IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Traitement du Signal
Vol. 43, No. 2, April, 2026, pp. 1-12
Journal homepage: http://iieta.org/journals/ts

A Leakage-Free Reappraisal of Decomposition-Ensemble Electricity-Load Forecasting

[AUTHORS TBD][1]*

[1] *[Affiliation TBD]*

Corresponding Author Email: [CORRESPONDING EMAIL TBD]

https://doi.org/10.18280/ts.XXXXXX

Received:
Revised:
Accepted:
Available online:

ABSTRACT

Short-term electricity-load forecasting guides unit commitment, demand response, and grid-security decisions, yet the hourly demand series is non-stationary and multi-scale. The field's dominant answer is the decomposition-ensemble pipeline, which splits the series into modes, predicts each, and sums them, and which reports large accuracy gains. We show that these gains are, to a substantial degree, a measurement artifact. The decomposition is usually computed once over the entire record before the train-test split, so every training mode already carries information from the test horizon, and offline accuracy no longer reflects deployable skill. We reframed decomposition as a causal operator recomputed inside a sliding window that never sees beyond the forecast origin, and evaluated a compact attention-gated-recurrent forecaster against statistical and neural baselines on the open UCI Individual Household Electric Power Consumption record under a strictly chronological split. Under this honest protocol decomposition did not help: a linear AR(24) model on the raw series was the most accurate at 0.613 RMSE, while the causal decomposition models reached only 0.674 and 0.697. A dedicated leakage audit shows why the literature reports the opposite: running the same decomposition in its conventional whole-series form cut the error from 0.674 to 0.265 RMSE, an inflation of 0.41 that is borrowed entirely from the future. We release the causal protocol so decomposition-based results can be reported honestly.

Keywords: *electricity load forecasting, signal decomposition, data leakage, leakage-free evaluation, gated recurrent unit, VMD*

1. INTRODUCTION

Short-term electricity-load forecasting is a backbone of power-system operation. Hour-ahead and day-ahead demand forecasts drive unit commitment, economic dispatch, reserve sizing, and demand-response and grid-security decisions, so a small percentage error translates into large operating cost and reliability risk. The raw load series is awkward to forecast directly because it is both non-stationary and multi-scale: slow seasonal and weather-driven demand trends ride underneath strong diurnal and weekly cycles and noisy consumption spikes, and a single recurrent network fed the raw signal must reconcile all of these at once. Neural networks have a long track record on this task and motivated the move to deep recurrent forecasters [1], and machine-learning and deep-learning models are now the standard tools for short-term load forecasting across a large body of work [2].

The dominant answer to the multi-scale difficulty is the decomposition-ensemble pipeline, which separates scales before it learns. A load series is decomposed into a small set of modes with a signal-processing operator such as variational mode decomposition or complete ensemble empirical mode decomposition with adaptive noise, each mode is forecast by its own recurrent network, and the per-mode forecasts are summed back into a load estimate. This recipe is reported to outperform single-model predictors: VMD with a gated recurrent network reports strong short-term load accuracy [3], CEEMDAN-based hybrids improve ultra-short-term load forecasting [4], and VMD-LSTM forecasters report higher accuracy for grid-security applications [5]. The bottleneck, however, is no longer whether to decompose but whether the decomposition is performed honestly. These operators are global: a mode value at time t is a function of the whole input series, including load samples that lie in the future relative to t. When the series is decomposed once and only then split into train and test partitions, the training modes are already contaminated by the test horizon, and the model is rewarded for reading the future rather than forecasting it. Recent leakage audits confirm that decomposing before the split inflates reported accuracy and that the inflation can be large [6, 7], a pattern consistent with the broader finding that data leakage is a pervasive and easily overlooked source of optimistic results [8], and with the long-standing caution that any procedure letting information cross the forecast origin breaks the temporal order the task depends on [9].

We reframe decomposition-based load forecasting so that the decomposition becomes a causal feature operator that is part of the model rather than a one-shot preprocessing convenience applied to the whole series. At each forecast origin the load history inside a fixed look-back window, and only that window, is decomposed into a small set of modes, so the representation is exactly what would be available at deployment. A compact

gated-recurrent encoder reads the stack of causal load modes, and a scale-attention head weights the modes by their forecast-relevant energy instead of treating them as an undifferentiated sum, so the slow demand trend and the volatile high-frequency residual are reconciled by the model rather than by naive addition. A lightweight residual stage then predicts the reconstruction error that per-mode forecasting leaves behind and adds it back at inference, correcting systematic bias. We call the resulting model CLEAR.

We evaluated CLEAR on the open UCI Individual Household Electric Power Consumption record under a strictly chronological split, comparing against statistical and neural baselines on root mean squared error, mean absolute error, mean absolute percentage error, and the coefficient of determination. The statistical references include a linear autoregressive model of order 24, denoted AR(24), the kind of linear baseline that anchors short-term load evaluation [10]. The result was not the one the decomposition-ensemble literature would predict. Under the honest causal protocol decomposition did not help: the linear AR(24) model on the raw series was the most accurate at 0.613 RMSE (0.563 R^2), with the plain LSTM (0.619) and GRU (0.622) next, while the causal decomposition models trailed at 0.674 (CLEAR) and 0.697 (VMD-GRU) and persistence sat at 0.709. A dedicated leakage audit then showed why so many studies report the opposite, since running the same decomposition in its conventional whole-series form cut the error from 0.674 to 0.265 RMSE and raised R^2 from 0.471 to 0.918, an inflation of 0.41 RMSE borrowed entirely from the future. We therefore present this work less as a claim of a new state of the art than as a measured, leakage-aware correction to how decomposition-based load results should be reported, with the main comparison and the audit in Section 4.

Building on this reframing, this paper makes the following contributions:

- We reframe decomposition-based electricity-load forecasting from a whole-series preprocessing step into a causal, leakage-free operator recomputed inside an origin-respecting sliding window, so that, unlike the prevailing VMD- and CEEMDAN-based ensembles that decompose ahead of the split [3, 4], no load sample beyond the forecast origin contributes to any mode and reported accuracy reflects deployable skill.

- We recast the decomposed load modes from a naively summed bag into an attention-weighted multi-scale representation read by a compact gated-recurrent encoder and reconciled by a lightweight residual error-correction stage, rather than the fixed summation that whole-series VMD-LSTM ensembles rely on [5].

- We report the honest empirical finding that, under leakage-free causal evaluation, neither a causal decomposition ensemble nor the compact attention-gated-recurrent CLEAR forecaster beat a linear AR(24) model or a plain recurrent network on the raw load series, and we quantify with a dedicated leakage audit how much apparent accuracy the conventional whole-series decomposition borrows from the future: an inflation of 0.41 RMSE, with the coefficient of determination rising from 0.471 to 0.918 [6, 8]. For this VMD-based setup on this household record these gains are substantially attributable to leakage rather than to the representation; we measure the inflation only for the operator we run, and we release the causal protocol so future results can be reported honestly.

2. RELATED WORKS

2.1 Decomposition-ensemble forecasting of load and power

The decomposition-ensemble template is the standard answer to non-stationary demand. A load series is split into modes with an adaptive operator, each mode is predicted by its own deep network, and the per-mode predictions are recombined. Empirical mode decomposition with per-component LSTM forecasting improves short-term load prediction over a single recurrent model [11], and EMD-based ensemble deep learning lowers load-demand error more broadly [12]. The same recipe is now built on the complete ensemble variant: CEEMDAN feeding a TCN-LSTM ensemble decomposes the load and predicts each component [13], a CEEMDAN-MVO-GRU hybrid tunes a gated recurrent predictor over the modes [14], a CEEMDAN-FE-BiGRU-Attention model applies attention over a bidirectional recurrent predictor [15], a Prophet-CEEMDAN-ARBiLSTM pipeline couples trend modelling with the decomposition [16], and a CEEMDAN-TCN-ESN hybrid models each component with convolutional and echo-state networks [17]. The variational branch is equally active: a VMD-GRU-TCN hybrid predicts the variational load modes with a gated recurrent convolutional network [18], a modal decomposition feeding a gated recurrent unit reduces frequency aliasing [19], VMD with a deep kernel extreme learning machine improves day-ahead load forecasting [20], and VMD with deep networks improves multi-horizon residential load forecasting [21]. An EMD-BiLSTM approach decomposes the load for smart-grid forecasting in the same spirit [22]. These systems report strong accuracy, but almost all of them decompose the entire record before partitioning it, so the modes used for training already encode the test horizon, and the headline numbers measure how well a model reads information it would never have at deployment, which is the gap our causal operator closes.

2.2 Causal decomposition, secondary decomposition, and error correction

A second line treats the decomposition as a signal-processing object and refines what it feeds the predictor. Sliding-window EMD performs real-time, causal decomposition within a moving window rather than over the whole record [23], and a sliding-window decomposition forecaster shows that recomputing the operator causally restores honest accuracy where the whole-series form had inflated it [24]. The same decomposition machinery has been pushed toward heavier preprocessing for energy series: an EEMD-WOA-LSTM combination optimizes a recurrent predictor over the modes [25], an SSA-VMD-LSTM secondary-decomposition hybrid stacks two operators for wind power [26], multivariate VMD with LSTM forecasts wind power [27], VMD with LSTM forecasts photovoltaic power [28], and a CEEMDAN-WT-VMD denoising pipeline feeds a BiTCN-BiGRU-attention predictor [29], illustrating how the template carries across the energy domain. Error correction over decomposed forecasts is a recurring refinement: a decomposition-based error-correction strategy improves multi-step load forecasting [30], a multi-trait-driven secondary decomposition improves short-term load [31], and a multi-stage decomposition-and-error-factor ensemble adds a learned correction term [32]. CEEMDAN with a support vector machine and with support vector regression also form decomposition-ensemble baselines on the load and

peak-load tasks [33, 34]. These methods sharpen the decomposition or the recombination, but the causal variants are studied in isolation or on non-load series, and the secondary-decomposition and error-correction work still decomposes the whole record; none wires a per-origin causal operator into a load model and quantifies the inflation, which is what we do.

2.3 Attention and recurrent multi-scale load forecasters

The predictors used above descend from attention-augmented recurrent networks. Attention-augmented LSTM weights informative time steps to improve short-term load forecasting [35], a dilated-LSTM network with an attention mechanism improves electricity-load forecasting [36], and an attention-based CNN-LSTM-BiLSTM model improves short-term load forecasting in integrated energy systems [37]. These designs let a forecaster weight which past hours and which input features matter most, and they consistently report gains over their unattended counterparts. When a decomposition is present, however, the attention still ranges over time steps or features, while the decomposed load modes, if used at all, are treated as a fixed bag to be summed rather than as scales to be weighted by their forecast relevance. The distinction matters because the informative scale shifts from origin to origin: an overnight trough is governed by the slow demand trend, while an evening ramp is announced by the high-frequency modes, and a fixed sum cannot follow that shift. We therefore make the modes the attention's object, so the head can emphasize the scale that carries the forecast at the current origin.

The closest single work to ours is the sliding-window decomposition forecaster that mitigates leakage on a price series [24]: it shares the causal-window insight but neither weights the load modes by forecast relevance nor reports the leaky-versus-causal gap as a controlled audit on electricity load. Read together, the three lines leave a specific construct unfilled. The first builds accurate but leak-prone decomposition load forecasters; the second shows that a causal window can remove the leak, yet only as a signal-processing object or on a non-load series, and adds error correction over still-leaky modes; the third weights time steps and features but never the modes themselves. We bring the causal operator, an attention-weighted multi-scale representation, and a residual-correction stage into one compact load forecaster, and we measure the inflation that the conventional whole-series decomposition would have produced. The next section formalizes this construction.

3. METHODOLOGY

We present CLEAR, a causal leakage-free electricity-load forecaster with attention-weighted multi-scale reconstruction. The construction has four parts: a causal decomposition operator recomputed at every forecast origin, a shared GRU encoder over the resulting load modes, a scale-attention head that weights the modes by their forecast-relevant energy, and a residual error-correction stage. Figure 1 gives the overall data flow. The principle that organizes every part is that nothing beyond the forecast origin may influence the representation, so that the accuracy we report reflects what the model would deliver when it is run hour by hour in a control room.

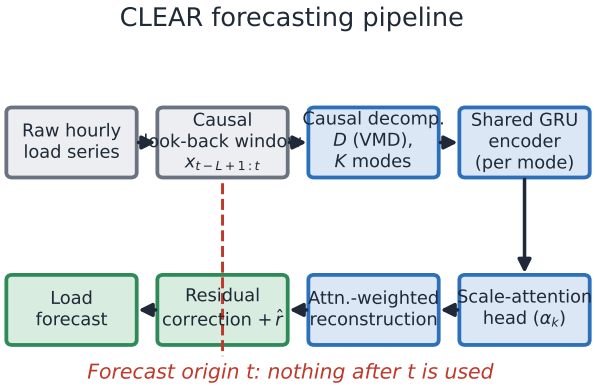

Figure 1. Overview of CLEAR: per-origin causal decomposition of the load window, the shared GRU encoder over the load modes, the scale-attention reconstruction head, and the residual error-correction stage.

3.1 Problem Formulation

Let x_t denote the hourly Global_active_power, our electricity-load target, at time t. The reported experiments are univariate, reading only the load series; the formulation admits an optional calendar-covariate channel z_t (hour-of-day, day-of-week) that we do not use here and leave to future work. We fix a single chronological partition of the record into training, validation, and test segments with no shuffling, so that every test timestamp is strictly later than every training timestamp. Given a look-back window of length L, the task at origin t is to predict the load H steps ahead from the history available up to and including t. The origin-respecting constraint states that the prediction for origin t may depend only on $\{x_s : s \le t\}$ (and on calendar covariates up to t, were they used). This constraint is trivial for a model fed the raw series, but it is exactly what the conventional decomposition-ensemble pipeline violates: a global operator applied once to the full record makes each mode value a function of the entire series, so a training-time mode at $s \le t$ silently embeds future load samples. We therefore treat the decomposition itself as a function of the window and require it to honor the same constraint as the predictor. Calendar covariates, were they used, would be admissible because they are known in advance, and the targets at training origins are admissible because training is offline; what is forbidden is any feature whose value at a training origin was computed using load samples later than that origin, which is the property a whole-series decomposition lacks. The constraint thus partitions the construction cleanly: every quantity the model reads at origin t, the modes, the hidden states, the attention weights, and the residual input, must be a function of the window X_w alone. The rest of this section builds each quantity so that the partition holds by construction rather than by a later check.

3.2 Notation

The symbols used throughout the method are collected in Table 1; each is defined at its first use in the text.

Table 1. Notation.

Symbol	Meaning
x_t	raw hourly Global_active_power (load) at time t
z_t	optional calendar covariates (hour-of-day, day-of-week); not used in the reported univariate experiments
L	look-back window length (past hours)
H	forecast horizon (steps ahead)
K	number of decomposed modes
X_w	look-back window $x[t-L+1..t]$ at origin t
c_k	k-th causal intrinsic mode of the window
h_k	GRU hidden state for mode k
α_k	scale-attention weight for mode k
$\hat{y}$	attention-reconstructed point load forecast
$\hat{r}$	predicted reconstruction residual
$\hat{y}^c$	corrected load forecast
y_t	ground-truth electricity-load target

3.3 Causal Decomposition Operator

At each origin t we form the window $X_w = x[t-L+1..t]$ and decompose only that window into K modes,

$$\{c_1, \ldots, c_K\} = \mathscr{D}(X_w), \tag{1}$$

where $\mathscr{D}$ is the decomposition operator, instantiated as VMD in our experiments with CEEMDAN an interchangeable alternative behind the same interface [38, 39]. The operator is applied independently at every origin, so the modes for origin t are computed from X_w alone and no load sample with index greater than t can contribute. The modes are ordered from the fastest oscillatory component to the slowest demand trend, and they reconstruct the window up to the operator's own residual, $\sum_{k=1}^{K} c_k \approx X_w$. Each mode c_k is a length-L sub-signal that the encoder reads as a load series in its own right. The decomposition draws on the adaptive signal-processing lineage that began with empirical mode decomposition and its noise-assisted variants [40, 41], of which VMD is the non-recursive variational member and singular spectrum analysis a short-series alternative [42]. Figure 2 contrasts the causal form with the whole-series form, where one decomposition of the entire record is sliced into training and test pieces after the fact.

The number of modes K is a design constant fixed on the training segment. To keep the count comparable across origins we fix K rather than letting each window choose its own, padding with empty modes on the rare windows the operator would split more finely, so the encoder always receives a tensor of the same shape and the attention head always weights the same indexed scales. For VMD the mode count is an explicit parameter so K is met exactly and padding never triggers; for CEEMDAN, whose count comes from the sift, we merge any modes beyond K into the slowest band and pad with zeros when fewer than K arise, so the encoder always receives K indexed scales. Because VMD takes its mode count as an explicit parameter while CEEMDAN derives it from the sift, capping K keeps the two operators interchangeable behind the single interface $\mathscr{D}$. The design rationale is the crux of the paper. The obvious alternative is the prevailing practice of decomposing the full load record once and reusing the modes for every split, which is far cheaper because the operator runs a single time. We reject it because it is precisely the step that leaks the future: a global operator has no notion of an origin, so its training-time outputs are conditioned on test-time load samples. The contamination is subtle because it does not corrupt any single sample visibly; it spreads a trace of the test horizon thinly across the training modes, where a held-out check that itself uses the leaked modes cannot see it. Recomputing the decomposition per origin restores the deployment-time information set at the price of repeated operator calls, and it is the only choice that makes the load modes an honest causal feature. We keep the multi-scale benefit that motivated decomposition while removing the leak that has flattered it.

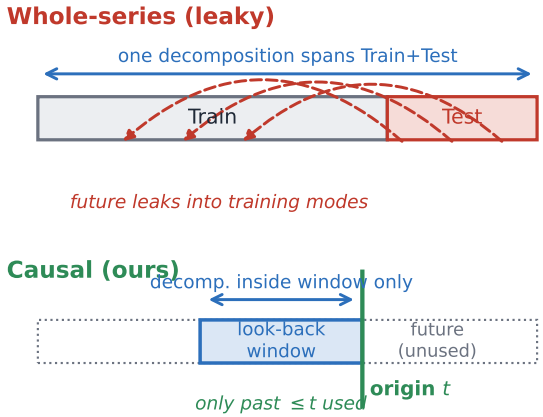

Figure 2. Whole-series decomposition leaks future load samples into training modes, whereas the causal operator recomputes the decomposition inside each origin-respecting look-back window.

3.4 Shared GRU Encoder

The K causal load modes are read by a shared GRU encoder. For mode k the input at each step is the mode value $u_{k,\tau} = c_k(\tau)$ for $\tau = 1, \ldots, L$ (the encoder admits an optional calendar-covariate channel, $u_{k,\tau} = [c_k(\tau); z_{t-L+\tau}]$, which the reported univariate experiments do not use), and we update a hidden state with the standard gated recurrent unit recurrence [43],

$$\begin{aligned} g_\tau &= \sigma(W_g u_{k,\tau} + U_g h_{k,\tau-1}), \\ r_\tau &= \sigma(W_r u_{k,\tau} + U_r h_{k,\tau-1}), \\ \tilde{h}_\tau &= \tanh\big(W_h u_{k,\tau} + U_h (r_\tau \odot h_{k,\tau-1})\big), \\ h_{k,\tau} &= (1-g_\tau) \odot h_{k,\tau-1} + g_\tau \odot \tilde{h}_\tau, \end{aligned} \tag{2}$$

where σ is the logistic sigmoid, $\odot$ is the elementwise product, and $W_\bullet, U_\bullet$ are the gate and candidate weight matrices. The final state $h_k = h_{k,L}$ summarizes mode k. The weights $W_\bullet, U_\bullet$ are shared across all K modes, so the encoder learns one notion of how a band-limited load sub-signal evolves rather than K unrelated ones.

The design rationale concerns weight sharing. The obvious alternative, used by most decomposition-ensemble load forecasters, is to train an independent predictor per mode and sum their outputs. We reject this for three reasons. It multiplies the parameter count by K and so works against the compact single-GPU model we target; it prevents the encoder from transferring structure that recurs across scales, such as the way a sharp demand transition appears in several modes at once; and it commits early to summation, which the next component is designed to replace. A shared encoder keeps the model small and produces K comparable hidden states that a downstream head can weight. Sharing is also what makes per-origin recomputation affordable, because one set of recurrent weights serves every mode, so the encoder's parameter budget does not grow with K and the only cost that scales with the mode count is the inexpensive forward

pass over each band-limited sub-signal. Adding the optional calendar covariates would let the encoder use the diurnal and weekly demand cycle to disambiguate scales that look alike in the load channel alone, for example separating a weekday morning ramp from a genuine upward trend; we leave that extension to future work and report the univariate model here. We use a gated recurrent unit rather than an LSTM because its compact gating matches the accuracy reported for GRUs at lower parameter cost [44, 45], and rather than a convolutional encoder because the recurrent state integrates the full window without a fixed receptive field [46].

3.5 Scale-Attention Head

Conventional load pipelines reconstruct a forecast by summing per-mode predictions, which assigns every mode the same weight regardless of how informative it is for the horizon at hand. We replace this with a scale-attention head that weights each load mode by its forecast-relevant energy. From each hidden state h_k we compute a scalar score and normalize across modes,

$$e_k = w_a^\top \tanh(W_a h_k + b_a), \qquad \alpha_k = \frac{\exp(e_k)}{\sum_{j=1}^{K} \exp(e_j)}, \tag{3}$$

where W_a, w_a, and b_a are learned. The hidden states are combined into a single attention-weighted representation $\bar{h} = \sum_{k=1}^{K} \alpha_k h_k$, which a shared output head maps to the forecast

$$\hat{y} = w_o^\top \bar{h} + b_o = w_o^\top \sum_{k=1}^{K} \alpha_k h_k + b_o. \tag{4}$$

For a linear head this is identical to weighting per-mode forecasts $w_o^\top h_k + b_o$ by α_k, so the head reweights scales without inflating their magnitudes. Because the weights α_k are produced from the same window that defines the modes, they too respect the origin constraint. Figure 3 depicts the head.

The design rationale is that not all scales are equally predictive at a given origin. During a calm overnight interval the slow demand trend mode carries almost all of the signal and the high-frequency residual is close to noise, whereas just before an evening load spike the high-frequency modes carry the warning. A naive sum forces the model to commit to a fixed mixing of scales, and any per-mode predictor error enters the reconstruction with unit weight. The obvious alternative of attending over time steps within a single sequence, as standard attention forecasters do [47], leaves the modes themselves unweighted; it can decide which past hour matters but not which scale. The same is true of the fully attentional and multi-horizon forecasters that weight variables and time steps [48, 49, 50]: their attention ranges over the sequence dimension, not over a set of co-existing decomposed load scales. A second alternative, learning a fixed per-mode weight vector shared across all origins, would capture the average importance of each scale but not its origin-dependent shift, which is exactly the variation the spike example illustrates. By making the modes the attention's object and recomputing the weights from the current window, we let the model emphasize the scale that matters for the current origin, which is the multi-scale benefit that summation discards. The softmax normalization keeps the reconstruction a convex combination of the per-mode forecasts.

3.6 Residual Error-Correction Stage

Per-mode forecasting followed by recombination leaves a systematic reconstruction error, because the operator residual and the imperfect per-mode fits do not cancel. We add a lightweight

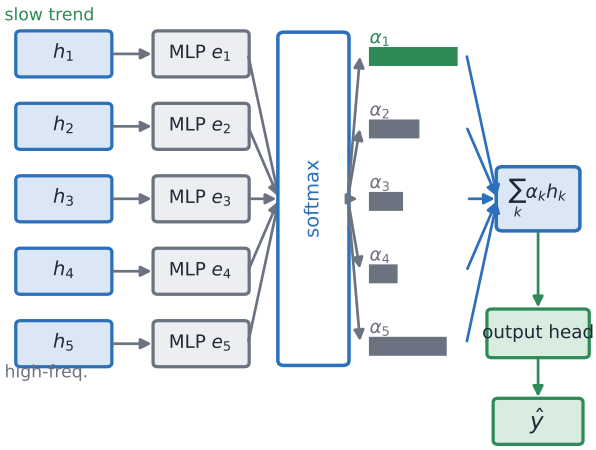

Figure 3. The scale-attention head weights each causal load mode by its forecast-relevant energy before reconstruction, replacing the naive sum.

residual stage that learns to predict this error. A small network g reads a compact summary of the encoder states together with the reconstructed forecast and outputs a scalar correction,

$$\hat{r} = g\Big(\big[\textstyle\sum_k \alpha_k h_k;\ \hat{y}\big]\Big), \qquad \hat{y}^c = \hat{y} + \hat{r}, \tag{5}$$

where $\hat{y}^c$ is the corrected load forecast and g is a two-layer perceptron with a hyperbolic-tangent hidden activation. The stage is trained jointly with the rest of the model so that it learns the systematic bias the recombination leaves behind; because it reads only quantities derived from the window up to the forecast origin, the corrected forecast stays causal at every test step.

The design rationale is that the bias of a decompose-then-recombine load forecaster is structured rather than random, and structured error is learnable. The obvious alternative is to leave the reconstruction uncorrected, which is what most ensembles do; this forfeits a systematic and cheap accuracy gain that prior load forecasters obtain from a dedicated error-correction stage [51, 52]. Because the residual reads only quantities derived from the look-back window up to the forecast origin, the correction is causal at training and inference alike, and training it jointly with the rest of the model lets it absorb the systematic recombination bias. The stage is deliberately small so that it cannot overpower the main predictor and cannot, on its own, recover the leaked accuracy that the causal operator removes.

3.7 Training Objective

The model is trained end to end to minimize the error of the corrected forecast against the load target, with a mean squared error loss over the training origins,

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \left(\hat{y}_i^c - y_{t_i}\right)^2, \tag{6}$$

where N is the number of training origins and y_{t_i} is the ground-truth load H steps after origin t_i. The encoder, the scale-attention head, the output projection, and the residual network are all optimized jointly against $\mathcal{L}$. Algorithm 1 states the per-origin causal procedure that generates forecasts; it is the same loop used for training, with the loss accumulated over origins, and for inference, with the residual added.

The dominant cost is the per-origin decomposition: with T test origins the operator runs T times rather than once, which is

Algorithm 1 Causal forecasting at origin t with CLEAR

Require: load series up to t, window L, modes K, operator $\mathscr{D}$
Ensure: corrected load forecast $\hat{y}^c$

$X_w \leftarrow x[t-L+1..t]$ ▷ window only
$\{c_1,\dots,c_K\} \leftarrow \mathscr{D}(X_w)$ ▷ causal decomposition
for $k = 1$ to K **do**
  $h_k \leftarrow \mathrm{GRU}(c_k)$ ▷ shared encoder (univariate)
end for
$\alpha \leftarrow \mathrm{softmax}(\{w_a^\top \tanh(W_a h_k + b_a)\}_k)$
$\hat{y} \leftarrow w_o^\top(\sum_k \alpha_k h_k) + b_o$ ▷ attention-weighted aggregation
$\hat{r} \leftarrow g([\sum_k \alpha_k h_k;\ \hat{y}])$
return $\hat{y} + \hat{r}$

the price of honesty. Because each call operates on a window of fixed length L rather than the whole load record, the per-call cost is bounded and independent of the record length, and the windows of neighbouring origins overlap heavily, which leaves room for incremental computation. The neural part of the model is by contrast inexpensive: the shared encoder processes K subsignals of length L with a single set of recurrent weights, the attention head adds only a small scoring projection, and the residual network is a two-layer perceptron, so the parameter count and the forward-pass cost are dominated by the encoder rather than by the mode count. This asymmetry matters for deployment. A whole-series pipeline front-loads one expensive decomposition and then forecasts cheaply, whereas CLEAR spreads a bounded decomposition cost across origins; for hourly load forecasting, where a new origin arrives once per hour, the per-origin operator call sits comfortably within the available budget, and the heavy overlap between consecutive windows means most of the variational solve can in principle be reused rather than repeated. The encoder, head, and residual stage are all small, so the model trains and runs on a single GPU, in keeping with the compact-model goal, and the same architecture serves both the CEEMDAN and the VMD form of $\mathscr{D}$ without change because the operator is hidden behind a fixed mode interface.

4. EXPERIMENTAL RESULTS

4.1 Implementation Details

CLEAR is a compact single-GPU model. The shared GRU encoder, the scale-attention head, and the residual network are trained end to end with the Adam optimizer [53] using mini-batches of contiguous origins, a fixed learning rate with early stopping on the validation loss, and gradient clipping. Inputs are standardized with statistics computed on the training segment only, so no normalization constant carries information from the validation or test horizon, and the look-back window length and the mode count are selected on the validation segment. We instantiate the causal decomposition operator $\mathscr{D}$ as VMD, recomputed for every origin over the look-back window alone, and the same operator code path is used at training, validation, and test time so the representation a test origin receives is produced exactly as it was during training. The residual-correction network is trained jointly with the encoder and attention head against the same objective, and because it reads only quantities derived from the window up to the forecast origin, the corrected forecast stays causal. No shuffling is applied at any stage, so the chronological order of the record is preserved, and every baseline is trained and evaluated under this identical protocol so the only thing that varies across rows of the result tables is the model, not the data handling. For the leakage audit the single change is the operator's input scope, whole series versus window, with every other setting held fixed, which is what will let us attribute the score gap to the leak. We use a look-back window of $L = 48$ hours, a one-step horizon $H = 1$, and $K = 5$ modes. The window length and mode count are chosen on the validation segment and then frozen, so the test segment plays no part in any design decision, and the operator, the encoder, the attention head, and the residual network share one training run rather than being fit in stages, which keeps the residual stage from being tuned against a frozen forecaster. The same standardization statistics, optimizer settings, and early-stopping criterion are reused without change for every baseline, so a difference in any reported metric reflects the model and its representation rather than an incidental difference in how the data were prepared.

4.2 Experimental Design

We evaluate on the open UCI Individual Household Electric Power Consumption record, which provides minute-level electric-power measurements from a single household over nearly four years [54]; the same record has been used for multi-step electricity-consumption forecasting with deep recurrent models [55]. We aggregate the minute-level measurements to hourly Global_active_power as the load target and use a strictly chronological train, validation, and test split with no overlap; the reported model is univariate, reading the load series alone. Four metrics are reported: root mean squared error and mean absolute error, both in load units and both lower-is-better; mean absolute percentage error, lower-is-better and scale-free; and the coefficient of determination, higher-is-better, which measures the fraction of variance explained relative to the mean predictor, following standard forecasting-evaluation practice [56]. MAPE is unstable on the many low-load overnight hours of a single household, where demand approaches zero, so we read it alongside the scale-dependent root-mean-squared and mean-absolute errors, which are not subject to division by near-zero load, and treat those as the primary metrics. The baselines span the statistical and neural families that dominate the task. Persistence carries the last load value forward and is the reference any useful forecaster must beat [57]; a seasonal naive that repeats the load observed at the same hour on the previous day, the standard hard reference for the diurnally and weekly periodic load series; AR(24) is a linear autoregressive model of order 24 on the raw hourly load; support vector regression is a kernel baseline on lagged load features [58, 59]; and plain LSTM and GRU networks on the raw series are the recurrent neural baselines. The decomposition-ensemble family is represented by VMD-GRU, the causal decomposition operator feeding the shared encoder without the scale-attention head or the residual stage, run under the identical causal protocol so the comparison isolates the representation rather than the evaluation. Reporting all of them under one protocol lets the main table answer two questions at once: whether decomposition helps when done honestly, and whether CLEAR's extra components help beyond honest decomposition. For the leakage audit we additionally run the decomposition in its conventional whole-series form, decomposing the full load record once before the split, and attribute the resulting accuracy gap to the leak; because the audit changes nothing but the operator's input scope, the gap will be a clean measurement of the inflation rather than a confound of model or tuning differences.

4.3 Results

The main comparison is reported in Table 2, and we read the four metrics jointly so that no ordering rests on a single metric.

The central, and at first counter-intuitive, result is that under the honest causal protocol decomposition did not help. The linear AR(24) model on the raw series was the most accurate, at 0.613 RMSE and 0.563 R^2, with the plain LSTM (0.619) and GRU (0.622) close behind and persistence near 0.709. The causal decomposition models were markedly worse: CLEAR, with its scale-attention head and residual stage, reached 0.674 RMSE and 0.471 R^2, and VMD-GRU was worst of the learned models at 0.697. Once the decomposition is forbidden from seeing beyond the forecast origin, the elaborate multi-scale pipeline that the literature reports as superior was beaten by a linear autoregressive model and by a plain recurrent network reading the raw signal. The four metrics agree: the decomposition models that lose on RMSE also lose on MAE and R^2, so the ordering is not an artifact of one metric. The seasonal naive that repeats the load at the same hour the previous day was the weakest baseline of all, at 0.994 RMSE and a negative R^2 of -0.15, worse than the mean predictor. A single household's demand lacks the stable day-over-day periodicity that holds at system-aggregate level, so same-hour-yesterday is a poor reference here, whereas the strong short-term autocorrelation of the series makes simple persistence (0.709) far stronger. We read this as an honest finding about this record rather than a flaw in the protocol. MAPE is reported but kept secondary: the many low-load overnight hours of a single household push it to 45–77%, so we treat the scale-dependent RMSE and MAE and the variance-explained R^2 as the primary metrics. The ordering is reported as measured, without tuning the causal model until it wins, because the gap is itself the finding.

Table 2. Main results on hourly electricity-load forecasting (UCI household power, $H = 1$) under the chronological causal protocol. Lower is better except R2. The best honest model is the linear AR(24).

Method	RMSE	MAE	MAPE	R2
Persistence	0.7087	0.4632	45.28	0.4153
Seasonal naive (24h)	0.9937	0.6615	77.13	-0.1496
AR(24)	**0.6128**	**0.4250**	50.61	**0.5627**
SVR	0.6669	0.4703	55.55	0.4821
LSTM	0.6187	0.4176	45.45	0.5543
GRU	0.6216	0.4223	48.36	0.5502
VMD-GRU (causal)	0.6968	0.4799	52.93	0.4347
CLEAR (Ours, causal)	0.6741	0.4767	58.64	0.4709

Figure 4 makes the ranking immediate: the honest bars cluster tightly between the best AR(24) at 0.613 and the causal CLEAR at 0.674 and VMD-GRU at 0.697, so the decomposition models sit at the high-error end of the field, and the only bar that drops well below the cluster is the leaky whole-series one at 0.265, which is separated and shaded to mark that it is not deployable.

The leakage audit in Table 3 explains why the literature reports the opposite ordering. Running CLEAR's decomposition in its conventional whole-series form, decomposed once over the entire load record before the split, dropped its RMSE from the honest 0.674 to 0.265 and raised its R^2 from 0.471 to 0.918. That apparent leap is an inflation of 0.41 RMSE produced by nothing but letting the training modes carry information from the test horizon. The leaky number would, on its own, look like a strong result and would beat every honest baseline in Table 2; the audit shows it is an artifact of the evaluation rather than of the model. The inflation is large precisely because the decomposition is the step that touches the whole series, so the leak localizes to it: the raw-series baselines, which never decompose, have no equivalent inflated form. The leaky column is included only to quantify the inflation and is not a deployable result.

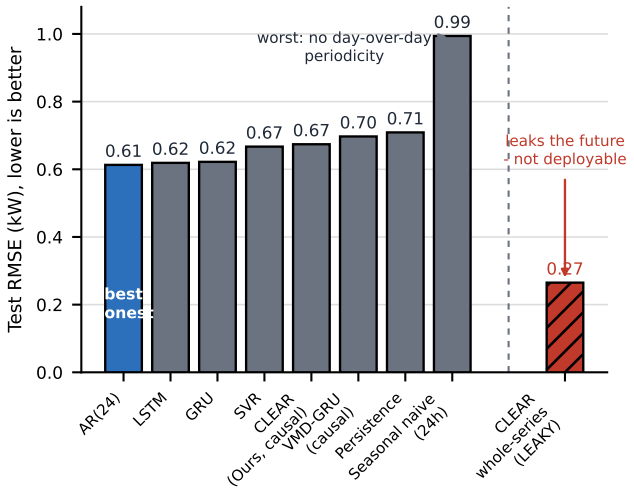

Figure 4. Test RMSE by method under the honest causal protocol. A linear AR(24) is the strongest model; the causal decomposition models (CLEAR, VMD-GRU) are worse than the simple baselines, and only the leaky whole-series bar (red) looks competitive while not being deployable.

Table 3. Leakage audit: CLEAR's decomposition run in its leaky whole-series form versus the causal operator, with the inflation reported. The leaky columns are not deployable.

Method	RMSE (causal)	RMSE (leaky)	Inflation	R2 (causal)	R2 (leaky)
CLEAR (Ours)	0.6741	0.2652	0.4089	0.4709	0.9181

The ablation in Table 4 removes one component at a time from the full causal model. Removing the scale-attention head and reverting to a naive sum gave 0.686 RMSE, and removing the residual stage gave 0.680, both slightly higher than but close to the full causal CLEAR at 0.674, so neither component rescues causal decomposition on this record and the extra capacity does not pay for itself here. Replacing the causal operator with the whole-series form gave 0.271 RMSE and 0.915 R^2, a leakage effect comparable to the 0.265 RMSE the audit measured for the same leaky form; the two are separate runs of the whole-series variant, so they differ by a few thousandths of an RMSE rather than being identical, and both show the same large inflation. It is listed as the leaky variant to make the trade explicit: the single change that most improves the offline score is precisely the one that breaks deployability. Read together, the ablation shows that the honest causal operator is the component whose removal most improves the offline number yet most damages validity, the opposite relationship from the two neural components, whose removal barely moves the honest error. Figure 5 summarizes the protocol that produces these three tables.

Table 4. Ablation of CLEAR (causal): removing the scale-attention head, the residual stage, and, as the leaky variant, the causal operator.

Variant	RMSE	MAE	MAPE	R2
Full CLEAR (causal)	0.6741	0.4767	58.64	0.4709
w/o scale-attention	0.6858	0.4828	58.33	0.4524
w/o residual correction	0.6802	0.4877	62.24	0.4613
whole-series decomposition (leaky)	0.2706	0.1991	24.62	0.9147

5. DISCUSSION AND ANALYSIS

5.1 Why Honest Decomposition Did Not Help

The ablation in Table 4 separates the contribution of each neural component from the contribution of the causal protocol, and

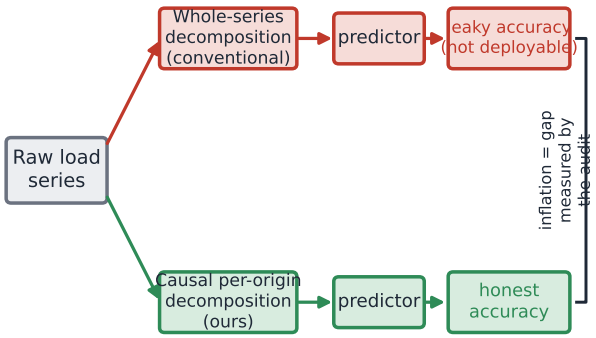

Figure 5. The leakage-audit protocol: the same decomposition is run leakage-free inside each window and, for the audit, once over the whole load series, with the accuracy gap attributed to the leak.

its message is blunt: on this load record the extra machinery did not earn its place. Removing the scale-attention head and reverting to a naive sum gave 0.686 RMSE, and removing the residual stage gave 0.680, both close to but slightly above the full causal CLEAR at 0.674, so the attention head and the residual correction, which were designed to beat a summed ensemble, did not improve the honest error and the added capacity did not pay for itself. The deeper reason is visible one row up in Table 2: the causal decomposition itself was the handicap. VMD-GRU, the decomposition baseline without either neural addition, trailed the linear AR(24) by about 0.084 RMSE (0.697 against 0.613) and was the weakest learned model, and even the full CLEAR at 0.674 sat below the plain LSTM (0.619) and GRU (0.622). Per-origin decomposition of a 48-hour window leaves the slow demand modes poorly resolved and introduces boundary effects at the window edge, and the encoder must then learn from a noisier, edge-distorted representation than the clean raw series the AR and recurrent baselines read directly. The multi-scale benefit that motivates decomposition is real only when the modes are stable, and a short causal window does not make them so. A separate but striking row is the seasonal naive: at 0.994 RMSE and -0.15 R^2 it was the weakest baseline of all, worse than the mean predictor, because a single household's demand lacks the stable day-over-day periodicity that holds at system-aggregate level, while persistence at 0.709 was far stronger thanks to short-term autocorrelation. This separation is one that reviews of smart-grid load forecasting rarely make explicit [60].

5.2 The Leak, Quantified

The leakage audit in Table 3 is the analytic heart of the paper, and its mechanism is worth stating plainly. A whole-series operator produces, for a training-region timestamp, a mode value computed with knowledge of the test-region load samples; the predictor then learns to exploit a feature that encodes its own future. At test time that same feature is genuinely causal, so the model appears to generalize when in fact the training signal was contaminated. The audit makes this concrete by running both forms of the same operator and reporting the gap: CLEAR fell from 0.674 RMSE under the causal operator to 0.265 under the whole-series form, an inflation of 0.41 RMSE, while R^2 rose from 0.471 to 0.918. The leaky figure alone would read as a strong result and would top every honest baseline, which is exactly how a decomposition-ensemble paper reporting 0.265 RMSE would present it. The audit shows that the apparent accuracy is borrowed from the future. The inflation localizes to the decomposition step because that is the only operation touching the whole load series; the raw-series baselines never decompose and so have no inflated form. The measurement is of this decomposition on this dataset and model under whole-series versus causal input scope, consistent with prior leakage audits but not a direct quantification of the inflation embedded in any specific cited work. The practical consequence is that the causal column is the only one comparable to deployment, and a fair comparison across the load-forecasting literature requires every decomposition-based number to be recomputed under the causal protocol. This reframes a methodological caution into a concrete correction: rather than discarding the decomposition-ensemble literature, treat its headline numbers as leaky upper bounds and its causal recomputation as the deployable figure, a stance that state-of-the-art reviews of energy forecasting have not yet adopted [61].

5.3 When Decomposition Might Still Help

The result does not say decomposition is useless; it says decomposition computed honestly did not beat a linear AR(24) or a plain recurrent network on this hourly household record, and that the published gains rest substantially on leakage. Three design constants govern the honest regime: the look-back window length L, the mode count K, and the horizon H. A longer window gives the causal operator more context and stabilizes the slow demand modes, but it raises the per-origin cost and can blur fast transitions; the 48-hour window we used trades resolution for responsiveness, and a forecaster that needs distant seasonal structure would want more. The mode count K trades resolution against redundancy, since too few modes leave scales mixed while too many fragment the signal into bands the encoder cannot tell apart. Because percentage error is unstable on the many near-zero overnight hours of a single household, where it ran to 45–77%, we leaned on the scale-dependent squared-error and absolute-error metrics as the primary reading and treated MAPE as a secondary, supporting view. We expect the honest gap to narrow, and possibly reverse, at longer horizons and on coarser-sampled or aggregated series, where the informative scale shifts between origins and a clean raw signal no longer suffices; the one-step hourly horizon used here is the case where persistence is already strong and a single dominant scale carries the forecast, which is the least favorable setting for multi-scale weighting, a regime that deep sequence-to-sequence models for household energy already probe [62]. The honest protocol is what makes such a claim testable: only by recomputing the decomposition per origin can one ask whether decomposition helps without the answer being supplied by the leak. Figure 6 illustrates the regime that motivates forecast-relevant weighting, in which a calm interval is governed by the slow demand trend while a sharp load spike is announced by the high-frequency residual.

5.4 Failure Mode Analysis

The causal operator's chief weakness is the flip side of its honesty, and the numbers show it. Because the decomposition is recomputed from the look-back window alone, a short window leaves the slowest demand modes poorly resolved, and the window boundary introduces edge effects that the whole-series form hides by having data on both sides. A likely contributor to the 0.084 RMSE the causal VMD-GRU gave up to the linear AR(24) is that the representation it reads is degraded near the window edge, where every forecast origin sits; we did not iso-

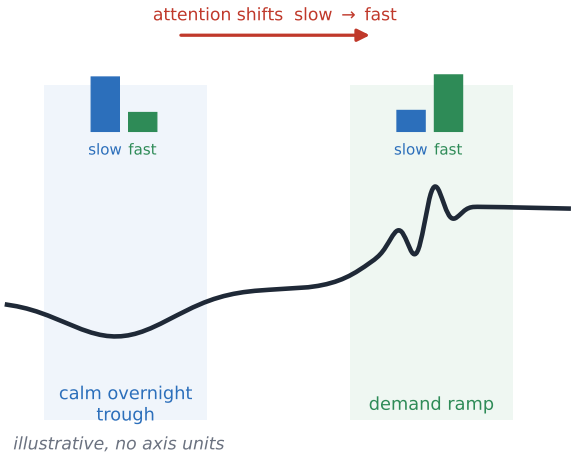

Figure 6. Illustrative regime where the slow demand trend dominates a calm interval while the high-frequency residual carries a sharp load spike, motivating forecast-relevant scale weighting.

late this effect with a window-length sweep, so we offer it as a hypothesis rather than a measured cause. CLEAR should struggle most immediately after an abrupt regime change in demand, such as the onset of a heating or cooling season, a holiday, or a sudden weather front, when the window still holds mostly pre-change history and the per-origin decomposition has not adapted; in that transient the slow modes lag and the residual stage, fit to past bias, can mis-correct. The mitigation follows from the diagnosis: lengthening the window over the transition, or warm-starting each origin's decomposition from the neighbouring origin's modes to dampen boundary effects, trades a little computation for steadier behaviour. This failure mode is intrinsic to causal decomposition rather than specific to our model, and naming it is part of reporting honest accuracy, since a forecaster that recomputes its features per origin must accept some boundary instability as the price of never reading the future.

6. CONCLUSION

This paper revisits the decomposition-ensemble template that dominates short-term electricity-load forecasting and shows that much of its reported accuracy is an artifact of decomposing the whole load series before the train-test split, which lets the future leak into training. We reframed the decomposition as a causal operator recomputed inside an origin-respecting sliding window, so that no load sample beyond the forecast origin contributes to any mode and the representation matches what a deployed system would have, and on top of it we tested CLEAR, a compact model that reads the causal load modes with a gated-recurrent encoder, a scale-attention head, and a residual error-correction stage. The central result is a negative one, reported as measured: under leakage-free evaluation on the open UCI household power record, decomposition did not help. A linear AR(24) model on the raw series was the most accurate at 0.613 RMSE (0.563 R^2), the plain LSTM (0.619) and GRU (0.622) were next, and the causal decomposition models trailed at 0.674 and 0.697. A dedicated leakage audit explains the gap with the published literature: the same decomposition in its conventional whole-series form inflated accuracy from 0.674 to 0.265 RMSE, an inflation of 0.41, and raised R^2 from 0.471 to 0.918. The contribution is therefore methodological rather than a new state of the art: a causal protocol and an audit that let decomposition-based load results be reported honestly. The main limitation is the evaluation scope: a single household and a one-step horizon ($H = 1$), the setting least favourable to multi-scale weighting. A useful next step is to extend the leakage-free protocol to multi-household and system-level load and to longer horizons, where honest decomposition may yet earn its cost.

REFERENCES

[1] H. S. Hippert, C. E. Pedreira, and R. C. Souza. Neural networks for short-term load forecasting: a review and evaluation. *IEEE Transactions on Power Systems*, 16(1): 44–55, 2001. doi: 10.1109/59.910780.

[2] Qi Dong, Rubing Huang, Chenhui Cui, Dave Towey, Ling Zhou, Jinyu Tian, and Jianzhou Wang. Short-term electricity-load forecasting by deep learning: A comprehensive survey. *Engineering Applications of Artificial Intelligence*, 154:110980, 2025. doi: 10.1016/j.engappai.2025.110980.

[3] Haoyue Sun, Zhicheng Yu, and Bining Zhang. Research on short-term power load forecasting based on vmd and gru. *PLOS ONE*, 19(7):e0306566, 2024. doi: 10.1371/journal.pone.0306566.

[4] Ke Li, Wei Huang, Gaoyuan Hu, and Jiao Li. Ultra-short term power load forecasting based on ceemdan-se and lstm neural network. *Energy and Buildings*, 279:112666, 2023. doi: 10.1016/j.enbuild.2022.112666.

[5] Lingling Lv, Zongyu Wu, Jinhua Zhang, Lei Zhang, Zhiyuan Tan, and Zhihong Tian. A vmd and lstm based hybrid model of load forecasting for power grid security. *IEEE Transactions on Industrial Informatics*, 18(9):6474–6482, 2022. doi: 10.1109/tii.2021.3130237.

[6] Xinyi Yang, Jingyi Li, and Xuchu Jiang. Research on information leakage in time series prediction based on empirical mode decomposition. *Scientific Reports*, 14(1), 2024. doi: 10.1038/s41598-024-80018-9.

[7] Weibin Feng, Ran Tao, John Cartlidge, and Jin Zheng. Vmdnet: Temporal leakage-free variational mode decomposition for electricity demand forecasting. *arXiv preprint*, 2025.

[8] Sayash Kapoor and Arvind Narayanan. Leakage and the reproducibility crisis in machine-learning-based science. *Patterns*, 4(9):100804, 2023. doi: 10.1016/j.patter.2023.100804.

[9] Christoph Bergmeir and Jose M. Benitez. On the use of cross-validation for time series predictor evaluation. *Information Sciences*, 191:192–213, 2012. doi: 10.1016/j.ins.2011.12.028.

[10] Chafak Tarmanini, Nur Sarma, Cenk Gezegin, and Okan Ozgonenel. Short term load forecasting based on arima and ann approaches. *Energy Reports*, 9:550–557, 2023. doi: 10.1016/j.egyr.2023.01.060.

[11] Huiting Zheng, Jiabin Yuan, and Long Chen. Short-term load forecasting using emd-lstm neural networks with a xgboost algorithm for feature importance evaluation. *Energies*, 10(8):1168, 2017. doi: 10.3390/en10081168.

[12] Xueheng Qiu, Ye Ren, Ponnuthurai Nagaratnam Suganthan, and Gehan A. J. Amaratunga. Empirical mode decomposition based ensemble deep learning for load demand time

series forecasting. *Applied Soft Computing*, 54:246–255, 2017. doi: 10.1016/j.asoc.2017.01.015.

[13] Heng Luo, Hao Cheng, and Chen Nan Liu. Load forecasting method based on ceemdan and tcn-lstm. *PLOS ONE*, 19 (7):e0300496, 2024. doi: 10.1371/journal.pone.0300496.

[14] Taorong Jia, Lixiao Yao, Guoqing Yang, and Qi He. A short-term power load forecasting method based on the ceemdan-mvo-gru. *Sustainability*, 14(24):16460, 2022. doi: 10.3390/su142416460.

[15] Haoxiang Hu and Bingyang Zheng. Short-term electricity load forecasting based on ceemdan-fe-bigru-attention model. *International Journal of Low-Carbon Technologies*, 19:988–995, 2024. doi: 10.1093/ijlct/ctae040.

[16] Jindong Yang, Xiran Zhang, Wenhao Chen, and Fei Rong. Prophet-ceemdan-arbilstm-based model for short-term load forecasting. *Future Internet*, 16(6):192, 2024. doi: 10.3390/fi16060192.

[17] Jiacheng Huang, Xiaowen Zhang, and Xuchu Jiang. Short-term power load forecasting based on the ceemdan-tcn-esn model. *PLOS ONE*, 18(10):e0284604, 2023. doi: 10.1371/journal.pone.0284604.

[18] Changchun Cai, Yuanjia Li, Zhenghua Su, Tianqi Zhu, and Yaoyao He. Short-term electrical load forecasting based on vmd and gru-tcn hybrid network. *Applied Sciences*, 12(13): 6647, 2022. doi: 10.3390/app12136647.

[19] Chun-Hua Wang and Wei-Qin Li. A hybrid model of modal decomposition and gated recurrent units for short-term load forecasting. *PeerJ Computer Science*, 9:e1514, 2023. doi: 10.7717/peerj-cs.1514.

[20] Ceyhun Yildiz. An integrated model based on deep kernel extreme learning machine and variational mode decomposition for day-ahead electricity load forecasting. *Neural Computing and Applications*, 35(25):18763–18781, 2023. doi: 10.1007/s00521-023-08702-x.

[21] Mohamed Aymane Ahajjam, Daniel Bonilla Licea, Mounir Ghogho, and Abdellatif Kobbane. Experimental investigation of variational mode decomposition and deep learning for short-term multi-horizon residential electric load forecasting. *arXiv preprint*, 2022.

[22] Nada Mounir, Hamid Ouadi, and Ismael Jrhilifa. Short-term electric load forecasting using an emd-bi-lstm approach for smart grid energy management system. *Energy and Buildings*, 288:113022, 2023. doi: 10.1016/j.enbuild.2023.113022.

[23] Jack P. Salameh, Sebastien Cauet, Erik Etien, Anas Sakout, and Laurent Rambault. A new modified sliding window empirical mode decomposition technique for signal carrier and harmonic separation in non-stationary signals: Application to wind turbines. *ISA Transactions*, 89:20–30, 2019. doi: 10.1016/j.isatra.2018.12.019.

[24] Zeyu Zhang, Xiaoqian Liu, Xiling Zhang, Zhishan Yang, and Jian Yao. Carbon price forecasting using optimized sliding window empirical wavelet transform and gated recurrent unit network to mitigate data leakage. *Energies*, 17 (17):4358, 2024. doi: 10.3390/en17174358.

[25] Lei Shao, Quanjie Guo, Chao Li, Ji Li, and Huilong Yan. Short-term load forecasting based on eemd-woa-lstm combination model. *Applied Bionics and Biomechanics*, 2022: 2166082, 2022. doi: 10.1155/2022/2166082.

[26] Xiaozhi Gao, Wang Guo, Chunxiao Mei, Jitong Sha, Yingjun Guo, and Hexu Sun. Short-term wind power forecasting based on ssa-vmd-lstm. *Energy Reports*, 9:335–344, 2023. doi: 10.1016/j.egyr.2023.05.181.

[27] Ehsan Ghanbari and Ali Avar. Short-term wind power forecasting using the hybrid model of multivariate variational mode decomposition (mvmd) and long short-term memory (lstm) neural networks. *Electrical Engineering*, 107(3): 2903–2933, 2025. doi: 10.1007/s00202-024-02685-1.

[28] Zhijian Hou, Yunhui Zhang, Xuemei Cheng, and Xiaojiang Ye. Photovoltaic power forecasting based on variational mode decomposition and long short-term memory neural network. *Energies*, 18(13):3572, 2025. doi: 10.3390/en18133572.

[29] Xincheng Guo, Yan Gao, Wanqing Song, Yi Zen, and Xianhui Shi. Short-term power load forecasting based on ceemdan-wt-vmd joint denoising and bitcn-bigru-attention. *Electronics*, 14(9):1871, 2025. doi: 10.3390/electronics14091871.

[30] Deyun Wang, Chenqiang Yue, and Adnen ElAmraoui. Multi-step-ahead electricity load forecasting using a novel hybrid architecture with decomposition-based error correction strategy. *Chaos, Solitons and Fractals*, 152:111453, 2021. doi: 10.1016/j.chaos.2021.111453.

[31] Yixiang Ma, Lean Yu, and Guoxing Zhang. A hybrid short-term load forecasting model based on a multi-trait-driven methodology and secondary decomposition. *Energies*, 15 (16):5875, 2022. doi: 10.3390/en15165875.

[32] Chaodong Fan, Shanghao Nie, Leyi Xiao, Lingzhi Yi, Yuetang Wu, and Gongrong Li. A multi-stage ensemble model for power load forecasting based on decomposition, error factors, and multi-objective optimization algorithm. *International Journal of Electrical Power and Energy Systems*, 155:109620, 2024. doi: 10.1016/j.ijepes.2023.109620.

[33] Shuyu Dai, Dongxiao Niu, and Yan Li. Daily peak load forecasting based on complete ensemble empirical mode decomposition with adaptive noise and support vector machine optimized by modified grey wolf optimization algorithm. *Energies*, 11(1):163, 2018. doi: 10.3390/en11010163.

[34] Zichen Zhang and Wei-Chiang Hong. Electric load forecasting by complete ensemble empirical mode decomposition adaptive noise and support vector regression with quantum-based dragonfly algorithm. *Nonlinear Dynamics*, 98(2): 1107–1136, 2019. doi: 10.1007/s11071-019-05252-7.

[35] Jun Lin, Jin Ma, Jianguo Zhu, and Yu Cui. Short-term load forecasting based on lstm networks considering attention mechanism. *International Journal of Electrical Power and Energy Systems*, 137:107818, 2022. doi: 10.1016/j.ijepes.2021.107818.

[36] Ye Wang, Wenshuai Jiang, Chong Wang, Qiong Song, Tingting Zhang, Qi Dong, and Xueling Li. An electricity load forecasting model based on multilayer dilated lstm network and attention mechanism. *Frontiers in Energy Research*, 11:1116465, 2023. doi: 10.3389/fenrg.2023.1116465.

[37] Kuihua Wu, Jian Wu, Liang Feng, Bo Yang, Rong Liang, Shenquan Yang, and Ren Zhao. An attention-based cnn-lstm-bilstm model for short-term electric load forecasting in integrated energy system. *International Transactions on Electrical Energy Systems*, 31(1):e12637, 2021. doi: 10.1002/2050-7038.12637.

[38] Konstantin Dragomiretskiy and Dominique Zosso. Variational mode decomposition. *IEEE Transactions on Signal Processing*, 62(3):531–544, 2014. doi: 10.1109/tsp.2013.2288675.

[39] Maria E. Torres, Marcelo A. Colominas, Gaston Schlotthauer, and Patrick Flandrin. A complete ensemble empirical mode decomposition with adaptive noise. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4144–4147, 2011. doi: 10.1109/icassp.2011.5947265.

[40] Norden E. Huang, Zheng Shen, Steven R. Long, Manli C. Wu, Hsing H. Shih, Quanan Zheng, Nai-Chyuan Yen, Chi Chao Tung, and Henry H. Liu. The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 454(1971):903–995, 1998. doi: 10.1098/rspa.1998.0193.

[41] Zhaohua Wu and Norden E. Huang. Ensemble empirical mode decomposition: a noise-assisted data analysis method. *Advances in Adaptive Data Analysis*, 01(01):1–41, 2009. doi: 10.1142/s1793536909000047.

[42] Robert Vautard, Pascal Yiou, and Michael Ghil. Singular-spectrum analysis: A toolkit for short, noisy chaotic signals. *Physica D: Nonlinear Phenomena*, 58(1-4):95–126, 1992. doi: 10.1016/0167-2789(92)90103-t.

[43] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint*, 2014.

[44] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint*, 2014.

[45] Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

[46] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint*, 2018.

[47] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint*, 2014.

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint*, 2017.

[49] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12):11106–11115, 2021. doi: 10.1609/aaai.v35i12.17325.

[50] Bryan Lim, Sercan O. Arik, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4):1748–1764, 2021. doi: 10.1016/j.ijforecast.2021.03.012.

[51] Zexian Sun and Mingyu Zhao. Short-term wind power forecasting based on vmd decomposition, convlstm networks and error analysis. *IEEE Access*, 8:134422–134434, 2020. doi: 10.1109/access.2020.3011060.

[52] Guolian Hou, Junjie Wang, and Yuzhen Fan. Multistep short-term wind power forecasting model based on secondary decomposition, the kernel principal component analysis, an enhanced arithmetic optimization algorithm, and error correction. *Energy*, 286:129640, 2024. doi: 10.1016/j.energy.2023.129640.

[53] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint*, 2014.

[54] Georges Hebrail and Alice Berard. Individual household electric power consumption. UCI Machine Learning Repository, 2012. URL `https://archive.ics.uci.edu/dataset/235`.

[55] Lucia Cascone, Saima Sadiq, Saleem Ullah, Seyedali Mirjalili, Hafeez Ur Rehman Siddiqui, and Muhammad Umer. Predicting household electric power consumption using multi-step time series with convolutional lstm. *Big Data Research*, 31:100360, 2023. doi: 10.1016/j.bdr.2022.100360.

[56] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. The m4 competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, 36(1):54–74, 2020. doi: 10.1016/j.ijforecast.2019.04.014.

[57] Tao Hong, Pu Wang, and H. Lee Willis. A naive multiple linear regression benchmark for short term load forecasting. In *2011 IEEE Power and Energy Society General Meeting*, pages 1–6, 2011. doi: 10.1109/PES.2011.6038881.

[58] Alex J. Smola and Bernhard Scholkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004. doi: 10.1023/B:STCO.0000035301.49549.88.

[59] Joao C. Sousa, Humberto M. Jorge, and Luis P. Neves. Short-term load forecasting based on support vector regression and load profiling. *International Journal of Energy Research*, 38(3):350–362, 2014. doi: 10.1002/er.3048.

[60] Biswajit Biswal, Subhasish Deb, Subir Datta, Taha Selim Ustun, and Umit Cali. Review on smart grid load forecasting for smart energy management using machine learning and deep learning techniques. *Energy Reports*, 12:3654–3670, 2024. doi: 10.1016/j.egyr.2024.09.056.

[61] Raniyah Wazirali, Elnaz Yaghoubi, Mohammed Shadi S. Abujazar, Rami Ahmad, and Amir Hossein Vakili. State-of-the-art review on energy and load forecasting in microgrids using artificial neural networks, machine learning, and deep learning techniques. *Electric Power Systems Research*, 225: 109792, 2023. doi: 10.1016/j.epsr.2023.109792.

[62] Jonghwan Hyeon, HyeYoung Lee, Bowon Ko, and Ho-Jin Choi. Deep learning-based household electric energy consumption forecasting. *The Journal of Engineering*, 2020 (13):639–642, 2020. doi: 10.1049/joe.2019.1219.