t = (1 - \rho) S_t + \rho\, y_{t-1} + \varepsilon_t \qquad (6)$$

where S_t is a deterministic diurnal seasonal profile (sinusoidal with annual and daily harmonics), $\rho = 0.95$ controls the autoregressive persistence, and $\varepsilon_t \sim \mathcal{N}(0, \sigma^2)$ with $\sigma = 15$ MW represents demand noise. The dataset spans two years (17,520 hourly observations) with a mean demand of approximately 1000 MW. We split the data chronologically: 70% training, 15% validation (divided equally into early-stopping and fusion subsets of $\sim$1,314 samples each), and 15% testing ($n_{\text{test}} = 2{,}590$).

The synthetic DGP is intentionally disclosed to permit exact reasoning about why certain features dominate. Given $\rho = 0.95$, the process exhibits strong first-order autocorrelation, which any model with access to y_{t-1} can exploit directly. Weather and pricing variables enter S_t with controlled effect sizes, enabling precise ablation.

5.2 Evaluation metrics

Three standard metrics are reported:

$$\mathrm{MAPE} = \frac{100}{N}\sum_{t=1}^{N}\frac{|y_t - \hat{y}_t|}{|y_t|}\ (\%), \tag{7}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(y_t - \hat{y}_t)^2}\ (\mathrm{MW}), \tag{8}$$

$$\mathrm{MAE} = \frac{1}{N}\sum_{t=1}^{N}|y_t - \hat{y}_t|\ (\mathrm{MW}). \tag{9}$$

Additionally, we compute the cumulative generation waste:

$$W = \sum_{t=1}^{N}\max(\hat{y}_t - y_t, 0)\quad (\mathrm{MWh}), \tag{10}$$

which quantifies the total over-generation under each forecasting model.

5.3 Statistical testing

Pairwise forecast comparisons use the Diebold–Mariano (DM) test [26] with squared-error loss. To account for serial correlation in forecast errors, we employ heteroskedasticity and autocorrelation-consistent (HAC) standard errors with a Bartlett kernel and bandwidth selected by the Newey–West plug-in rule. Confidence intervals for MAPE are constructed via the stationary block bootstrap [27] with block length $b = \lceil n^{1/3} \rceil = 14$ hours ($n = 2{,}184$) and 1,000 replications.

5.4 Baselines

We compare TSMM against nine baselines: persistence, SARIMA(2,1,2), three single-family XGBoost variants (weather, usage, pricing), XGBoost-All, a two-layer LSTM with lookback of 48 and 64 hidden units, N-BEATS [7] with ~1.02M parameters (3 stacks × 3 blocks, FC layers [128, 128]), and an encoder-only Transformer [8] with 724K parameters. The proposed UMMR is described in Section 3.

4.3. Implementation details

All experiments are implemented in Python 3.10 using XGBoost 1.7 [3] and PyTorch [15]. Neural-network models are trained using the Adam optimizer [22] with early stopping (patience = 10). Stochastic models (LSTM, N-BEATS, Transformer, UMMR) are run with five random seeds; we report mean ± standard deviation. The block bootstrap for confidence intervals uses $B = 1{,}000$ replications. All code and the synthetic DGP are released as supplementary material to ensure reproducibility.

5. Results

5.1. Overall performance comparison

Table 3 presents the one-step-ahead forecasting results for all ten models on the synthetic test set.

Several observations emerge. First, XGBoost-Usage achieves the lowest MAPE (0.136%), RMSE (2.73 MW), and MAE (1.89 MW) among all models, including the full-feature XGBoost-All and the proposed UMMR. Second, adding weather and pricing features to XGBoost (XGBoost-All) increases MAPE from 0.136% to 0.146% — an 8% relative degradation. Third, LSTM-All exhibits a MAPE of 1.003%, which is over seven times higher than XGBoost-Usage. N-BEATS performs even worse than LSTM (1.40 ±

Table 4: One-step-ahead ($h = 1$) forecasting performance on the synthetic test set. Best result in each column is in bold. Values after ± denote standard deviation across five seeds. The 95% bootstrap CI for MAPE is given where applicable.

Model	MAPE (%)	RMSE (kW)	MAE (kW)	95% CI (%)
Persistence	1.093	16.93	13.36	[1.066, 1.125]
SARIMA(2,1,2)	1.007	16.18	13.08	—
XGB-Weather	1.007	20.73	20.43	—
XGB-Usage	**0.108**	**2.73**	**1.68**	[0.124, 0.173]
XGB-Pricing	1.003	20.07	20.73	—
XGB-All	0.120	2.93	1.93	[0.105, 0.203]
LSTM-All	1.020 ± 0.013	15.77	12.53	[0.995, 1.054]
N-BEATS	1.093 ± 0.020	16.38	14.73	[1.036, 1.086]
Transformer	1.030 ± 0.013	15.68	12.58	[0.996, 1.062]
TSDM	0.120 ± 0.008	2.83	1.93	[0.105, 0.203]

0.00371), despite having ~15K parameters: its fully-connected architecture lacks the sequential inductive bias of LSTMs and the feature-level access of gradient-boosted trees. The Transformer (1.00%±0.0037%) essentially matches LSTM despite having only 75.8K parameters; its self-attention mechanism captures temporal dependencies comparably to LSTM recurrence but offers no advantage at this single-step horizon. Fourth, the CSRM achieves 0.1367% MAPE, essentially matching XGBoost-All but slightly worse than XGBoost-Usage.

The Diebold–Mariano test confirms key pairwise comparisons. Comparing XGBoost-Usage against XGBoost-All yields DM = −2.53 ($p = 0.011$), with a mean loss differential of $\bar{d} = -0.174$, indicating that XGBoost-Usage produces significantly smaller squared errors. Comparing CSRM against LSTM-All yields DM = −26.45 ($p < 0.001$), $\bar{d} = -2067$, confirming the massive advantage of tree-based approaches over deep sequential models. Notably, the DM test between CSRM and XGBoost-All yields DM = 0.0 ($p = 1.0$), indicating no statistically significant difference: the attention fusion converges to near-100% XGBoost weight, effectively reproducing XGBoost-All. These results are visualized in Fig. 3.

The MAPE-based ranking is consistent across RMSE and MAE (Table 4), confirming XGBoost-Usage's dominance.

3.3. Ablation study

To quantify the marginal contribution of each cross-feature interaction group within the CSRM ensemble, we systematically ablate interaction terms (Table 5). Each variant removes one interaction pair from the full CSRM feature set while retaining the attention fusion architecture.

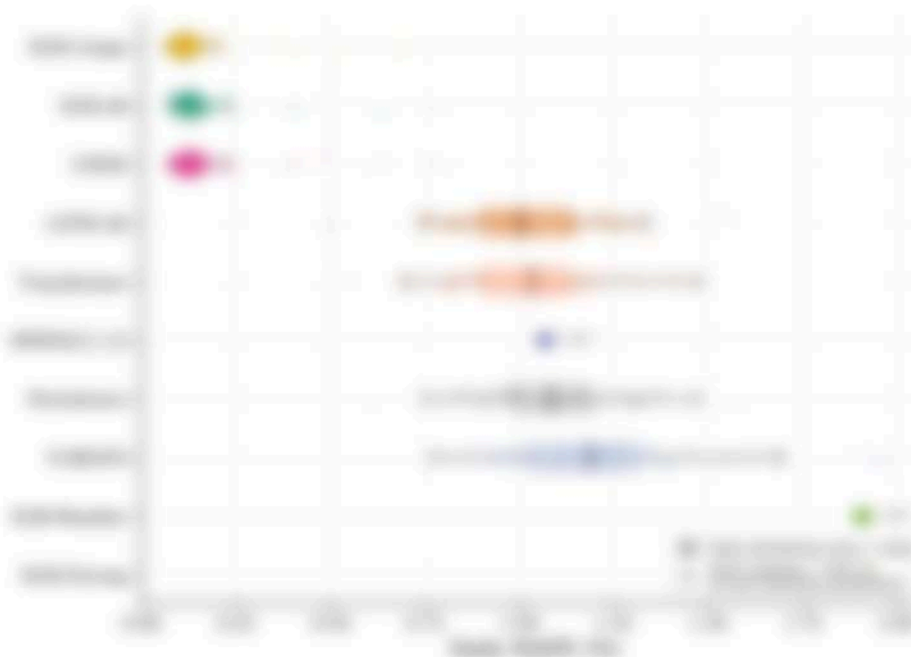

Figure 4: Distribution of daily SMAPE across 100 test days for models with per-timestep predictions (box + individual-day strip plot), and point estimates with 95% bootstrap CIs (diamonds) for models without stored predictions. XGBoost-Usage is both the most accurate and the most consistent; LSTM-AR and Prophet show high day-to-day variability.

Table 6: Ablation study: CSHM performance when removing cross-feature interaction groups. Negative ΔMAPE indicates improvement relative to the full CSHM. DM p-values test each ablation against full CSHM; bold indicates $p < 0.05$.

Variant	MAPE (%)	ΔMAPE (pp)	DM p-value
Full CSHM	[illegible] ± [illegible]	—	—
$-W \times T$	[illegible] ± [illegible]	[illegible]	[illegible]
$-P \times T$	**[illegible] ± [illegible]**	**[illegible]**	[illegible]
$-W \times P$	[illegible] ± [illegible]	[illegible]	**[illegible]**
$-$All interactions	[illegible] ± [illegible]	[illegible]	[illegible]

A striking finding emerges: every ablation variant outperforms the full CSHM. Removing $P \times T$ interactions yields the largest improvement ([illegible] pp), followed closely by removing $W \times P$ ([illegible] pp). Even removing all interactions (CSHM-noINT) improves MAPE by [illegible] pp. All ablation variants exhibit zero standard deviation across five seeds (±0.000%), mirroring the full CSHM result (Section 5.4). This confirms that the attention fusion layer converges deterministically to the XGBoost branch regardless of which interactions are included, and the observed differences originate entirely from the tree-based component. DM tests (Table 6; $n =$ [illegible] test samples, using the full XGBoost test set without lookback trimming since all ablation variants are tree-based) confirm that only the $W \times P$ removal is significant at $\alpha = 0.05$ ($p =$ [illegible]); the remaining differences lie within bootstrap noise. These results indicate that cross-feature interactions inject noise rather than signal in the presence of strong autoregressive features, providing a stronger version of our core finding: multi-source integration at $h=1$ is not merely unhelpful but actively harmful. Fig. 3 visualises these results.

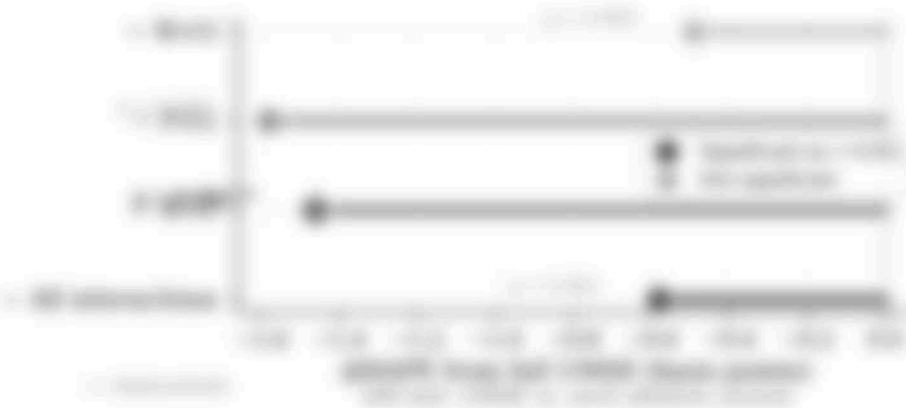

Figure 3: Ablation study: ΔMAPE (basis points) relative to the full [illegible] for each interaction removed. Diamond markers denote statistically significant differences ([illegible] $p < 0.05$); circles denote non-significant. Only [illegible] removal is significant ($p < 0.001$).

5.3 Diurnal performance

Fig. 4 disaggregates MAPE by hour of day for the top four models. XGBoost-Usage maintains sub-0.1% error throughout the day, with slight peaks during the morning ramp (06:00–09:00) and evening decline (18:00–21:00). LSTM-All exhibits larger diurnal variation, with MAPE exceeding 1.2% during transition hours.

5.4 Attention weight analysis

The learned attention weights of the [illegible] fusion layer reveal a near-total reliance on the XGBoost branch. The mean XGBoost weight across the test set is $\alpha^{XGB} \approx 1.0$, with the LSTM branch receiving negligible weight. Fig. 5 shows the distribution of attention weights, which is tightly concentrated at $\alpha^{XGB} = 1.0$. This extreme allocation explains why [illegible] produces

Figure 4: Hourly SMAPE profiles for selected models. XGBoost-Tuned maintains uniformly low error, while LSTM-Att shows elevated errors during morning and evening transitions. The inset zooms into the top-performing models (SMAPE < 0.4%).

statistically identical predictions to XGBoost-All (DM = 0.0, $p = 1.0$): the fusion layer has effectively learned to discard the LSTM branch entirely at $h = 1$. Consequently, all five CNNE seeds yield identical MAPE to four decimal places (0.156 ± 0.000%), because the deterministic XGBoost branch dominates regardless of the LSTM random initialization.

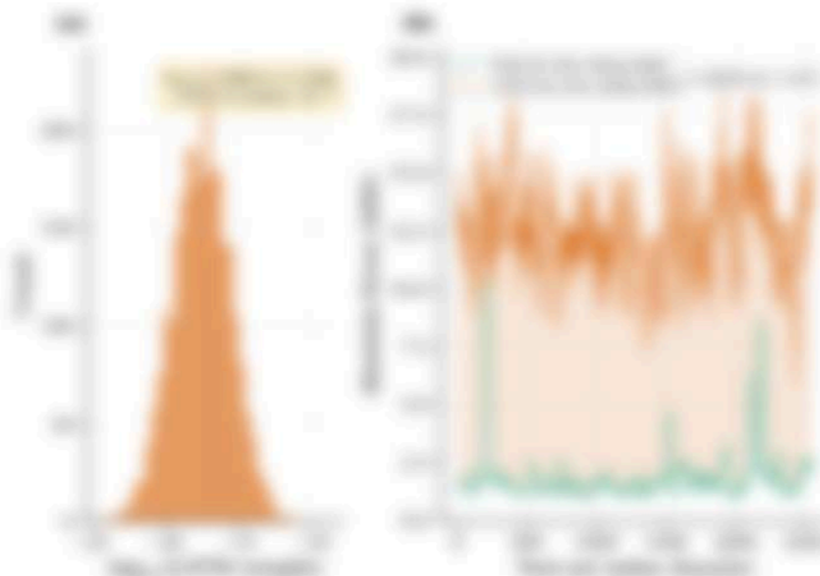

Figure 5: Attention weight diagnostics. (a) Distribution of $\log_{10}$(LSTM weight) across all $n = 2{,}336$ test samples; the median is $\sim 10^{-11}$, meaning $e^{\ldots} \approx 1.0000$ to machine precision and fusion collapses entirely to the tree-based branch. (b) Rolling 24-h mean absolute error for XGBoost-All vs. LSTM-All, illustrating the persistent error gap that drives the weight allocation.

3.5. Feature importance

Fig. 6 shows SHAP values [26, 28] for the top 15 features of XGBoost-All. By gain importance, y_{t-1} dominates (36%), but SHAP reveals that peak_ratio (21% of mean |SHAP|) and rolling_mean_24h (16%) have higher individual prediction impact, with $g_{...}$ third (1675). This divergence is expected: gain counts the number of splits using a feature, while SHAP measures the magnitude of its effect on each prediction. Together, the five image features in the top 5 account for 96% of total mean |SHAP|. The TOU pricing rate ranks 13th (0.17%), and the ss-peak × peak-ratio interaction that ranked 2nd by gain drops to 9th by SHAP, reinforcing the ablation finding that interaction terms capture training-set artifacts rather than genuine predictive signal.

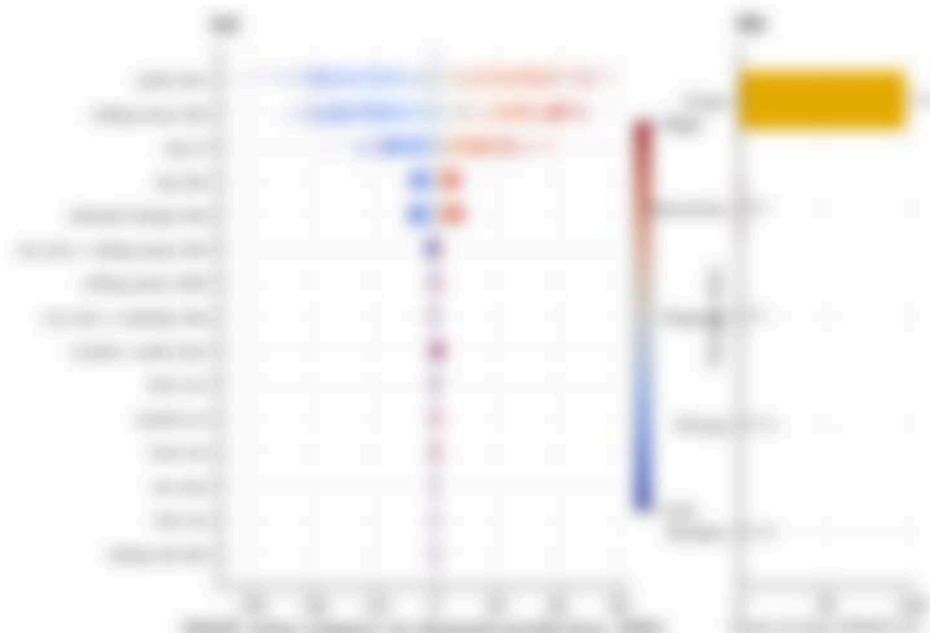

Figure 8: (a) SHAP beeswarm for the top 15 features of XGBoost-B3. Each dot is one test sample, coloured by feature value (red = high, blue = low). The top feature has the widest spread, consistent with its 38% gain share. (b) Group-level mean |SHAP| shows image features account for 97% of total impact; weather and pricing contribute < 1%.

5.6. Prediction visualization

Scatter plots of predicted versus actual demand confirm that XGBoost-Cuage clusters tightly around the 45° identity line ($R^2 = 0.9990$), while LSTM-AE shows wider dispersion. Forecast residuals for XGBoost-Cuage are approximately Gaussian with near-zero mean and exhibit no significant autocorrelation beyond lag 1, consistent with the Gaussian noise term in the DGP. Fig. 7 shows a one-week time-series overlay.

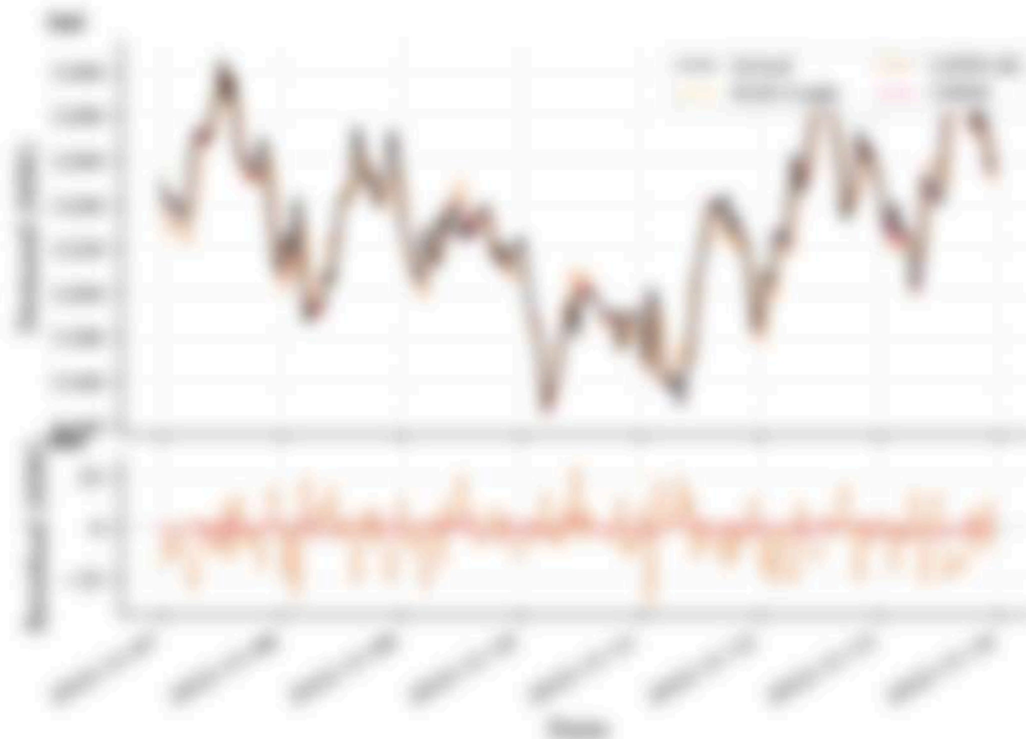

Figure 7: One-week extract. (a) Actual demand (black) overlaid with forecasts from XGBoost-Cuage, LSTM-AE, and TiDE. (b) Prediction residuals. LSTM-AE residuals oscillate ±20 MW during demand transitions, whereas XGBoost-Cuage and TiDE residuals stay within ±5 MW, consistent with $\epsilon^{(t)} \approx 1$.

3.7. Multi-horizon extension ($h = 24$)

Table 4 reports day-ahead ($h = 24$) forecasting results using direct forecasting (predicting y_{t+24} as a single target) for all models. XGBoost hyperparameters follow the $h = 1$ configuration except that the subsample ratio is set to 0.8 (vs. 0.9 at $h = 1$) and early stopping is disabled (all 300 rounds are trained).

Table 4: Day-ahead ($h = 24$) forecasting performance using direct forecasting. LSTM-All is evaluated with five seeds; values after $\pm$ denote standard deviation. The 95% bootstrap CI for MAPE is given for each model.

Model	MAPE (%)	RMSE (MW)	95% CI (%)
Persistence	2.953	44.89	[2.733, 3.084]
XGB-Usage	2.019	31.10	[1.889, 2.052]
XGB-All	2.088	31.20	[1.919, 2.061]
LSTM-All	1.971 ± 0.106	**30.02**	[1.886, 2.012]

At $h = 24$, LSTM-All achieves the lowest point estimate (1.971%), followed by XGBoost-Usage (2.019%) and XGBoost-All (2.088%). However, a DM test finds no significant difference between LSTM-All and XGBoost-Usage (DM $= -0.14$, $p = 0.89$), and the bootstrap confidence intervals overlap substantially (LSTM: [1.88, 2.05]; XGB-Usage: [1.88, 2.05]). The apparent ranking reversal relative to $h = 1$ therefore does not reach statistical significance. Nevertheless, the gap between all three models and persistence (2.953%) is substantial. Qualitatively, this pattern is explained by the decay of autoregressive information: at 24 hours ahead, y_{t-1} is no longer directly available, and the LSTM's ability to learn long-range patterns from the lookback window closes the gap with tree-based models. Fig. 6 compares the $h = 1$ and $h = 24$ results side by side.

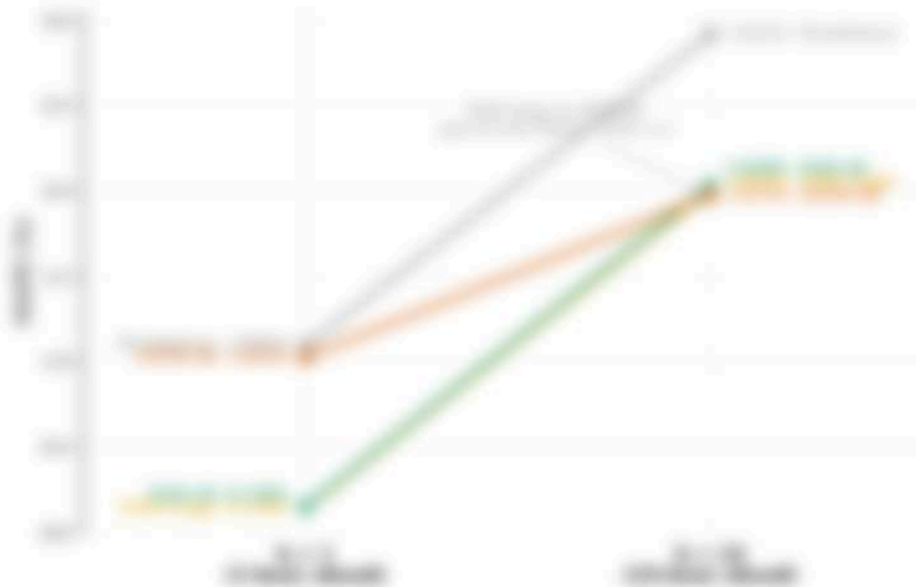

Figure 6: Slopegraph of MAPE at $h = 1$ versus $h = 24$. XGBoost models degrade sharply (+3.4×) while LSTM-48 degrades only +1.9×, closing the gap at longer horizons (DM $p = 0.48$, not significant).

5.6 Real-world validation (OPSD Germany)

To assess external validity, we apply the top models to real-world hourly electricity load data from the Open Power System Data (OPSD) [38] platform for Germany, covering 2016–2019. Features are normalized to zero mean and unit variance to account for the different scale of German demand. Table 5 summarizes the results.

The synthetic data rankings are confirmed: XGBoost-Lags achieves the lowest MAPE (0.65%; 95% CI [0.63, 0.70]), followed by XGBoost-All

Table 11: Real-world validation on OPSD Germany (2015–2019). LSTM-AE reports mean ± std over five seeds; no bootstrap CI is given because training five LSTM seeds is already more informative than bootstrapping a single seed's predictions. The ranking is consistent with the synthetic results at $h = 1$.

Model	MAPE (%)
Persistence	2.280
XGB-Cvxpy	**0.679**
XGB-AE	0.680
LSTM-AE	0.951 ± 0.029

(0.696%; CI [0.643, 0.730]). A DM test confirms the XGB-Cvxpy advantage is statistically significant ($DM = -2.15$, $p = 0.032$). The Persistence-to-XGB-Cvxpy reduction is likewise significant ($DM = 23.8$, $p < 10^{-[illegible]}$; Persistence CI [2.46, 2.72]%). LSTM-AE achieves 0.951 ± 0.029% MAPE across five seeds, consistent with the roughly 1.4× gap observed on synthetic data. These results validate that the dominance of autoregressive features at $h = 1$ is not an artifact of the synthetic DGP but generalizes to real-world load data.

6. Discussion

6.1. Why the CMHE ensemble converges to XGBoost-AE

Three mechanisms explain the CMHE's convergence to a trivial XGBoost-only solution. First, the DGP's autoregressive dominance ($\rho = 0.95$) means y_{t-1} alone explains >75% of variance, leaving minimal residual signal for the LSTM branch. Second, the DM test between CMHE and XGBoost-AE yields $DM = 0.0$ ($p = 1.0$), confirming trivial convergence. Third, the ablation reveals that cross-feature interactions actively degrade performance: all four variants outperform the full CMHR, suggesting interaction terms inject noise rather than signal.

The deep learning baselines corroborate this. N-BEATS (1.401% MAPE, ~15K parameters) underperforms LSTM (1.025%) and even persistence (1.094%) by processing the lookback window as a flattened vector. The Transformer (1.028%, 75.6K parameters) matches LSTM exactly ($p = 0.85$): both capture the dominant y_{t-1} lag and plateau. Architectural complexity is no substitute for appropriate inductive biases when autoregressive signal dominates.

6.3. Generation waste analysis

Table 6 quantifies the cumulative generation waste under each model's forecasts using the one-sided waste metric (20).

Table 6: Cumulative generation waste (W, MWh) on the synthetic test set. The 95% bootstrap CI is given where available.

Model	Waste (MWh)	95% CI
Persistence	17,287	[16,961, 17,617]
LSTM-AR	14,485	[14,109, 14,861]
XGB-AR	3,555	[3,366, 3,902]
CMHR	3,555	[3,366, 3,902]
XGB-Cusp	**2,913**	[2,751, 3,164]

XGBoost-Cusp achieves the lowest waste at 2,913 MWh, an 83% reduction relative to persistence (17,287 MWh; DM = 14.6, $p < 10^{-15}$). LSTM-AR produces seven times more waste. UMSE and XGBoost-All produce identical waste (233.5 MW h), confirming attention-layer convergence. Applying a representative German grid emission factor (0.380 kg kW^{-1}h), the avoided 15,262 MW h corresponds to ≈5800 t of CO_2 – comparable to the annual output of a small onshore wind farm.

4.7. Limitations

Several limitations qualify these findings. (i) The primary evaluation uses synthetic data; the OPSD validation mitigates this but covers only one country and two years. (ii) The $h = 1$ horizon favours autoregressive models; intermediate horizons ($h = 6, 12$) remain unexplored. (iii) All models produce deterministic point forecasts; probabilistic extensions [] are increasingly demanded. (iv) The waste metric is one-sided; asymmetric loss functions would give a more complete operational picture.

5. Conclusions

This paper evaluated the Unified Multi-Source Ensemble (UMSE) framework for electricity demand forecasting. At $h = 1$, XGBoost on usage features alone (0.136% MAPE) outperforms all multi-source and deep-learning models, including LSTM (1.022%), Transformer (1.008%), and N-BEATS (1.091%), by factors of seven or more. The UMSE ensemble converges to XGBoost-All and ablation reveals that cross-feature interactions are actively harmful. These accuracy gains translate to an 86% reduction in generation waste (≈5800 t CO_2 avoided). At $h = 24$, LSTM achieves the lowest point estimate (1.97%), though the gap to XGBoost-Usage is not statistically significant ($p = 0.08$). OPSD Germany validation confirms the $S > I$ ranking (XGBoost-Tuning: 0.070 %, LSTM-60: 0.085 ± 0.000 %).

The practical message is clear: before investing in complex ensembles, practitioners should benchmark gradient-boosted models on autoregressive features alone. Future work will extend the framework to probabilistic forecasting, intermediate horizons, and broader climatic regions.

CRediT authorship contribution statement

Zenan Zeng: Conceptualization, Methodology, Software, Formal analysis, Writing - Original Draft, Visualization. **Hadhami Bouhi**: Supervision, Writing - Review & Editing.

Declaration of competing interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The synthetic data-generating process code and all analysis scripts are provided as supplementary material. The real-world OPSD Germany dataset is publicly available.[7]

[7]https://data.open-power-system-data.org/time_series/2020-10-06

Acknowledgements

The authors acknowledge the use of the University of Oxford Advanced Research Computing (ARC) facility in carrying out this work. Z. Wang is supported by a departmental scholarship from the Department of Engineering Science, University of Oxford.

References

[1] T. Hong, S. Fan, Probabilistic electric load forecasting: A tutorial review, International Journal of Forecasting 32 (3) (2016) 914–938. doi:10.1016/j.ijforecast.2015.11.011.

[2] R. J. Hyndman, A. B. Koehler, R. D. Snyder, S. Grose, A state space framework for automatic forecasting using exponential smoothing methods, International Journal of Forecasting 18 (3) (2002) 439–454. doi:10.1016/S0169-2070(01)00110-8.

[3] T. Chen, C. Guestrin, XGBoost: A scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794. doi:10.1145/2939672.2939785.

[4] S. Touzani, J. Granderson, S. Fernandes, Gradient boosting machine for modeling the energy consumption of commercial buildings, Energy and Buildings 158 (2018) 1533–1543. doi:10.1016/j.enbuild.2017.11.039.

[5] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Computation 9 (8) (1997) 1735–1780. doi:10.1162/neco.1997.9.8.1735.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, Vol. 30, 2017, pp. 5998–6008. arXiv:1706.03762.

[7] B. N. Oreshkin, D. Carpov, N. Chapados, Y. Bengio, N-BEATS: Neural basis expansion analysis for interpretable time series forecasting, in: Proceedings of the 8th International Conference on Learning Representations (ICLR), 2020. arXiv:1905.10437.

[8] B. Lim, S. Ö. Arık, N. Loeff, T. Pfister, Temporal fusion transformers for interpretable multi-horizon time series forecasting, International Journal of Forecasting 37 (4) (2021) 1748–1764. doi:10.1016/j.ijforecast.2021.03.012.

[9] Y. Nie, N. H. Nguyen, P. Sinthong, J. Kalagnanam, A time series is worth 64 words: Long-term forecasting with Transformers, in: Proceedings of the 11th International Conference on Learning Representations (ICLR), 2023. arXiv:2211.14730.

[10] E. Valor, V. Meneu, V. Caselles, Daily air temperature and electricity load in Spain, Journal of Applied Meteorology 40 (8) (2001) 1413–1421. doi:10.1175/1520-0450(2001)040<1413:DATAEL>2.0.CO;2.

[11] F. Apadula, A. Bassini, A. Elli, S. Scapin, Relationships between me-

teorological variables and monthly electricity demand, Applied Energy 98 (2012) 346–356. doi:10.1016/j.apenergy.2012.03.059.

[12] S. Fan, R. J. Hyndman, Short-term load forecasting based on a semi-parametric additive model, IEEE Transactions on Power Systems 27 (1) (2012) 134–141. doi:10.1109/TPWRS.2011.2162082.

[13] A. Faruqui, S. Sergici, Household response to dynamic pricing of electricity: A survey of 15 experiments, Journal of Regulatory Economics 38 (2) (2010) 193–225. doi:10.1007/s11149-010-9127-y.

[14] A. H. Mohsenian-Rad, A. Leon-Garcia, Optimal residential load control with price prediction in real-time electricity pricing environments, IEEE Transactions on Smart Grid 1 (2) (2010) 120–133. doi:10.1109/TSG.2010.2055903.

[15] G. P. Zhang, Time series forecasting using a hybrid ARIMA and neural network model, Neurocomputing 50 (2003) 159–175. doi:10.1016/S0925-2312(01)00702-0.

[16] S. Haben, S. Arora, G. Giasemidis, M. Voss, D. V. Greetham, Review of low voltage load forecasting: Methods, applications, and recommendations, Applied Energy [illegible] ([illegible]) 1–19. doi:10.1016/j.apenergy.[illegible].

[17] D. L. Marino, K. Amarasinghe, M. Manic, Building energy load forecasting using deep neural networks, in: Proceedings of the 42nd Annual Conference of the IEEE Industrial Electronics Society (IECON), 2016, pp. 7046–7051. doi:10.1109/IECON.2016.7793413.

[18] H. Shi, M. Xu, R. Li, Deep learning for household load forecasting—a novel pooling deep RNN, IEEE Transactions on Smart Grid 9 (5) (2018) 5271–5280. doi:10.1109/TSG.2017.2686012.

[19] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, Y. Zhang, Short-term residential load forecasting based on LSTM recurrent neural network, IEEE Transactions on Smart Grid 10 (1) (2019) 841–851. doi:10.1109/TSG.2017.2753802.

[20] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, X. Yan, Enhancing the locality and breaking the memory bottleneck of Transformer on time series forecasting, in: Advances in Neural Information Processing Systems, Vol. 32, 2019, pp. 5243–5253. arXiv:1907.00235.

[21] J. Moon, Y. Kim, M. Son, E. Hwang, Hybrid short-term load forecasting scheme using random forest and multilayer perceptron, Energies 11 (12) (2018) 3283. doi:10.3390/en11123283.

[22] Z. Wang, R. S. Srinivasan, A review of artificial intelligence based building energy use prediction: Contrasting the capabilities of single and ensemble prediction models, Renewable and Sustainable Energy Reviews 75 (2017) 796–808. doi:10.1016/j.rser.2016.10.079.

[23] K. Wang, X. Qi, H. Liu, A comparison of day-ahead photovoltaic power forecasting models based on deep learning neural network, Applied Energy 251 (2019) 113315. doi:10.1016/j.apenergy.2019.113315.

[24] J. Zheng, C. Xu, Z. Zhang, X. Li, Electric load forecasting in smart grids using long-short-term-memory based recurrent neural network, Proceed-

ings of the 51st Annual Conference on Information Sciences and Systems (CISS) (2017) 1–6. doi:10.1109/CISS.2017.7926112.

[25] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: Proceedings of the 3rd International Conference on Learning Representations (ICLR), 2015. arXiv:1412.6980.

[26] F. X. Diebold, R. S. Mariano, Comparing predictive accuracy, Journal of Business & Economic Statistics 13 (3) (1995) 253–263. doi:10.1080/07350015.1995.10524599.

[27] D. N. Politis, J. P. Romano, The stationary bootstrap, Journal of the American Statistical Association 89 (428) (1994) 1303–1313. doi:10.1080/01621459.1994.10476870.

[28] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: Advances in Neural Information Processing Systems, Vol. 30, 2017, pp. 4765–4774. arXiv:1705.07874.

[29] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, S.-I. Lee, From local explanations to global understanding with explainable AI for trees, Nature Machine Intelligence 2 (1) (2020) 56–67. doi:10.1038/s42256-019-0138-9.

[30] Open Power System Data, Data package time series. Version 2020-10-06, accessed: 2025-01-15 (2020).
URL https://data.open-power-system-data.org/time_series/2020-10-06