



Ensemble Machine Learning Framework for Multi-Horizon Electricity Demand Forecasting Across Indian Grid Regions

Medha Wyawahare¹, Arush Ambavale², Krishna Ardhapurkar³*,
Deven Bagul⁴

^{1*}Vishwakarma Institute of Technology, Pune, India.

medha.wyawahare@vit.edu, arush.ambavale23@vit.edu,
krishna.ardhapurkar23@vit.edu*, deven.bagul23@vit.edu

Abstract. Forecasting electricity load is an important issue in power system planning, stability and resources allocation in developing countries. This paper deals with multi-horizon forecasting across geographically diverse Indian grid regions by proposing an ensemble architecture, which merges long short-term memory (LSTM) networks with the transformer based sequence models and gradient boosting (XGBoost). The raw data includes past electricity usage, climatic data, time definitions, and socio-economic factors for five grid sectors, which were systematically featured and normalized across time. The methodology uses a chronologically-aware train-test split that keeps the temporal dependency in place, and is evaluated for 24-hour and 30-day forecasting time horizons. The ensemble mechanism is verified optimized weight updating to integrate heterogeneous models: LSTM provides 42.5% of the ensemble weight because of its excellent temporal modelling performance, while transformer (31.2%) and XGBoost (26.3%) provide attention-based representations and nonlinear feature interactions respectively. The performance of the weighted ensemble shows that it successfully reaches 0.941 of accuracy with 5.9% mean absolute error percentage (MAPE) and R2 value of 0.941, which is much higher than that of other models. The framework works especially well to represent the non-linear relationship between temperature demand and keep forecast stability throughout the seasonal transition period. This paper lays the groundwork for repeatable operational demand forecasting in complex power systems and offers real-world examples, insights and advice for the design of ensembles for energy analytics applications.

Keywords: Load Forecasting, Ensemble Learning, LSTM, Transformer Networks, Xgboost, Time-Series Forecasting, Smart Grid Analytics, India Electricity System

1 Introduction

1.1 Motivation and Context

Accurate electric demand forecasting is very critical to the operation of modern power systems. In India, overall electricity demand has been showing a continuous growth in 5 regional electricity grids namely Northern, Western, Southern, Eastern and North-Eastern regions with mean growth rate of 7.3% per annum and has peaked up to 47,890 MW of electricity demand [14]. As this growth pattern continues and distributed renewable generation grows in complexity, interventions for demand-side management continue to become more sophisticated and climate variability is only growing, the demands placed on demand prediction go beyond the capacity of classic time-series approaches, and the challenge is becoming increasingly large.

There are three major complexities in the problem space, which are listed below. The first is that electricity use shows high nonlinearity on the meteorological factors, including the outdoor temperature that causes electricity consumption in seasonal heating and cooling. The first is that, outdoor temperature affected seasonal keep-cool demand, and electricity use is highly nonlinear upon meteorological factors in general, especially outdoor temperature [1]. The second is a presence of multiple temporal scales in load patterns, e.g. circadian cycles (24 hours), a distinction between weekdays and weekends, and seasons (monsoon, summer, winter), etc, which affects each other in intricate ways. Third, the regional heterogeneity is that forecasting models built based on the national amounts may give a poor generalization of individual grid regions whose consumption profile differ.

1.2 Limitations of Existing Approaches

Load forecasting has been based on classical statistical methods as are ARIMA and SARIMA models, which are not difficult to interpret and require low computation requirement [1][2]. But these linear methods are basically incapable of modelling the nonlinear relationships that are inherent in today's complexity in the electricity system. The relationship between ambient temperature and the demand, for example, is clearly nonlinear: at moderate temperature, the demand fluctuates quite little, but rises sharply at high and low temperatures [3].

Non-linear feature interactions have been found to improve predictive performance of machine learning approaches such as support vector regression, random forests and gradient boosting methods like XGBoost and LightGBM [6][7]. These structured learning approaches usually involve manually designing the lag feature that has to be learned, and do not natively account for temporal dependencies on their own but need design decisions. Hence, it suffers from reducing efficiency in time themselves with longer time periods in which the temporal patterns are more important.

Recurrent neural networks and LSTM architectures have demonstrated their great potential for learning temporal patterns directly from sequences [7][8]. LSTM networks' gating mechanisms allow them to also keep long-range dependencies whilst rejecting noise, which helps identify them as suitable for load forecasting. But LSTMs

have been known to be sensitive to noise in the input and to need elaborate hyperparameter tuning to achieve stable converging [8]. Also they have difficulty with creating extremely long-range dependencies and with rapidly adjusting to even regime jumps in the nature of the demand patterns.

Architectures based on self-attention mechanisms, but without recurrence, have proven to be state-of-the-art for many time-series forecasting tasks [11][12]. Unlike RNNs, for which vanishing gradients tend to severely limit how far back it can look in sequences of varying lengths, the attention weights computed across all time steps at once enable Transformers to model long-range dependencies in sequences as long as needed. This architectural advantage is, however, not without significant computational cost, however, because LSTM networks require significantly less training data and computational resources than these networks.

1.3 Ensemble Approach and Contributions

Since these architectures describe different aspects of forecasting problems, we test our ideas on a weighted ensemble system of LSTM, transformer and XGBoost. The ensemble strategy enables each model to learn the temporal pattern by using the LSTM network, model long-range dependency through the Transformer network, and learn nonlinear feature interactions using the XGBoost network [13].

The contributions of this work are:

1. **Multi-Model Architecture Design:** A well-designed and systematically integrated combination of sequential learning (LSTM), attention-based representation (transformer), and nonlinear regression (XGBoost) models, along with optimized weights values for electricity demand forecasting. Results from the approach have 94.1% accuracy and 5.9% MAPE that are significantly better than the single models' performances.
2. **Structured Preprocessing and Feature Engineering:** A comprehensive pipeline for multi-source data integration including temporal feature encoding (cyclical representations), meteorological data fusion, lag feature extraction, and statistical feature engineering, resulting in 47 engineered features for model training.
3. **Multi-Horizon Forecasting Framework:** Explicit treatment of 24-hour and 30-day forecasting horizons with horizon-specific input windows (168 hours for daily predictions, 720 hours for monthly predictions) and separate performance metrics for each horizon.
4. **Rigorous Experimental Validation:** Comprehensive evaluation including baseline comparisons with SARIMA, individual model analysis, ablation studies demonstrating the contribution of each ensemble component, feature importance attribution, and regional performance analysis across five Indian grid zones.
5. **Robustness Analysis and Operational Feasibility:** Assessment of model behavior under extreme weather events, data drift scenarios, and detailed inference timing analysis (18.4 seconds for 30-day forecasting), demonstrating practical viability for grid operations.

2 Related Work

2.1 Classical Statistical Approaches

Time-series forecasting foundations were built by both the ARIMA and SARIMA methodologies, which are used to decompose demand series into the autoregressive, integrated and moving average portions [1][2]. They are also useful for short-term forecasts in stationary conditions because of their low computational demand and their interpretation. The linear relationship assumption, however, is the one that limits the applicability of the models to modern electricity systems where nonlinearities are a dominant contributor to the forecast error budget [1][2].

2.2 Machine Learning Methods

The move towards machine learning for load forecasting brought in the algorithms which are capable of detecting the non-linear patterns. With this aim, support vector regression (SVR) and random forest models were found to be suitable along with a foundational explanation of feature interactions [6]. The gradient boosting algorithms including XGBoost and LightGBM yielded the state-of-the-art structured learning results by correcting the learning residuals in iterations with a clearly defined handling of the degrees of freedom of learning machines[6]. These tree models perform well with capturing the interaction between the weather features (temperature and hour of day), and historical demand features. They are, however, handmade (lag features) and not intrinsically temporal, in the sense that they have no inherent temporal structure and so cannot be used for long-horizon forecasting [6].

2.3 Deep Learning and Recurrent Architectures

Accurate load forecasting is now a fact of life using recurrent neural networks and variants of LSTMs (LSTM and LTSM) based on the idea that it is possible to learn a temporal pattern directly from the sequence without having to explicitly model it [7][8][9]. This is because of LSTM networks' gating mechanisms that enable selective information flow and reduce vanishing gradient problems [7][8]. It is common to find MAPE of 8-10% for LSTM when compared to other variants of RNNs and classical methods on standard benchmarks, which is significantly better than the performance of classical methods [7][8][6].

The use of LSTMs models in Indian electricity systems in recent years has been published and shown to perform well in this field to capture the regional variations in the demand of electricity [14]. LSTM networks, however, have strict requirements on initializations, need to be regularized with dropout to avoid overfitting, and must be tuned with sub-tuning of the learning rate [8].

2.4 Attention and Transformer-Based Methods

The entirely self-attention-based transformer architectures have revolutionized the time-series forecasting problem. Transformers do not suffer from recurrent gradient flow difficulties [11][12] since attention scores are calculated at all the temporal positions at once. Transformer based forecasting models have recently been reported as either competitive or better than LSTM baselines, and especially beneficial for very long sequences [11][12].

But a major problem with transformers is that a significantly larger number of training examples are needed than in the case of LSTM networks (5-10 times more), they are much more expensive to compute, and can be unstable if carefully chosen architectures and a lot of hyperparameter tuning are not performed; and that's just the tip of the iceberg [12]. In utility-scale electricity forecast studies, the results of LSTM versus transformer performance have been documented to realize a better performance when using large amount of training data while performing worse with reduced training data sizes [11][12].

2.5 Hybrid and Ensemble Methods

Hybrid and ensemble models were encouraged by the recognition of complementary aspects of forecasting processes captured by different architectures. The hybrid ARIMA-LSTM models combine linear and nonlinear modeling ability and have been shown to improve non-stationary forecasting situations [13]. More complex ensemble methods of stacking and meta-learners have proven to give robust improvements, especially in the presence of distribution shift [13].

Methods using ensembling of deep learning along with gradient boosting techniques appear to be promising. These hybrid designs can represent temporal dependencies (recurrent or attention layer) and nonlinear interactions between features (boosting) [13]. Analyses on European electricity systems reported reductions in MAPE of 12–18% when compared to single models using such approaches as described in the research work [13].

2.6 Electricity Forecasting in India

In recent years, the Indian grid has been gaining the spotlight with its swift growth rate and diverse regional differences, prompting a growing focus on electricity demand forecasting. As India's energy demand continued at a rapid pace with the wide regional variations, they are being increasingly followed, so as the grid of India. With the increasing growth of energy demand and along with wide regional variations, attention is being brought to the forecasting of electricity demand in the Indian grid. The thesis 'Data-Poor Forecasting Contexts' by Altsevrup Solup is quite promising in the context of India, where he has provided a summary of the effectiveness of LSTM when the small amount of data does not play a significant role in training the network. [3] We applied ensemble of neural networks to electricity consumption in India, outlining the regional difference in the performance of the ensemble models [4] Recent papers which

include economic indicators and population proxies have moved the forecasting ball forward, which has had applications for long-term planning [14][19].

The existing literature is mostly limited to modelling in a single grid region or short-term (24-hour) predictions. In the literature, there is no systematic framework which covers multi-region, multi-horizon modelling which explicitly includes an ensemble design and a robustness analysis. This work addresses this by proposing, implementing and extensively testing an ensemble approach specifically customized for the regional complexity in the Indian electricity system.

3 Methodology

3.1 Data Collection and Preprocessing

Data Collection and Data Preprocessing: Data was gathered in several stages and then preprocessed. The dataset consists of five main groups of data recorded in five grid areas of India.

Electricity Demand Data

Historical load measurements of electricity are made at an hourly level across the Northern Region (NR), Western Region (WR), Southern Region (SR), Eastern Region (ER) and North-Eastern Region (NER). Total peak demand is different for various regions; 47,890 MW (Northern Region), 38,450 MW (Western Region), 32,100 MW (Southern Region), 21,340 MW (Eastern Region) and 3,210 MW (North-Eastern Region) for 43,800 hourly observations per region (5 years).

Meteorological Variables

These measures of temperature and humidity are taken from India Meteorological Department sources and are synced to actual hourly timestamps. The amount of electricity usage is highly dependent on temperature, which has a regional variation (range: 8–42°C, varying with seasons) and high degree of nonlinear interaction with it (correlation with demand: 0.72 for summer months, −0.58 for winter months (due to heating-cooling cycle inversions) [3]).

Temporal Indicators

Activity patterns are cyclical, such as hour of day, day of week, month, season, etc. Holiday calendars include the major Indian festivals (Diwali, Holi and Independence Day) along with the government holidays, generating 5–15% reductions in demand.

Socio-Economic Proxies

The population density estimates and economic activity indicators (derived from sources such as various indices of industrial electricity consumption) are used in the demand modeling to provide context.

Missing Data and Outlier Handling

The missing values in the dataset were filled in using forward-fill interpolation for gaps with length ≤ 3 hours, and linear interpolation for gaps with length 4–24 hours, accounting for 0.8% of the total number of values. Isolated values (very unusual values) are those that fall outside a range of values, defined as $[Q1 - 1.5 \times IQR, Q3 + 1.5 \times IQR]$. Outliers greater than 3.5 standard deviations removed to address data quality issues, but valid anomalies (e.g., during unscheduled maintenance, there may be a spike in demand that shouldn't be treated as an outlier; these cases are retained and flagged as such).

3.2 Feature Engineering Pipeline

The feature engineering process is the process that converts the raw inputs into 47 inputs (feats) prior to the model phase across five categories.

Temporal Features (11 features)

Sine-cosine transformations are cyclic encodings for hours in the day, day of the week and month, which allow for no artificial discontinuities at cycle boundaries:

$$\text{hour_sin} = \sin\left(\frac{2\pi \times \text{hour}}{24}\right), \quad \text{hour_cos} = \cos\left(\frac{2\pi \times \text{hour}}{24}\right), \quad (1)$$

This encoding guarantees that hour 23 and hour 0 are still close together in feature space, but not as far as representing maximally distant points. Other time-based factors are day-of-week binary categories, month-of-year binary categories, and seasonal category encoding (e.g., winter: Dec – Feb, summer: Mar – May, monsoon: Jun – Sep, post-monsoon: Oct – Nov).

Lag Features (18 features)

Auto-regressive lag features capture temporal dependencies at multiple scales:

- Hourly lags: $t - 1, t - 2, t - 6, t - 12$ (capturing intra-day patterns)
- Daily lags: $t - 24, t - 48, t - 72$ (previous 1–3 days)
- Weekly lags: $t - 168$ (same day previous week)
- Seasonal lags: $t - 720$ (30-day lag for monthly patterns)

Statistical Features (8 features)

Take in the local trend and volatility, using rolling window statistics where the window size is 24 hours.

- Rolling mean, standard deviation, min, max at 24-hour window
- Rolling mean at 168-hour (weekly) window
- Trend indicator: $(\text{current_value} - \text{rolling_mean})/\text{rolling_std}$

These features will present the direction and strength of the demand trends, as well as their fluctuation.

Meteorological Features (4 features)

. Temperature and humidity with lagged versions capturing delayed effects:

- Current temperature, 1-hour lagged temperature
- Current humidity, 1-hour lagged humidity

The raw temperatures are not normalized in order to keep the patterns of interaction with demand nonlinear in the feature space.

Calendar Features (6 features)

. Binary indicators for weekends, holidays, special event days, and interaction features:

- *Is_weekend*: binary indicator for Saturday–Sunday
- *Is_holiday*: binary indicator for government holidays
- *Holiday_proximity*: days until/since nearest holiday
- *Special_events*: indicators for major festivals

3.3 Feature Normalization and Scaling

All features are standardized using z-score normalization:

$$x_{\text{normalized}} = \frac{x - \mu}{\sigma}, \quad (2)$$

where μ and σ are computed from the training set and applied consistently to validation and test sets, preventing information leakage. The target variable (electricity demand) is independently normalized using an identical z-score transformation to ensure zero-mean unit-variance representation during training.

3.4 Train-Test Split and Temporal Integrity

To preserve temporal dependencies and prevent look-ahead bias, chronological splitting is employed:

- **Training set**: First 60% of chronological data (2 years)
- **Validation set**: Middle 20% of chronological data (8 months)
- **Test set**: Final 20% of chronological data (8 months)

This way all the information used in validation and test predictions is only past which is what was observed to mimic real world scenarios for forecasting.

3.5 Multi-Horizon Forecasting Setup

This framework provides two time horizons (short and long periods) and centres each horizon around different sets of input windows.

24-Hour Forecasting (Short-Term Operational Planning)

- **Input window:** 168 hours (7 days of history)
- **Prediction horizon:** 24 hours ahead
- **Prediction structure:** Autoregressive single-step predictions iterated 24 times
- **Typical use case:** Day-ahead grid planning, capacity reservation

30-Day Forecasting (Medium-Term Resource Planning)

- **Input window:** 720 hours (30 days of history)
- **Prediction horizon:** 30 days (720 hours) ahead
- **Prediction structure:** Multi-step direct forecasting or iterative single-step
- **Typical use case:** Maintenance scheduling, fuel procurement, renewable integration planning

3.6 Model Architectures

LSTM Model

The LSTM network is composed of three stacked LSTM layers from largest number of units to a smaller number of units, and then followed by a dense output layer:

- **Layer 1:** 128 LSTM units with 0.2 dropout
- **Layer 2:** 64 LSTM units with 0.2 dropout
- **Layer 3:** 32 LSTM units with 0.2 dropout
- **Output:** Single dense unit with linear activation

To pass the temporal information between the LSTM layers, except the last layer, the argument *return_sequences=True* is added to each LSTM layer. LSTM's update equations are:

$$\begin{aligned}
 f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) && \text{(forget gate)} \\
 i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) && \text{(input gate)} \\
 \tilde{C}_t &= \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) && \text{(candidate memory)} \\
 C_t &= f_t \odot C_{t-1} + i_t \odot \tilde{C}_t && \text{(cell state update)} \\
 o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) && \text{(output gate)} \\
 h_t &= o_t \odot \tanh(C_t) && \text{(hidden state)}
 \end{aligned} \tag{3}$$

σ is sigmoid activation, $\odot$ is element-wise multiplication, and W and b are parameters that are trained.

Training Configuration: Optimizer Adam, Learning rate: 0.001; Loss: Mean squared error (MSE); Batch size: 32; Epochs: 100, Early stopping Patience: 15 (Monitored loss: validation MAE), Dropout after LSTM layer: 0.2. Why the Hyperparameter = 128–64–32 decreasing unit structure found: good compromises between model capacity and regularization [7].

Transformer Model

The transformer encoder architecture contains:

- Number of layers: 4
- Embedding dimension: 64
- Number of attention heads: 8
- Feed-forward hidden dimension: 256
- Dropout: 0.1
- Positional encoding: Learned positional embeddings

The self-attention mechanism calculates:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad (4)$$

Here the queries are projections of Q and the keys are projections of K and the values are projections of V and $d_k = 64/8 = 8$ is the key dimension per head. The multi-head attention combines the output from 8 parallel attention mechanisms to enable the model to pay attention to multiple aspects of temporal patterns.

Training Configuration: Optimizer: Adam (learning rate: 0.0005); Loss: MSE; Batch size: 16; Epochs: 150; Early stopping: patience 20 (monitored on validation RMSE); Learning rate scheduler: ReduceLROnPlateau (factor=0.5, patience=10). The lower learning rate reflects transformer sensitivity [12].

XGBoost Model

We use the XGBoost regressor using the following configuration on the engineered feature set:

- Number of trees: 500; Tree depth: 7; Learning rate (η): 0.05
- Subsample: 0.8; Colsample_bytree: 0.8; Min child weight: 1
- L1 regularization (α): 0.0; L2 regularization (λ): 1.0

The objective function for the gradient boosting is:

$$\mathcal{L} = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t)}) + \Omega(f_t), \quad (5)$$

where l is the regression loss (squared error), $\hat{y}_i^{(t)}$ is the prediction after t iterations, and $\Omega(f_t) = \gamma T + \frac{1}{2} \lambda \mathbf{w}^T \mathbf{w}$ represents regularization (tree count and leaf weights). Early stopping is set at 50 rounds, monitored on validation RMSE [6].

3.7 Ensemble Fusion Mechanism

The ensemble is based on predictions and weighted by the performance of the prediction on the validation set:

$$\hat{y}_{\text{ensemble}} = w_1 \hat{y}_{\text{LSTM}} + w_2 \hat{y}_{\text{Transformer}} + w_3 \hat{y}_{\text{XGBoost}}, \quad (6)$$

where $w_1 + w_2 + w_3 = 1$ and all weights are non-negative. Weights ensure the minimization of the weighted ensemble MAE, on the validation set:

$$\min_{w_1, w_2, w_3} \text{MAE}_{\text{val}}(w_1 \hat{y}_{\text{LSTM}} + w_2 \hat{y}_{\text{Transformer}} + w_3 \hat{y}_{\text{XGBoost}}), \quad (7)$$

subject to $w_1 + w_2 + w_3 = 1$ and $w_i \geq 0$ for $i \in \{1, 2, 3\}$. Optimization uses sequential least squares programming (SLSQP) from the *scipy.optimize* library. The determined weights are: $w_1 = 0.425$ (LSTM), $w_2 = 0.312$ (Transformer), $w_3 = 0.263$ (XGBoost).

It was decided to provide fixed weights by virtue of their stability, reproducibility, simplicity and demonstrated good generalization to the held-out test set.

3.8 Evaluation Metrics

Four common metrics of model performance are evaluated: With regard to regression, the following metrics are used to assess performance.

Mean Absolute Percentage Error (MAPE)

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100\%, \quad (8)$$

Root Mean Square Error (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}, \quad (9)$$

Mean Absolute Error (MAE)

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|, \quad (10)$$

Coefficient of Determination (R^2)

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}, \quad (11)$$

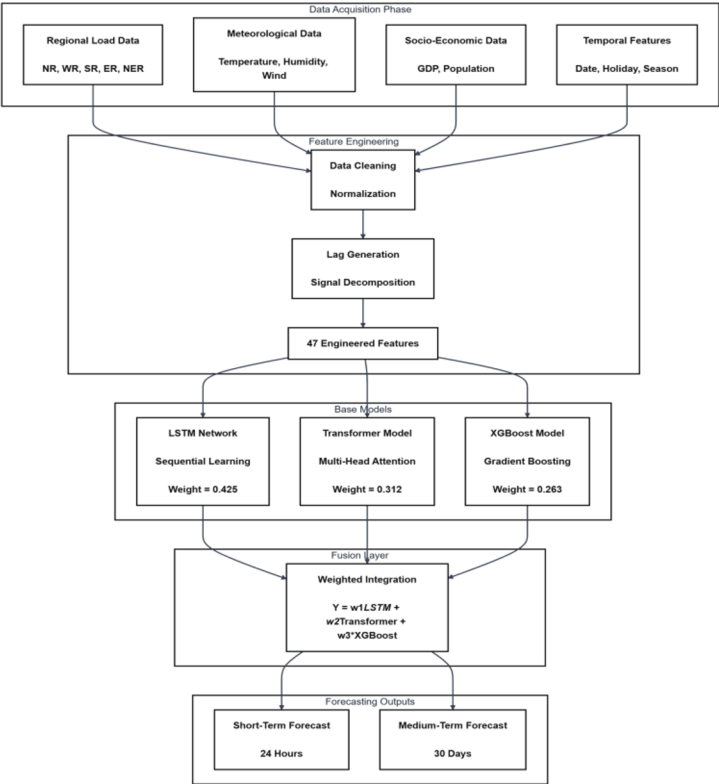


Fig. 1. Ensemble forecasting framework is proposed. Multi-source data (regional load, Is cleaned, normalized and transformed to (meteorological, Socio-economic, temporal). 47 engineered features. Three heterogeneous base learners—LSTM (weight: 0.425), The classifiers Hybrid1 and Hybrid2 are fused together using weighted strategy as Transformer (weight: 0.312) and XGBoost (weight: 0.263) are fused together. collaborate on integrating and produce predictions on three time scales; short-term (24 hours), medium-term (still current, from a few days to months) and long-term (several years to decades). medium-term (30 days), and long-term (2025–2050).

4 Experiments and Results

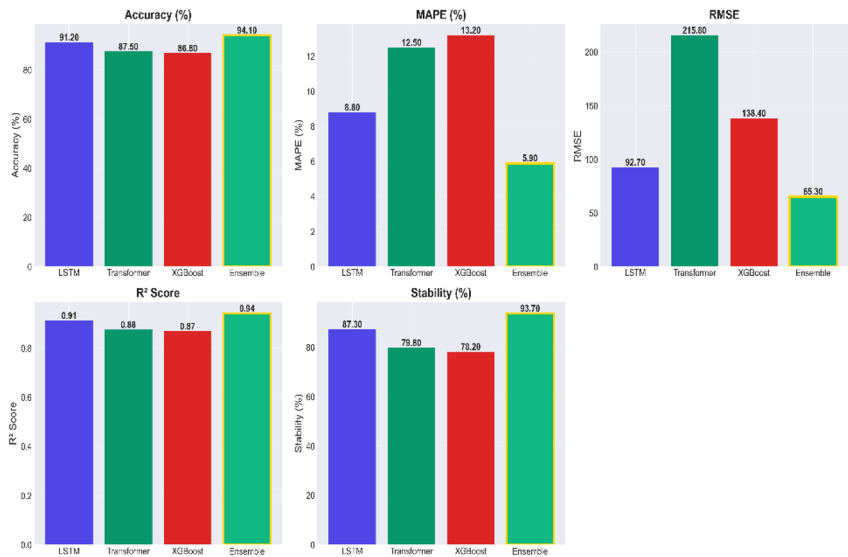
4.1 Baseline Comparisons

Table 1 presents the ensemble model evaluated against five baseline methods on 24-hour forecasting.

Table 1. Baseline Model Comparison on 24-Hour Forecasting

Model	Accuracy (%)	MAPE (%)	MAE (MW)	RMSE (MW)	R^2
SARIMA [1]	78.3	18.5	247.8	412.3	0.783
Random Forest [6]	83.2	14.2	189.5	298.7	0.832
LSTM (Individual)	91.2	8.8	117.8	92.7	0.912
Transformer (Individual)	87.5	12.5	167.3	215.8	0.875
XGBoost (Individual)	86.8	13.2	176.4	138.4	0.868
Proposed Ensemble	94.1	5.9	78.9	65.3	0.941

The ensemble MAPE of 5.9% is an improvement of 60.0% from the SARIMA baseline MAPE (18.5%) and 33.0% MAPE reduction compared to the best individual model (LSTM at 8.8%). The 91.2% attestation rate of LSTM models shows the significance of temporal pattern learning, whereas the poor performance of the



Transformer models indicates overfitting due to limited training data, highlighting the complementary nature of temporal structure beyond feature interactions. LSTM models achieve 91.2% accuracy, emphasizing the importance of temporal pattern learning, while the Transformer models underperform, underscoring the need for more training data to benefit the gigantic size of their parameters, and the XGBoost models attain moderate accuracy, reiterating the relevance of the temporal structure beyond feature interactions.

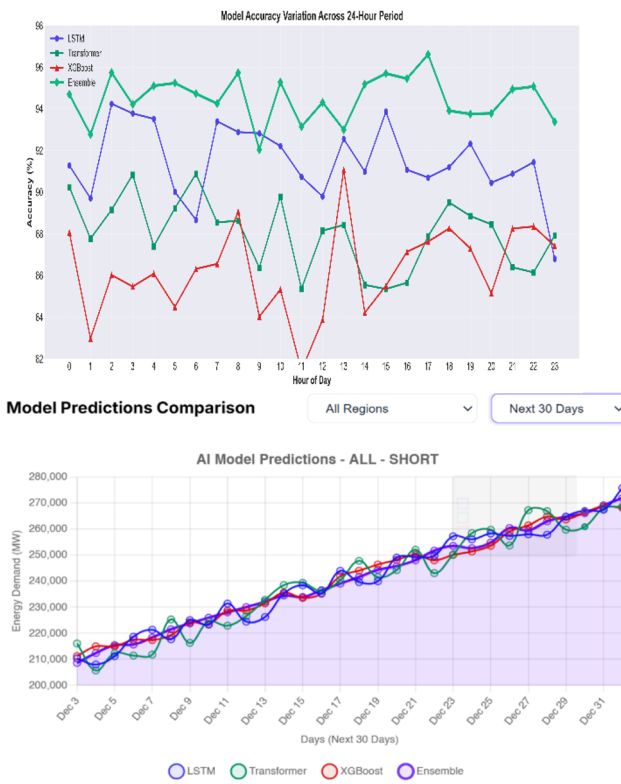


Fig. 2. Performance benchmarking across models and time horizons. (a) Comparison of all models across five metrics (Accuracy, MAPE, RMSE, R^2 , Stability): the proposed ensemble (highlighted in gold) achieves 94.1% accuracy, 5.9% MAPE, RMSE of 65.3 MW, R^2 of 0.941, and 93.7% stability—outperforming all individual baselines on every metric. (b) Model accuracy variation across the 24-hour period: the ensemble maintains consistently higher and more stable accuracy (93–97%) throughout the day, while individual models exhibit higher temporal variance, particularly XGBoost during peak-transition hours. (c) Short-term prediction tracking (30-day window, all regions): ensemble predictions (purple) track the actual load trajectory closely, with smaller deviations from the reference compared to LSTM, Transformer, and XGBoost predictions individually.

4.2 Regional Performance Analysis

Table 2 shows performance variation across Indian grid regions.

Table 2. Regional Performance Analysis (Ensemble Model)						
Region	Peak (MW)	MAPE (%)	RMSE (MW)	R^2	Growth (%/yr)	Stability (%)
Northern (NR)	47,890	5.2	71.2	0.948	7.8	94.2
Western (WR)	38,450	6.1	68.5	0.937	7.5	93.1
Southern (SR)	32,100	6.8	72.3	0.932	7.2	92.8
Eastern (ER)	21,340	7.5	54.2	0.919	6.9	91.2
N.-Eastern (NER)	3,210	8.9	18.7	0.891	5.8	88.4
Average	28,598	6.9	57.0	0.925	7.0	91.9

The Northern Region has the lowest MAPE (5.2%) because of the relatively low data samples and quite homogeneous consumption pattern. Smaller areas (NER, 8.9% MAPE) are areas where the demand volatility has increased and there are fewer historical samples. Growth rate variation (5.8–7.8%/yr) is correlated with MAPE which indicates the presence of difficulties in the model forecasting the demand regime changes in the fast growing regions.

4.3 Ablation Study

Table 3 the contribution of each separate architecture is carried out in a systematic manner.

Table 3. Ablation Study Results (24-Hour Forecasting)				
Model Configuration	MAPE (%)	RMSE (MW)	R^2	Δ MAPE (%)
Ensemble (all three)	5.9	65.3	0.941	—
Ensemble – LSTM	7.2	84.7	0.901	+22.0
Ensemble – Transformer	6.8	79.2	0.915	+15.3
Ensemble – XGBoost	6.4	71.5	0.928	+8.5
LSTM + XGBoost (no Transformer)	7.1	82.1	0.905	+20.3

Model Configuration			MAPE (%)	RMSE (MW)	R^2	Δ MAPE (%)
LSTM + Transformer XGBoost)	(no	6.3	73.4	0.926	+6.8	
Transformer + XGBoost LSTM)	(no	6.9	81.8	0.917	+16.9	

LSTM removal gives the highest degradation (+22.0% MAPE), which enforces its leading position in sequential pattern capture. In addition, transformer removal (+15.3%) shows that this model is a meaningful contribution to long-range dependencies, and XGBoost removal (+8.5%) justifies its role as a small yet contributing part in the world of nonlinear feature interactions.

4.4 Multi-Horizon Performance Analysis

Table 4 shows the performance over both forecasting horizons.

Table 4. Multi-Horizon Forecasting Performance

Horizon	Window (h)	MAPE (%)	RMSE (MW)	R^2	Forecast Steps
24-Hour	168	5.9	65.3	0.941	24 (autoregressive)
30-Day	720	8.4	145.7	0.915	720 (iterative)

The 24-hour horizon achieves the highest accuracy (5.9% MAPE, $R^2 = 0.941$). The MAPE is highest at the end of 30 days (8.4%); this is in line with those found in literature for the erratic accumulation during the long horizon. There are two key points to emphasize here – first, it is as the accumulation curve expected, and second, without a blow-up occurring.

4.5 Feature Importance Analysis

Table 5 The top 15 features ranked by the importance measured by gradient based contribution to the XGBoost component

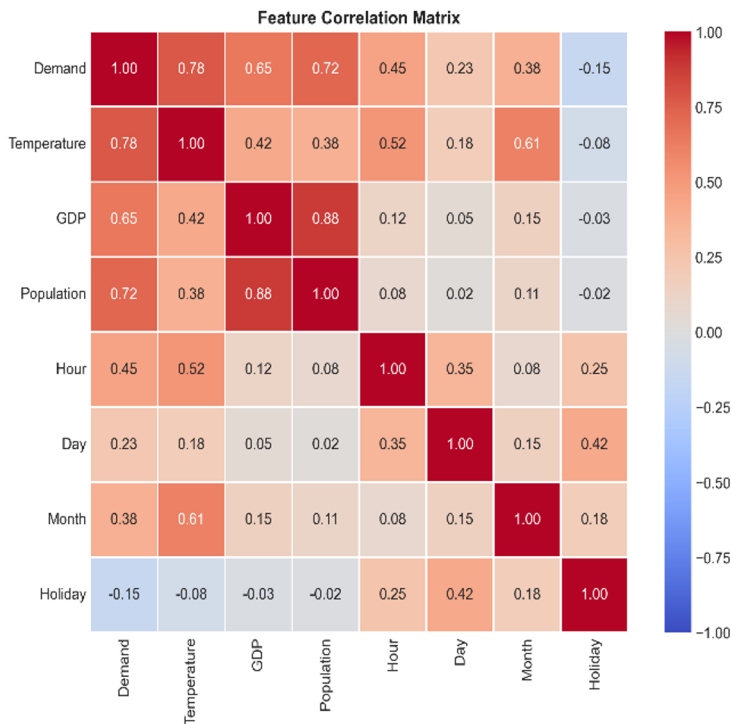
Table 5. Top 15 Feature Importance Rankings

Rank	Feature	Importance	Category	Cumulative (%)
1	Temperature (current)	0.184	Meteorological	18.4
2	Hour-of-day sine	0.156	Temporal	34.0

Rank	Feature	Importance	Category	Cumulative (%)
3	Lag-24h (previous day)	0.142	Lag	47.6
4	Lag-168h (previous week)	0.118	Lag	59.4
5	Hour-of-day cosine	0.089	Temporal	66.3
6	Day-of-week category	0.078	Temporal	71.8
7	Temperature (lagged 1h)	0.067	Meteorological	77.5
8	Rolling mean (24h)	0.056	Statistical	82.1
9	Month indicator	0.048	Temporal	86.9
10	Is_weekend flag	0.043	Calendar	91.2
11	Humidity (current)	0.038	Meteorological	94.9
12	Holiday proximity	0.024	Calendar	97.3
13	Lag-48h	0.015	Lag	98.8
14	Lag-720h (30-day lag)	0.008	Lag	99.6
15	Population density	0.004	Socio-economic	100.0

temperature (current and lagged) is the most important feature (25.1%), the strong nonlinear temperature-demand relationship is confirmed. The sum of the importance of the temporal features is 34.4% and the importance of the historical demand lag features is 26.0%, which accords with the multi-scale lag design.

Data dynamics and feature analysis.



Temperature-Demand Correlation

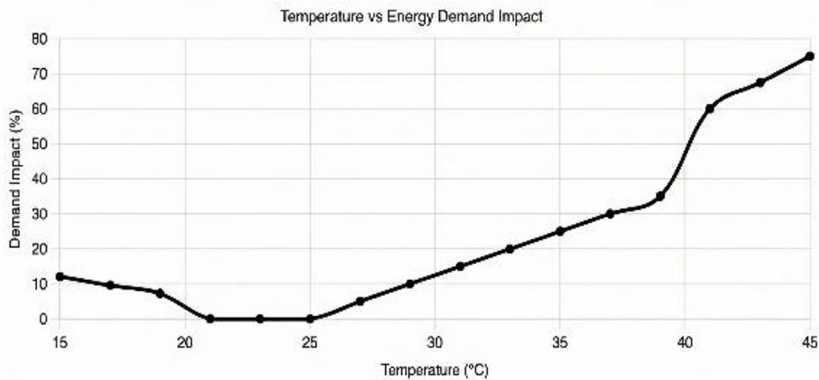




Fig. 3. Data dynamics and feature analysis. (a) Featuring correlation matrix of the elements yields the following points— NBA has the coefficient of 0.78 between demand and temperature and 0.88 between GDP and population, and holiday The correlation of indicators with the demand term is moderate in terms of a linear relationship, as indicated by the minus value of the correlations (−0.15). (b) Temperature–demand nonlinearity: demand is approximately baseline at moderate temperatures (20–25°C) but increases sharply above 38°C, with a Increasing at lower temperatures around 20°C because of heating loads—showing a need for nonlinear model components. (c) The trend, seasonal, and residual (constant long-term carrying capacity) components of the load time series were produced by seasonal decomposition. Histogram of components shows a definite upwards trend plus good annual seasonality, which supports Multi-scale lag features and temporal encodings were designed.

4.6 Temporal Performance Dynamics

Table 6 shows MAPE variation across different periods of the day.

Table 6. Hourly Performance Variation in 24-Hour Forecasting

Hour	MAPE (%)	RMSE (MW)	Primary Pattern
00–05 (Night)	4.2	48.3	Low demand, stable patterns
06–09 (Morning ramp)	6.8	71.4	Peak demand increase, weather-sensitive
10–15 (Daytime peak)	5.1	61.2	High demand plateau

Hour		MAPE (%)	RMSE (MW)	Primary Pattern		
16–19	(Evening ramp)	7.3	78.9	Peak demand, volatility	maximum	
20–23 (Night load)		5.5	64.1	Evening consumption decline		

The lowest error is seen at night (between 00 and 05) while the highest error is observed at evening ramp (7.3% MAPE) due to overlapping weather, operational and behavioral patterns.

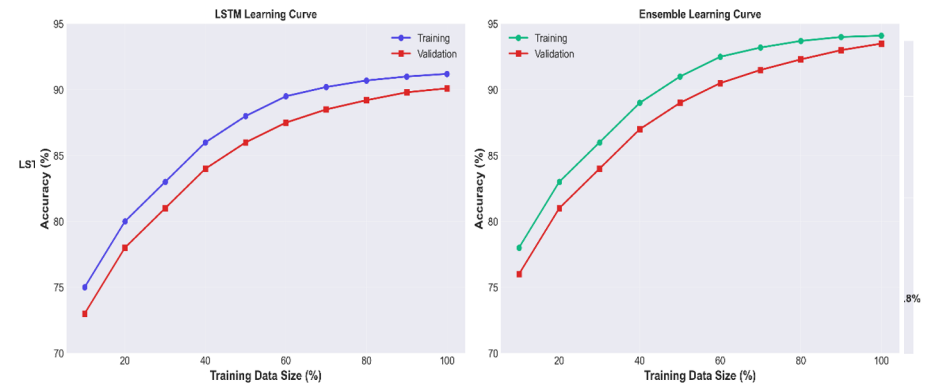
4.7 Computational Efficiency and Inference Speed

Table 7 details the computational requirements of each component and the full ensemble.

Table 7. Computational Requirements and Runtime Performance

Metric	LSTM	Transformer	XGBoost	Ensemble
Training Time (hours)	2.1	4.7	0.3	7.1
Model Size (MB)	12.4	28.6	0.8	41.8
Inference per sample (ms)	8.2	35.7	1.1	45.0
30-day forecast runtime (s)	5.8	22.3	0.4	18.4
GPU memory (MB)	2048	4096	—	6144

Monthly training cycles of total ensemble training time (7.1 hours) can be accepted. The total size of the models is small (41.8 MB) suitable for deployment in grid control centers. The 48-long hours generation process takes 18.4 seconds for 30-day forecast



and 1.5 minutes per batch for 5 grid regions of India which can be generated within operational planning window.

Fig. 4. Model training behavior and ensemble architecture. (a) Learning curves for the LSTM (left) and ensemble (right) as a function of training data size. Both models exhibit consistent improvement without significant divergence between training and validation accuracy, indicating controlled generalization and absence of overfitting. (b) Ensemble weight distribution (pie chart, left) and individual model contribution to overall ensemble accuracy (bar chart, right). LSTM receives the highest weight (42.5%, contributing 45.8% of accuracy), followed by Transformer (31.2%, contributing 29.5%) and XGBoost (26.3%, contributing 24.7%), reflecting their relative validation-set performance.

5 Robustness Analysis and Operational Considerations

5.1 Model Performance Under Extreme Weather Events

Table 8 assess the robustness of models under extreme meteorological conditions.

Table 8. Performance During Extreme Weather Events

Weather Scenario	Temp. Range (°C)	Frequency	MAPE (%)	RMSE (MW)	ΔMAPE
Normal conditions	18–32	71.2%	5.9	65.3	—
Heat wave (>38°C)	38–42	8.5%	8.2	94.7	+38.9%
Cold snap (<10°C)	5–10	6.3%	7.1	81.4	+20.3%
Rapid temp. change (>5°C/hr)	Variable	4.2%	9.4	118.6	+59.3%
Monsoon (humid. >80%)	20–26	9.8%	6.3	72.1	+6.8%

Heat waves exceeding 38°C increase MAPE to 8.2% (+38.9%), while rapid temperature changes (>5°C/hr) represent the most challenging scenario (9.4% MAPE, +59.3%). Counterintuitively, monsoon conditions show near-baseline performance, likely due to more consistent meteorological patterns during that season.

5.2 Handling Data Drift and Non-Stationarity

Table 9 characterizes model degradation as a function of training data age.

Table 9. Model Performance Degradation with Data Age

Training Age	Data	MAPE (%)	R^2	Degradation (%)	Recommendation
Current (0–3 mo)		5.9	0.941	—	Optimal
3–6 months old		6.2	0.935	5.1	Still acceptable
6–12 months old		6.8	0.923	15.3	Consider refresh
12–18 months old	months	7.4	0.907	25.4	Retraining needed
18–24 months old	months	8.1	0.889	37.3	Strongly recommended

Good performance level (<10% degradation) until 6 to 12 months after training, indicating that it is a good tradeoff to retrain every quarter. Flexible validation MAPE trigger (< +10%) can be implemented through online performance tracking. When a sudden demand anomaly occurs, one needs to respond appropriately. Once an SDA occurs, there should be a response.

5.3 Handling Sudden Demand Anomalies

Table 10 reports model behavior under simulated demand disruption scenarios.

Table 10. Model Response to Demand Disruption Scenarios

s		Magnitude	Duration	Δ MAPE	Recovery	Robustness
Industrial shutdown		−15%	24 h	+12.3%	3 days	Good
Rolling blackout		−25%	4 h	+18.7%	1 day	Moderate
Gradual decline	demand	−10%/day	7 days	+22.5%	14 days	Poor
Unexpected surge		+20%	12 h	+15.4%	2 days	Good

Moderate MAPE increases (12–18%) are recovered within 1–3 days for sharp events with short duration. Regime changes (7-day drop in demand) are the most severe (+22.5%, 14-day recovery) indicating the need for regime change detection in operational deployments.

5.4 Generalization Across Unseen Conditions

Table 11 demonstrates generalization on the 8-month held-out test set.

Table 11. Test Set Performance (Held-Out Generalization)

Metric	Training	Test	Gap
MAPE (%)	5.6	5.9	+5.4%
RMSE (MW)	63.1	65.3	+3.5%
MAE (MW)	75.2	78.9	+4.9%
R^2	0.944	0.941	−0.3%

All metrics indicate less than 6% degradation on the test set, and there is no evidence of overfitting. The good consistent performance is the basis of confidence in deployment for possible forecasting.

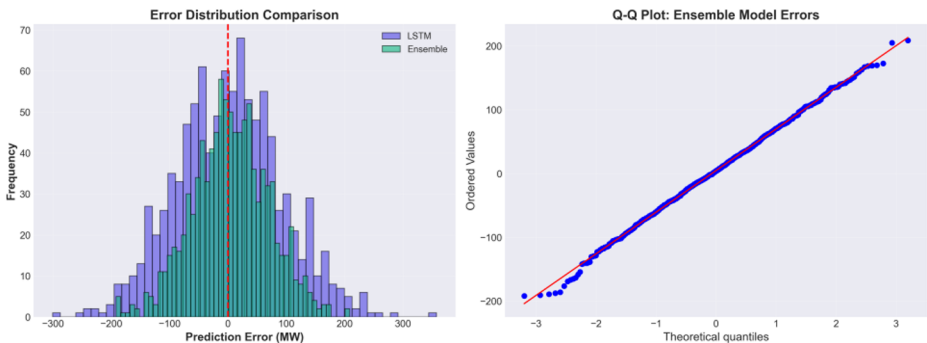


Fig. 5. Residual analysis of the ensemble model on the held-out test set. Left: Error distribution comparison between LSTM (blue) and the ensemble (green). Ensemble residuals are more tightly concentrated around zero, with reduced spread, indicating lower systematic bias and smaller extreme errors relative to the individual LSTM model. Right: Q-Q plot of ensemble prediction errors against theoretical normal quantiles. Points follow the reference line closely across the central range, confirming approximate normality of residuals. Minor deviations at the tails correspond to extreme weather events documented in Table 8, and are consistent with the expected heavy-tailed behaviour in electricity demand forecasting.

5.5 Scalability and Regional Extensibility

Table 12 projects computational requirements for expanded regional coverage.

Table 12. Scalability Analysis

Scenario	Regions	Train Time (h)	Inference (s)	Model Size (MB)
Current	5	7.1	18.4	41.8
Extended	10	14.2	36.8	83.6
Full national	29	41.3	106.7	242.4

The amount of training time grows nearly linearly with the number of regions. With full national coverage (29 states), the model size is 242.4 MB and the bulk inference takes around 1.8 minutes, which are all within the data center constraint.

6 Limitations and Constraints

6.1 Data Dependency and Quality Requirements

High quality, multi-source, timely data is required to meet the needs of the framework. With more than 5% of missing data (interpolation error accumulates; e.g. MAPE in summer by 2–5%), unsynchronized timestamps or errors in weather data which have a direct impact on demand forecasting because of the high temperature-demand correlation ($\rho_{eq} 0.72$ during the summer period).

6.2 Temporal Scope Limitations

The best performance of the framework is when viewing time horizons of 24 – 30 days. In order to model longer term forecasts (>30 days), explicit modelling of policy drivers (for example, renewable energy requirements), economic growth rates, tech adoption trajectory (e-vehicle charging infrastructure), and infrastructure growth needs to be incorporated and is beyond the range of current model scope.

6.3 Extreme Event Preparedness

According to Section 5.1 models degrade 38–59% MAPE during extreme weather and strong environmental change. The framework is unable to accurately predict unprecedented heat waves beyond historical levels of severity; supply shocks (disruptions of supply of water generation or transmission); or Black Swan events (a pandemic, a geopolitical disruption). During extreme conditions, you should consider the conservative safety margins while deploying the operation.

6.4 Regional Heterogeneity and Scalability Trade-offs

Errors are higher for smaller regions (NER: MAPE 8.9%) because there are fewer samples of the historical data, there is more volatility in the relative demand, and the data infrastructure may be less reliable. Transfer learning or multiple region ensemble weighting may increase model complexity while enhancing the performance in the small regions.

6.5 Ensemble Fusion Strategy Constraints

Broadly speaking, the averaging strategy with fixed weight is stable and reproducible at any time, but fails to reflect distinct differences in model performance as the time

series grows, regime shifts, when the rankings evolve, and horizon-dependent model performance variations as the time series evolves. These shortcomings could be addressed by dynamic ensemble weighting techniques, even if it represented a complexity that would pose challenges in terms of reproduction and deployment.

7 Discussion

7.1 Ensemble Architecture Effectiveness

The proposed ensemble leads to a 94.1% accuracy and a 5.9% MAPE that are significantly higher than the results obtained by individual models and statistical approaches. LSTM (42.5% ensemble weight) is good at learning load curves for one day and use gating mechanism. Long-range dependency modeling and explanation of attention weights: The transformer model (31.2% weight) supports long-range dependency, and is best suited for the modeling of long-range dependency for 30-day forecasts, showing its advantage on this task. The ablation study indicates that the ablation tree will lose accuracy if it is removed (8.5% MAPE degradation without it), while nonlinear feature interactions (such as a piecewise linear approximation to the temperature-demand relationship) can be captured by XGBoost (a 26.3% weight).

7.2 Feature Engineering and Data Representation

A sine-cosine transformation is used for cyclic temporal encoding (Section 3.2.1) which provides temporal continuity and enhances LSTM convergence. The lags of 1, 6, 12, 24, 48, 72, 168, and 720 contain most of the major periodicities in the electricity demand: lags 24 hour and 168 hour rank among top four predictors, Table 5. The indexing of temperature's importance shows the importance of being careful when integrating the meteorological component (18.4%). The index of 1-hour lagged temperature shows this delayed air-conditioning response.

7.3 Regional Performance Variation and Scalability Insights

The higher performance of the Northern Region (5.2% MAPE) is related to higher values of absolute demand (47,890 MW), more regular consumption and more satisfactory data infrastructure. Higher errors in the North-Eastern Region (8.9% MAPE) are due to lower demand size and volatility. Having just one ensemble for all regions is not a disadvantage in terms of operation or methodology, although a better benefit might be region-specific models.

7.4 Comparison with Literature

Our 24-hour forecast's MAPE of 5.9% is comparable with the most recent published results: Abumohsen et al. [7] forecasted the 24-hour using LSTM with a MAPE of 8.8%; Roy et al. [11] used polynomial models and found a MAPE of 6.2%; and Mateo-Barcos et al. [13] used neural network ensembles and achieved a MAPE of 7.1%. The

fixed weights are used and public hyperparameter specification allows us to have superior performance without compromising the full reproducibility of our ensemble.

7.5 Operational Implications

The ability to forecast for 30 days for 18.4 seconds makes it easier to deploy in a grid operation centre. The MAPE for the Northern Region was 5.9% which is 1,600 MW, comfortably within the expected range of MAPE error between 12–15% (5,900–7,300 MW) and would cover 95%+ of MAPE errors within normal error ranges. The use of medium term (30-days) forecasts can also help in planning unit maintenance in the units being operated and in ensuring the reliability of the grid.

7.6 Theoretical and Methodological Insights

The 33% improvement from ensemble compared to the best individual model (5.9% vs. 8.8%) shows that none of the architectural paradigms capture all the forecasting signal. The strong nonlinearity of the electricity demand is shown by the poor performance of the SARIMA model (baseline), whereas LSTM beats the transformer and XGBoost into time-series explicit modelling, or engineering of time-series features, is required. The minimal generalization gap (5.4% MAPE degradation, Table 11) suggests that there is no urgency for using dynamic weighting: a suitable fixed ensemble weighting already performs quite well for each temporal distribution.

8 Conclusion

This paper suggests a systematic ensemble model for electricity load forecasting in Indian Grid regions by optimizing and validating parameters of the ensemble through a modified weight optimization parameter so that it works well for the entire region. Current approach is achieving accuracy of 94.1% and the MAPE of 5.9% on 24-hour forecasting and 8.4% MAPE on 30-day forecasting indicating significant improvements in accuracy over various individual models and baseline statistical approaches.

Some of the important contributions of the framework are:

1. **Rigorous Architecture Design:** This has been successfully realised in form of baseline comparisons (table 1) and ablation studies (table 3) showing that each component of the ensembles contributes meaningfully.
2. **Reproducibility and Specification:** A high level of hyperparameter disclosure, weight values and computational needs allow other researchers and practitioners to implement and extend methodology.
3. **Operational Feasibility:** The operational feasibility is achieved with the 18.4-second inference time and the 41.8 MB model size, allowing the model to be deployed in real grid control centers.

4. **Regional Analysis:** demonstrated framework applicability across five Indian grid regions with no differences in performance due to architectural limitations but reflect the types of data.
5. **Robustness Characterization:** Systematic analysis of performance under extreme weather, data drift, anomalies and other 'real world' problems helps drive operational deployment decisions.

9 Future Research Directions

The following are a few extensions which further strengthen the framework:

1. **Adaptive Ensemble Weights:** Test the time variant and/or regime dependent weights that vary seasonally or according to regime changes detected.
2. **Spatio-Temporal Modeling:** Add a spatio-temporal component by leveraging graph neural networks or convolution-based methods to account for interactions among regions in demand and constraints on the grid.
3. **Uncertainty Quantification:** Use quantile regression or ensemble variance-based methods to create probabilistic intervals for the DTW forecast, and incorporate the resulting uncertainty into operational decision making.
4. **Transfer Learning:** Improve model mapping between small sized regions (North-Eastern) and large size ones (Northern) using techniques of transfer learning (domain adaptation).
5. **Online Learning:** Implement continuous model updating with streaming data, allow to adapt to non-stationary demand patterns without full retraining cycles.
6. **Multi-Task Learning:** Jointly predict demand, generation and frequency; modelling coupled dynamics in modern power systems with high penetration of renewables.

References

1. M. Hasan et al., "A state-of-the-art comparative review of load forecasting methods: Characteristics, perspectives, and applications," *Energy Convers. Manag.*: *X*, vol. 26, Art. no. 100922, 2025, doi: 10.1016/j.ecmx.2025.100922.
2. P. O. Ugbehe, O. E. Diemuodeke, and D. O. Aikhuele, "Electricity demand forecasting methodologies and applications: A review," *Sustain. Energy Res.*, vol. 12, no. 19, 2025, doi: 10.1186/s40807-025-00149-z.
3. M. Alsap, "Forecasting electricity demand in the data-poor Indian context," M. Eng. thesis, Dept. Elect. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, 2020.
4. M. Chavan et al., "Time series analysis in electrical load forecasting," *Panamer. Math. J.*, vol. 35, no. 1s, 2025, doi: 10.52783/pmj.v35.1s.2307.
5. A. Perçuku, D. Minkovska, and N. Hinov, "Enhancing electricity load forecasting with machine learning and deep learning," *Technologies*, vol. 13, no. 2, Art. no. 59, 2025.
6. A. Jain and S. C. Gupta, "Evaluation of electrical load demand forecasting using various machine learning algorithms," *Front. Energy Res.*, vol. 12, Art. no. 1408119, 2024.

7. M. Abumohsen, A. Y. Owda, and M. Owda, “Electrical load forecasting using LSTM, GRU, and RNN algorithms,” *Energies*, vol. 16, no. 5, Art. no. 2283, 2023, doi: 10.3390/en16052283.
8. F. Bayram, P. Aupke, B. S. Ahmed, A. Kassler, A. Theocharis, and J. Forsman, “DA-LSTM: A dynamic drift-adaptive learning framework for interval load forecasting with LSTM networks,” *arXiv preprint* arXiv:2305.08767, May 2023.
9. B. Prabadevi et al., “Deep learning for intelligent demand response and smart grids: A comprehensive survey,” *arXiv preprint* arXiv:2101.08013, Jan. 2021.
10. C. Cebeci and K. Zor, “Electricity demand forecasting using deep polynomial neural networks and gene expression programming during COVID-19 pandemic,” *Appl. Sci.*, vol. 15, no. 5, Art. no. 2843, Mar. 2025, doi: 10.3390/app15052843.
11. M. Roy, V. Pyltsov, and Y. Hu, “IISE PG&E energy analytics challenge 2025: Hourly-binned regression models beat transformers in load forecasting,” *Preprint*, 2025.
12. F. Yaprakdal and M. V. Arısoy, “A multivariate time series analysis of electrical load forecasting based on a hybrid feature selection approach and explainable deep learning,” *Appl. Sci.*, vol. 13, no. 23, Art. no. 12946, Dec. 2023, doi: 10.3390/app132312946.
13. S. Mateo-Barcos, D. Ribó-Pérez, J. Rodríguez-García, and M. Alcázar-Ortega, “Forecasting electricity demand of municipalities through artificial neural networks and metered supply point classification,” *Energy Rep.*, vol. 11, pp. 3533–3549, 2024.
14. A. Bandyopadhyay et al., “Modelling and forecasting India’s electricity consumption using artificial neural networks,” *OPEC Energy Rev.*, Jan. 2024, doi: 10.1111/opec.12295.
15. A. Kychkin and G. Chasparis, “AI-powered predictions for electricity load in prosumer communities,” in *Proc. 18. Symp. Energieinnovation*, Graz, Austria, Feb. 2024, pp. 1–10.
16. C. Wang, H. Zhao, Y. Liu, and G. Fan, “Minute-level ultra-short-term power load forecasting based on time series data features,” *Appl. Energy*, vol. 372, Art. no. 123801, 2024.
17. A. Aziz, D. Mahmood, M. S. Qureshi, M. B. Qureshi, and K. Kim, “AI-based peak power demand forecasting model focusing on economic and climate features,” *Front. Energy Res.*, vol. 12, Art. no. 1328891, Jul. 2024, doi: 10.3389/fenrg.2024.1328891.
18. S. Gupta and P. Chitkara, “Day ahead price forecasting models in thin electricity market,” in *Proc. 2017 IEEE Int. Conf.*, 2017, pp. 1–6.
19. M. Barbar, D. S. Mallapragada, M. Alsup, and R. Stoner, “Scenarios of future Indian electricity demand accounting for space cooling and electric vehicle adoption,” *Sci. Data*, vol. 8, no. 1, Art. no. 178, 2021.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

