W-LSTMix: A Hybrid Modular Forecasting Framework for Trend and Pattern Learning in Short-Term Load Forecasting

Shivam Dwivedi [*1] **Anuj Kumar** [*1] **Harish Kumar Saravanan** [1] **Pandarasamy Arjunan** [1]

Abstract

Recently, there has been growing research interest in developing domain-specific Time Series Foundation Models (TSFMs), particularly in areas such as energy. We present **W-LSTMix**—a modular, lightweight hybrid architecture with only $\sim \mathbf{0.13\,M}$ parameters—designed to generalize across diverse building types for efficient building-level load forecasting. **W-LSTMix** integrates wavelet-based signal decomposition, neural basis expansion, and patch-based temporal mixing. It forecasts decomposed time series components separately: long-term trends via an LSTM-enhanced N-BEATS stack, and residual plus seasonal patterns through an MLP-Mixer-augmented N-BEATS structure. We benchmark **W-LSTMix** against state-of-the-art TSFMs such as Lag-Llama, Moirai, Chronos, and Tiny Time Mixers, as well as N-BEATS, under both zero-shot and fine-tuned settings. Our model consistently outperforms these baselines, demonstrating robust generalization and suitability for real-world intelligent energy management.

1. Introduction

Over the past two decades, rising global electricity demand—driven by economic growth and workforce expansion—has underscored the need for intelligent energy management. Accurate short-term load forecasting (STLF) is essential for intelligent building-level energy management, reliable microgrid operation, and the integration of renewable resources into power systems. However, conventional approaches such as ARIMA and traditional machine-learning algorithms often fail to scale or generalize across diverse buildings, due to the highly variable and nonlinear nature of building's energy consumption (Mohamed et al., 2010).

[*]Equal contribution [1]Robert Bosch Centre for Cyber Physical Systems, Indian Institute of Science, Bangalore, India. Correspondence to: Pandarasamy Arjunan <samy@iisc.ac.in>.

Proceedings of the 1st ICML Workshop on Foundation Models for Structured Data, Vancouver, Canada. 2025. Copyright 2025 by the author(s).

Foundation models have recently gained traction in load forecasting, achieving state-of-the-art results (Kumar et al., 2025; Saravanan et al., 2024). Models such as Tiny Time Mixers (TTMs) (Ekambaram et al., 2024a), Lag-Llama (Rasul et al., 2024), Moirai (Woo et al., 2024), and Chronos (Ansari et al., 2024) demonstrate strong generalization, but their large transformer-based architectures hinder deployment on edge or resource-constrained systems, and their full applicability to STLF remains underexplored.

To address these limitations, we propose **W-LSTMix**, a lightweight model that integrates wavelet-based signal decomposition, combines the ensemble power of N-BEATS (Oreshkin et al., 2020) with the gated memory of LSTM (Vennerød et al., 2021), and leverages the patch-mixing efficiency of MLP-Mixer (Chen et al., 2023b).

Key contributions: (a) **W-LSTMix**: A lightweight model for STLF in smart buildings, designed for efficiency and edge deployment; (b) **Large-scale Pretraining**: Trained on 0.81 billion real-world hourly electricity readings from 38,956 buildings worldwide; and (c) **Comprehensive Benchmarking**: Evaluated W-LSTMix in zero-shot and fine-tuned settings against TSFMs (TTMs, Chronos, Lag-Llama, Moirai) and N-BEATS on 1,000 unseen buildings.

2. Methodology

2.1. Problem Statement

We consider the task of STLF at *building-level energy forecasting*, where the objective is to predict future energy consumption values based on past observations. Formally, given a historical univariate time series observations $\mathbf{x}_{1:L} = \{x_1, x_2, \ldots, x_L\} \in \mathbb{R}^L$, the goal is to predict the next H values $\mathbf{y}_{L+1:L+H} = \{y_{L+1}, y_{L+2}, \ldots, y_{L+H}\} \in \mathbb{R}^H$. We formulate this as a *point forecasting* problem by directly learning a mapping using a parametric function $f_\theta : \mathbb{R}^L \to \mathbb{R}^H$, such that:

$$\hat{\mathbf{y}}_{L+1:L+H} = f_\theta(\mathbf{x}_{1:L}),$$

where $\hat{\mathbf{y}}_{L+1:L+H}$ represents the predicted energy values over the forecast horizon. Performance is evaluated using the Normalized Root Mean Squared Error (**NRMSE**); see Appendix E for details..

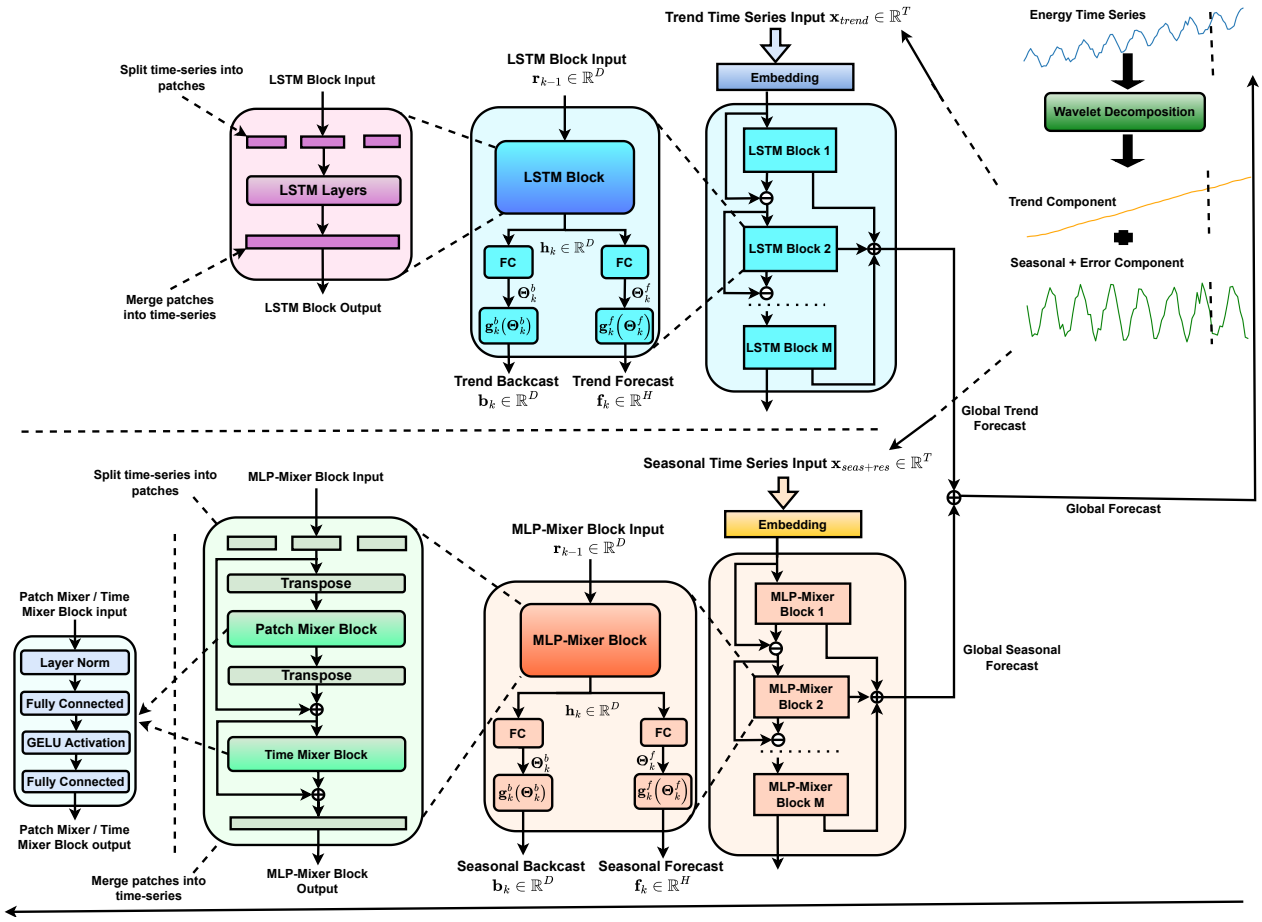


Figure 1. W-LSTMix Model Architecture

2.2. W-LSTMix

Our proposed model, **W-LSTMix**, leverages a hybrid, modular architecture that integrates signal decomposition, neural basis expansion, and patch-based time-mixing to enhance time series forecasting (see Figure 1). In the preprocessing stage, we adopt a decomposition strategy inspired by DLinear (Zeng et al., 2022), which separates the input series into trend and residual components. This decomposition allows W-LSTMix to handle each component individually, improving the learning of distinct temporal characteristics. The forecasting module builds upon the N-BEATS block architecture (Oreshkin et al., 2020), with basis expansion structured into stacks via the doubly residual stacking principle. For modeling long-term stationary trends, we extend the N-BEATS block by incorporating stacked LSTM (Vennerød et al., 2021) layers. To capture recurring patterns over fixed intervals (e.g. yearly or monthly cycles) and residual components, we enhance each N-BEATS block with stacks of the Patch-Time mixing mechanism from MLP-Mixer (Chen et al., 2023a), enabling rich patch-wise temporal modeling and stronger short-term pattern extraction.

Input and Output: The model receives a backcast window $\mathbf{x} \in \mathbb{R}^T$ and aims to predict a forecast window $\mathbf{F} \in \mathbb{R}^H$, where T and H are the input and output series lengths, respectively.

Decomposition of Time Series: We employ the Discrete Wavelet Transform (DWT) (**Appendix D & F**), specifically using the Daubechies 4 (db4) wavelet, to decompose a given univariate time series $\mathbf{x} \in \mathbb{R}^T$:

$$\mathbf{x} = A_n(t) + \sum_{i=1}^{n} D_i(t) \tag{1}$$

Here, $A_n(t)$ denotes the approximation (low-frequency) component at level n, and $D_i(t)$ represents the detail (high-frequency) component at level i.

The trend component is extracted via inverse DWT on $A_n(t)$:

$$\mathbf{x}_{\text{trend}} = \text{IDWT}(A_n(t)), \quad \mathbf{x}_{\text{trend}} \in \mathbb{R}^T \tag{2}$$

The seasonal plus residual components are given by:

$$\mathbf{x}_{\text{seasonal+residual}} = \mathbf{x} - \mathbf{x}_{\text{trend}}, \quad \in \mathbb{R}^T \tag{3}$$

This decomposition allows the model to separately process long-term trends and short-term fluctuations.

2.2.1. Architecture Overview

The model comprises two stacks, each containing M blocks. The first stack is designed to model long-term stationary trends, while the second stack focuses on capturing the seasonal and residual components. Each block learns to approximate the input signal and contributes incrementally to the final forecast.

Embedding: To retain the inherent sequential structure, we use a feed-forward layer with dropout as the embedding mechanism. Since this transformation operates independently at each time step, it implicitly preserves temporal order without requiring explicit positional encoding. Embedding: $\mathbb{R}^T \to \mathbb{R}^D$

$$\mathbf{x}_{\text{embed}} = \text{Embedding}(\mathbf{x}), \quad \mathbf{x}_{\text{embed}} \in \mathbb{R}^D \tag{4}$$

where D is the model dimension.

LSTM Block: Each LSTM block processes the input sequence in non-overlapping patches using a stack of LSTM layers and produces an output of the same dimensionality.

- The residual input from the previous block, $\mathbf{r}_{k-1} \in \mathbb{R}^D$, is divided into P patches of length L, forming $\mathbf{Z}_k \in \mathbb{R}^{P\times L}$.
- **LSTM Processing:** An LSTM maps the input sequence $\mathbf{Z}_k = (z_1, \ldots, z_P)$ to an output sequence $\mathbf{Y}_k = (y_1, \ldots, y_P)$, where $z_t \in \mathbb{R}^L$. At each patch index $t \in \{1, \ldots, P\}$, each LSTM layer computes:

  $$\mathbf{y}_t = \text{LSTMLayer}(\mathbf{z}_t), \quad \mathbf{y}_t \in \mathbb{R}^L$$

  The output features $\mathbf{Y}_k \in \mathbb{R}^{P\times L}$ are taken from the last layer of the LSTM across all patches.
- The output $\mathbf{Y}_k$ is reshaped to recover the residual input form $\mathbf{h}_k \in \mathbb{R}^D$.

This design allows the model to capture long-term stationary patterns by leveraging the LSTM's gated memory mechanism to retain and propagate temporal information across patches.

MLP-Mixer Block: Each MLP-Mixer block applies two sequential MLP-based mixing operations: patch mixing and time mixing.

- The residual input $\mathbf{r}_{k-1} \in \mathbb{R}^D$ is first divided into P patches of length L as $\mathbf{z}_1 \in \mathbb{R}^{P\times L}$.
- **Patch Mixing:** The input is reshaped as $\text{Transpose}(\mathbf{z}_1) \in \mathbb{R}^{L\times P}$ and mixed along the patch dimension:

  $$\mathbf{z}_2 = \text{LayerNorm}(\text{Transpose}(\mathbf{z}_1))W_k^p \tag{5}$$

  $$\mathbf{z}_3 = \mathbf{z}_1 + \text{Transpose}(\text{GELU}(\mathbf{z}_2)W_k^{p'}) \tag{6}$$
- **Time Mixing:** Performed mixing along the temporal dimension.

  $$\mathbf{z}_4 = \text{LayerNorm}(\mathbf{z}_3)W_k^l \tag{7}$$

  $$\mathbf{z}_5 = \mathbf{z}_3 + \text{GELU}(\mathbf{z}_4)W_k^{l'} \tag{8}$$
- $\mathbf{z}_5$ is reshaped to recover the residual input form $\mathbf{h}_k \in \mathbb{R}^D$.

These stages efficiently model inter- and intra-patch dependencies using lightweight MLPs.

Backcast-Forecast Branches: Each LSTM and MLP-Mixer block produces a hidden state $\mathbf{h}_k$, linearly projected into backcast and forecast components.

$$\mathbf{\Theta}_k^b = \mathbf{W}_k^b \cdot \mathbf{h}_k \in \mathbb{R}^p, \quad \mathbf{\Theta}_k^f = \mathbf{W}_k^f \cdot \mathbf{h}_k \in \mathbb{R}^q$$

where $p, q \ll D$, and $\mathbf{W}_k^b$, $\mathbf{W}_k^f$ are learnable. Outputs are processed by g_k^b and g_k^f.

$$\mathbf{b}_k = g_k^b(\mathbf{\Theta}_k^b) \in \mathbb{R}^D, \quad \mathbf{f}_k = g_k^f(\mathbf{\Theta}_k^f) \in \mathbb{R}^H \tag{9}$$

Here, $\mathbf{b}_k$ aids residual learning by reconstructing the input, while $\mathbf{f}_k$ contributes to the forecast; both use linear functions.

Residual Learning: The model follows a residual update mechanism:

$$\mathbf{r}_k \leftarrow \mathbf{r}_{k-1} - \mathbf{b}_k; \quad \mathbf{F} \leftarrow \mathbf{F} + \mathbf{f}_k \tag{10}$$

This process iteratively traverses the two stacks and their constituent blocks, progressively refining the forecast at each stage

Loss Function: To jointly optimize trend and seasonal components, we use a dynamically weighted total loss:

$$\mathcal{L}_{\text{total}} = \alpha \cdot \mathcal{L}(\hat{y}_{\text{trend}}, y_{\text{trend}}) + \beta \cdot \mathcal{L}(\hat{y}_{\text{season}}, y_{\text{season}}) \tag{11}$$

where weights α and β are adaptively normalized by component losses:

$$\alpha = \frac{\mathcal{L}_{\text{season}}}{\mathcal{L}_{\text{trend}} + \mathcal{L}_{\text{season}}}, \quad \beta = \frac{\mathcal{L}_{\text{trend}}}{\mathcal{L}_{\text{trend}} + \mathcal{L}_{\text{season}}} \tag{12}$$

We employ the **Huber loss** $\mathcal{L}$ with $\delta = 1.0$, balancing sensitivity to small errors and robustness to outliers, ensuring balanced learning of trends and seasonal variations.

3. Experiments and Results

3.1. Datasets, Implementation and Evaluation Setup

In this study, we curated energy consumption datasets comprising 0.81 billion data points from 39,956 real-world buildings, spanning both commercial and residential types across

Table 1. **Comparison of the Performance of Zero-shot and Fine-tuned models using NRMSE across different datasets. A context length of 168 (7 days) and a forecast horizon of 24 (1 day) are used. (Bold - Best, Underline - 2nd Best)**

DATASET	ZERO-SHOT						FINE-TUNED		
	MOIRAI	CHRONOS	LAG-LLAMA	TTMS	NBEATS	**W-LSTMIX**	MOIRAI	LAG-LLAMA	**W-LSTMIX**
COMMERCIAL BUILDINGS									
ENERNOC	29.20	25.99	51.68	23.21	28.60	18.13	24.68	27.98	17.03
IBLEND	27.30	16.17	63.16	21.51	23.00	13.85	22.00	18.44	13.47
AVERAGE	28.25	<u>21.08</u>	57.42	22.36	25.80	**15.99**	23.34	<u>23.21</u>	**15.25**
RESIDENTIAL BUILDINGS									
MATHURA	109.60	102.78	113.49	136.06	141.72	98.43	91.46	99.34	102.23
BAREILLY	69.90	73.84	89.88	64.42	65.92	57.18	54.41	69.27	50.74
MFRED	27.10	25.39	61.97	24.74	26.46	21.11	20.80	30.81	20.06
NEEA	80.40	82.64	91.93	69.31	74.51	66.60	66.81	84.27	62.99
NEST	71.70	71.33	85.00	64.30	64.95	61.37	54.98	66.90	60.26
PRAYAS	90.40	87.93	100.47	101.97	106.85	96.59	56.83	78.03	91.49
SMART*	65.90	69.69	83.61	61.54	62.75	56.10	65.14	89.12	65.03
IRELAND	86.50	92.32	115.63	81.52	83.45	73.45	70.01	82.84	70.93
GOIENER	111.62	114.59	131.23	117.69	120.99	112.08	99.34	111.07	108.49
SGSC	92.01	99.43	111.91	92.90	94.98	81.36	85.24	89.23	74.01
AVERAGE	<u>80.51</u>	81.99	98.51	81.45	84.26	**72.43**	**66.50**	80.09	<u>70.63</u>

multiple countries and time periods (See Appendix A). To ensure an unbiased evaluation, we excluded public datasets previously used for pre-training in existing TSFM models. W-LSTMix, TTMs and N-BEATS are trained exclusively on our curated dataset. Test buildings are strictly held out from pretraining to prevent any data leakage (Appendix G).

All experiments were run in `JupyterLab` using open-source libraries for time series forecasting, including `GluonTS`[1], `AutoGluon`[2] for TSFMs. W-LSTMix (two stacks of three LSTM and three MLP-Mixer blocks; patch size = 8; hidden = 256) is pre-trained on energy data from 38,956 buildings using an 8-day (192-hour) sliding window with a 1-day stride, where each sample includes a 168-hour context to forecast the next 24 hours for up to 100 epochs with early stopping on a RTX 5090 setup. Optimal parameters are chosen through extensive tuning. For evaluation, zero-shot forecasting is performed on 1,000 unseen commercial and residential buildings. Fine-tuning is performed individually for each building. Specifically, the first 6 months of data from each unseen building are used for fine-tuning, and the subsequent months are held out for testing, demonstrating W-LSTMix strong generalization. Reproducible code can be found at https://github.com/AI-IoT-Lab/W-LSTMix.

3.2. Zero-shot and Fine-tuned Forecasting Results

We first evaluate all pre-trained models in a zero-shot setting, forecasting 24 hours ahead on diverse commercial and residential buildings without task-specific fine-tuning. As shown in Table 1, W-LSTMix consistently outperforms state-of-the-art TSFMs (Moirai, Chronos, Lag-Llama, and TTM) and N-BEATS across both building types, demonstrating strong generalization.

[1] `https://ts.gluon.ai/stable/index.html`
[2] `https://auto.gluon.ai/`

Next, in the fine-tuned setting, we compare W-LSTMix to fine-tuned Moirai and Lag-Llama. Table 1 shows fine-tuning yields substantial gains, especially on residential data. W-LSTMix remains best on commercial loads, while fine-tuned Lag-Llama and Moirai surpass it in some residential cases—though W-LSTMix stays competitive. These results highlight the value of domain adaptation for improving foundation-model performance in complex, data-rich energy applications. Predicted vs actual load profiles for commercial buildings are visualized in Appendix C.

4. Conclusion and Future Work

We proposed **W-LSTMix**, a lightweight, hybrid modular architecture for short-term load forecasting (STLF) in smart buildings. The model demonstrated superior performance on both commercial and residential buildings. In the zero-shot setting, W-LSTMix achieved the lowest NRMSE, outperforming other models, including Chronos and TTM. Under fine-tuning, W-LSTMix remained best on commercial buildings, while Moirai surpassed it on residential buildings, achieving the lowest overall NRMSE, with W-LSTMix ranking second. These results highlight the significant benefits of fine-tuning, especially for residential forecasting. Beyond accuracy, W-LSTMix offers strong practical efficiency, with a mean inference time of **0.18s** and a compact size of **0.13M** parameters—making it well-suited for real-time and edge deployments. A key limitation is the model's focus on a 24-hour forecasting horizon without incorporating exogenous covariates. Future work will extend W-LSTMix to longer forecasting horizons, integrate covariate inputs for enhanced contextual understanding, and generalize across more diverse domains. We also aim to adapt W-LSTMix for other downstream tasks such as classification and anomaly detection, moving towards a general-purpose, domain-adaptive foundation model for time series forecasting.

References

Agrawal, S., Mani, S., Ganesan, K., and Jain, A. High frequency smart meter data from two districts in India (Mathura and Bareilly), 2021. URL https://doi.org/10.7910/DVN/GOCHJH.

Ansari, A. F., Stella, L., Turkmen, C., Zhang, X., Mercado, P., Shen, H., Shchur, O., Rangapuram, S. S., Pineda Arango, S., Kapoor, S., Zschiegner, J., Maddix, D. C., Wang, H., Mahoney, M. W., Torkkola, K., Gordon Wilson, A., Bohlke-Schneider, M., and Wang, Y. Chronos: Learning the language of time series, 2024.

Chen, S.-A., Li, C.-L., Yoder, N., Arik, S. O., and Pfister, T. Tsmixer: An all-mlp architecture for time series forecasting, 2023a.

Chen, S.-A., Li, C.-L., Yoder, N., Arik, S. O., and Pfister, T. Tsmixer: An all-mlp architecture for time series forecasting, 2023b. URL https://arxiv.org/abs/2303.06053.

Ekambaram, V., Jati, A., Dayama, P., Mukherjee, S., Nguyen, N. H., Gifford, W. M., Reddy, C., and Kalagnanam, J. Tiny time mixers (ttms): Fast pre-trained models for enhanced zero/few-shot forecasting of multivariate time series, 2024a. URL https://arxiv.org/abs/2401.03955.

Ekambaram, V., Jati, A., Nguyen, N. H., Dayama, P., Reddy, C., Gifford, W. M., and Kalagnanam, J. Ttms: Fast multi-level tiny time mixers for improved zero-shot and few-shot forecasting of multivariate time series, 2024b.

Emami, P., Sahu, A., and Graf, P. Buildingsbench: A large-scale dataset of 900k buildings and benchmark for short-term load forecasting. *Advances in Neural Information Processing Systems*, 36:19823–19857, 2023.

Heer, P. Nest open building data for energy demand and user practice, Apr 2024. URL https://figshare.com/collections/NEST_Open_Building_Data_for_Energy_Demand_and_User_Practice/7178787/1.

Khadem, S., Trivedi, R., Bahloul, M., Saif, A., and Patra, S. Comprehensive dataset on electrical load profiles for energy community in ireland, Jun 2024. URL https://springernature.figshare.com/collections/Comprehensive_Dataset_on_Electrical_Load_Profiles_for_Energy_Community_in_Ireland/6829134/1.

Kumar, A., Saravanan, H. K., Dwivedi, S., and Arjunan, P. Mixforecast: Mixer-enhanced foundation model for load forecasting. In *Proceedings of the 2nd International Workshop on Foundation Models for Cyber-Physical Systems & Internet of Things*, pp. 25–30, 2025.

Meinrenken, C. J., Rauschkolb, N., Abrol, S., Chakrabarty, T., Decalf, V. C., Hidey, C., McKeown, K., Mehmani, A., Modi, V., and Culligan, P. J. MFRED (public file, 15/15 aggregate version): 10 second interval real and reactive power in 390 US apartments of varying size and vintage, 2020. URL https://doi.org/10.7910/DVN/X9MIDJ.

Mohamed, N., Ahmad, M. H., Ismail, Z., and Suhartono, S. Short term load forecasting using double seasonal arima model, 2010.

Oreshkin, B. N., Carpov, D., Chapados, N., and Bengio, Y. N-beats: Neural basis expansion analysis for interpretable time series forecasting, 2020. URL https://arxiv.org/abs/1905.10437.

Prayas, E. G. Processed data, 2021. URL https://doi.org/10.7910/DVN/YJ5SP1.

Quesada, C., Astigarraga, L., Merveille, C., and Borges, C. E. An electricity smart meter dataset of spanish households: insights into consumption patterns. *Scientific Data*, 11(1):59, 2024.

Rashid, H., Singh, P., and Singh, A. I-blend, a campus scale commercial and residential buildings electrical energy dataset, Feb 2019. URL https://springernature.figshare.com/collections/I-BLEND_a_campus_scale_commercial_and_residential_buildings_electrical_energy_dataset/3893581/1.

Rasul, K., Ashok, A., Williams, A. R., Ghonia, H., Bhagwatkar, R., Khorasani, A., Bayazi, M. J. D., Adamopoulos, G., Riachi, R., Hassen, N., Biloš, M., Garg, S., Schneider, A., Chapados, N., Drouin, A., Zantedeschi, V., Nevmyvaka, Y., and Rish, I. Lag-llama: Towards foundation models for probabilistic time series forecasting, 2024. URL https://arxiv.org/abs/2310.08278.

Saravanan, H. K., Dwivedi, S., Praveen, P., and Arjunan, P. Analyzing the performance of time series foundation models for short-term load forecasting. In *Proceedings of the 11th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, pp. 346–349, 2024.

Vennerød, C. B., Kjærran, A., and Bugge, E. S. Long short-term memory rnn, 2021. URL https://arxiv.org/abs/2105.06756.

Woo, G., Liu, C., Kumar, A., Xiong, C., Savarese, S., and Sahoo, D. Unified training of universal time series forecasting transformers, 2024.

Zeng, A., Chen, M., Zhang, L., and Xu, Q. Are transformers effective for time series forecasting?, 2022. URL https://arxiv.org/abs/2205.13504.

A. Dataset Description

This appendix presents a comprehensive overview of the datasets employed for training and evaluating our proposed model.

A.1. Commercial Building Datasets

A.1.1. EnerNOC

The EnerNOC dataset provides anonymized electricity usage measurements for 100 commercial and industrial facilities across the United States over the full calendar year of 2012. [3]

A.1.2. I-Blend

The I-Blend dataset offers minute-level energy consumption data spanning from 2013 to 2017 for an academic institution in Delhi, India. It covers seven buildings: Academic, Lecture, Library, Facilities, Dining, Boys' Dormitory, and Girls' Dormitory, as documented in (Rashid et al., 2019).

A.2. Residential Building Datasets

A.2.1. CEEW

This dataset includes 3-minute interval electricity consumption data collected from approximately 100 smart meters installed in residential areas of Bareilly and Mathura districts in Uttar Pradesh, India, during the period of May 2019 to October 2021 (Agrawal et al., 2021). The dataset is categorized by district.

A.2.2. Ireland

Collected as part of the StoreNet project (Khadem et al., 2024), this dataset provides 1-minute resolution smart meter data from residential households in Ireland. It contains various metrics such as active/reactive power, PV generation, import/export, and battery storage status (charging, discharging, and state of charge) for the year 2020.

A.2.3. MFRED

The Multifamily Residential Electricity Dataset (MFRED) captures electricity consumption data for 390 apartments in the Northeastern U.S., recorded at 10-second intervals throughout 2019 (Meinrenken et al., 2020). Apartments vary by size and HVAC systems. For privacy, the data is aggregated across groups of 15 units based on annual usage.

A.2.4. NEEA

The Northwest End Use Load Research (EULR) initiative, led by the Northwest Energy Efficiency Alliance, collected 15-minute interval power consumption data from about 150 homes. Although access is restricted, further details are available online. [4]

A.2.5. NEST

NEST (Next Evolution in Sustainable Building Technologies) (Heer, 2024) is a research platform that provides high-resolution energy data (1-minute interval) over four years from three buildings. It includes detailed measurements on electricity, heating/cooling, water usage, operational settings, and occupant behavior. Our study focuses on the residential building UMAR from this platform.

A.2.6. Prayas

This dataset comprises 15-minute interval readings from 115 households in Maharashtra and Uttar Pradesh, collected via eMARC meters. It spans January 2018 to June 2020 and includes only data from main line meters (Prayas, 2021).

[3] https://open-enernoc-data.s3.amazonaws.com/anon/index.html

[4] https://neea.org/data/nw-end-use-load-research-project/energy-metering-study-data

Table 2. Details of real building datasets used in this study (hourly data). (C: Commercial, R: Residential)

DATASET	LOCATION	# BUILDINGS	# OBS.	YEARS
IBLEND (C)	INDIA	9	296,357	2013–17
ENERNOC (C)	USA	100	877,728	2012
NEST (R)	SWITZERLAND	1	34,715	2019–23
IRELAND (R)	IRELAND	20	174,398	2020
MFRED (R)	USA	26	227,622	2019
CEEW (R)	INDIA	84	923,897	2019–21
SMART* (R)	USA	114	958,998	2016
PRAYAS (R)	INDIA	116	1,536,409	2018–20
NEEA (R)	USA	192	2,922,289	2018–20
SGSC (R)	AUSTRALIA	13,735	172,277,213	2011–14
GOIENER (R)	SPAIN	25,559	632,313,933	2014–22
TOTAL		**39,956**	**812,543,559**	

A.2.7. SMART*

SMART* includes whole-house power usage data from 114 single-family residences recorded at 1-minute intervals for the year 2016. [5]

A.2.8. SGSC (SMART GRID SMART CITY)

SGSC, an Australian government-backed project (2010–2014), offers half-hourly energy usage data linked with household demographics and appliance usage. The dataset spans roughly 13,735 buildings and includes variables related to climate, tariffs, and peak load events. [6]

A.2.9. GOIENER

The GoiEner dataset (Quesada et al., 2024) includes 7.2 GB of smart meter electricity data from Spanish households, collected between late 2014 and mid-2022. The data density increases significantly after January 2018 due to broader smart meter deployment. It represents usage across approximately 25,559 households, with the final month being incomplete.

B. Details of Time Series Foundation Models (TSFM)

Several recent time series foundation models (TSFMs) offer diverse architectural innovations for large-scale forecasting tasks. **Lag-Llama** (Rasul et al., 2024) is a decoder-only transformer model tailored for univariate probabilistic forecasting, with 2.45 million parameters and a context length of 1024, trained on over 1 billion observations. **Moirai** (Woo et al., 2024), developed by Salesforce AI Research, is an encoder-only architecture with 14 million parameters, pre-trained on the massive LOTSA dataset containing 27 billion time series points across nine domains. It supports a long context window of 5000, allowing broad temporal modeling capabilities. **Chronos** (Ansari et al., 2024), based on the T5 transformer family, follows an encoder-decoder design with 46 million parameters and a 512-length context, trained on 893K multivariate series for general-purpose sequence modeling. In contrast, **N-BEATS** (Oreshkin et al., 2020) is a fully MLP-based model designed for point forecasting, with 2.2 million parameters and an input length of 240, known for its interpretable block-wise architecture. **Tiny Time Mixers (TTMs)** (Ekambaram et al., 2024b) are lightweight MLP-based models that mix tokens and channels efficiently, using 1 million parameters and a 1536 context window to deliver strong performance with minimal compute. Our proposed **W-LSTMix** model is the most lightweight among all, with just 0.18 million parameters and a context length of 168. It integrates the interpretability of N-BEATS with the efficiency of TSMixer layers, enabling accurate point forecasting on the same 282K-series dataset used by N-BEATS and TTMs. Despite its compact size, W-LSTMix achieves competitive performance and is ideal for edge deployment in real-world smart building applications.

[5] https://traces.cs.umass.edu/docs/traces/smartstar/

[6] https://data.gov.au/data/dataset/smart-grid-smart-city-customer-trial-data

C. Visualization

We present forecast visualizations (see figure 2) on the Enernoc and I-BLEND datasets across various buildings, demonstrating that our model effectively learns complex time series structures.

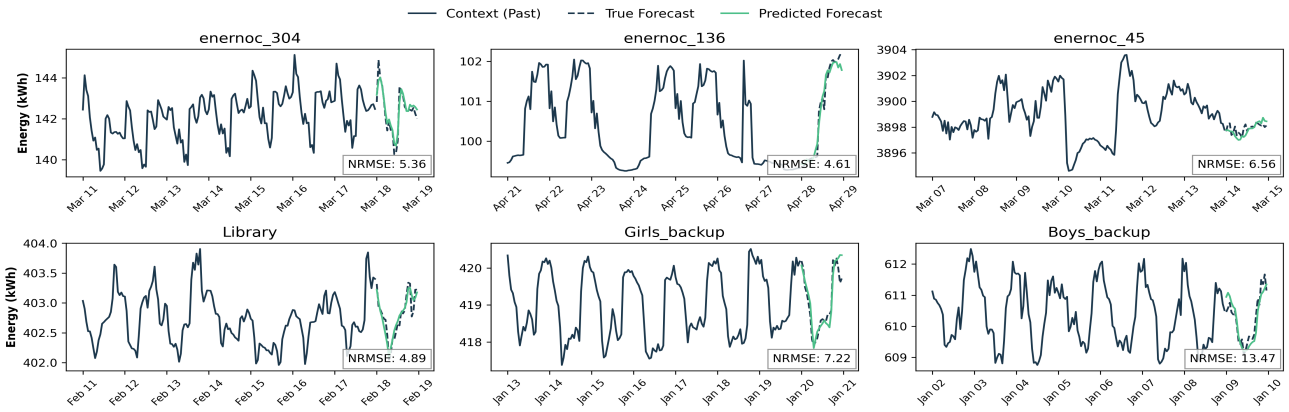


Figure 2. Load forecasting by W-LSTMix for subsequent windows for Commercial Building dataset.

D. Discrete Wavelet Transform

The Discrete Wavelet Transform (DWT) is a widely used signal decomposition technique, effective for denoising and compression in time series and images. Unlike the Fourier Transform, DWT captures both frequency and temporal localization, making it well-suited for time series analysis.

Given a signal $x[n]$, DWT decomposes it using a low-pass filter $g[n]$ and a high-pass filter $h[n]$:

$$y_{\text{low}}[n] = \sum_{k=-\infty}^{\infty} x[k]\, g[2n-k], \quad y_{\text{high}}[n] = \sum_{k=-\infty}^{\infty} x[k]\, h[2n-k]$$

The outputs are the approximation (low-frequency) and detail (high-frequency) coefficients. After filtering, the signal is downsampled by a factor of two (per Nyquist's rule). This process is recursively applied to the approximation coefficients to generate a multi-resolution decomposition, progressively capturing coarse and fine signal structures. The final reconstructed

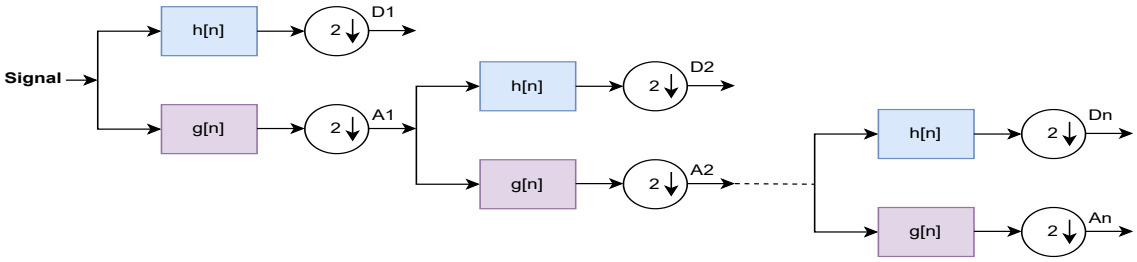


Figure 3. multi-level Discrete Wavelet Transform (DWT) decomposition

signal after n levels of decomposition is given by:

$$x(t) = A_n(t) + \sum_{i=1}^{n} D_i(t)$$

where $A_n(t)$ is the final approximation component and $D_i(t)$ are the detail components at each level.

E. Evaluation Metric

To evaluate forecasting accuracy, we use the Normalized Root Mean Square Error (NRMSE), a standard, scale-independent metric in energy load forecasting that quantifies the deviation between predicted and actual load profiles, as in Buildings-Bench (Emami et al., 2023). This measure is particularly suitable for assessing forecasting performance in applications such as energy consumption prediction in smart buildings. For a K-day horizon with 24-hour resolution, NRMSE is defined as:

$$NRMSE = 100 \times \frac{1}{\bar{x}} \sqrt{\frac{1}{24K} \sum_{k=1}^{K} \sum_{h=1}^{24} (x_{k,h} - \hat{x}_{k,h})^2} \tag{13}$$

where $x_{k,h}$ and $\hat{x}_{k,h}$ are the true and predicted loads at hour h on day k, and $\bar{x}$ is the mean of all true values. Normalizing by $\bar{x}$ enables comparison across buildings, and the factor 100 expresses the error as a percentage.

F. Comparison of Time Series Decomposition Methods

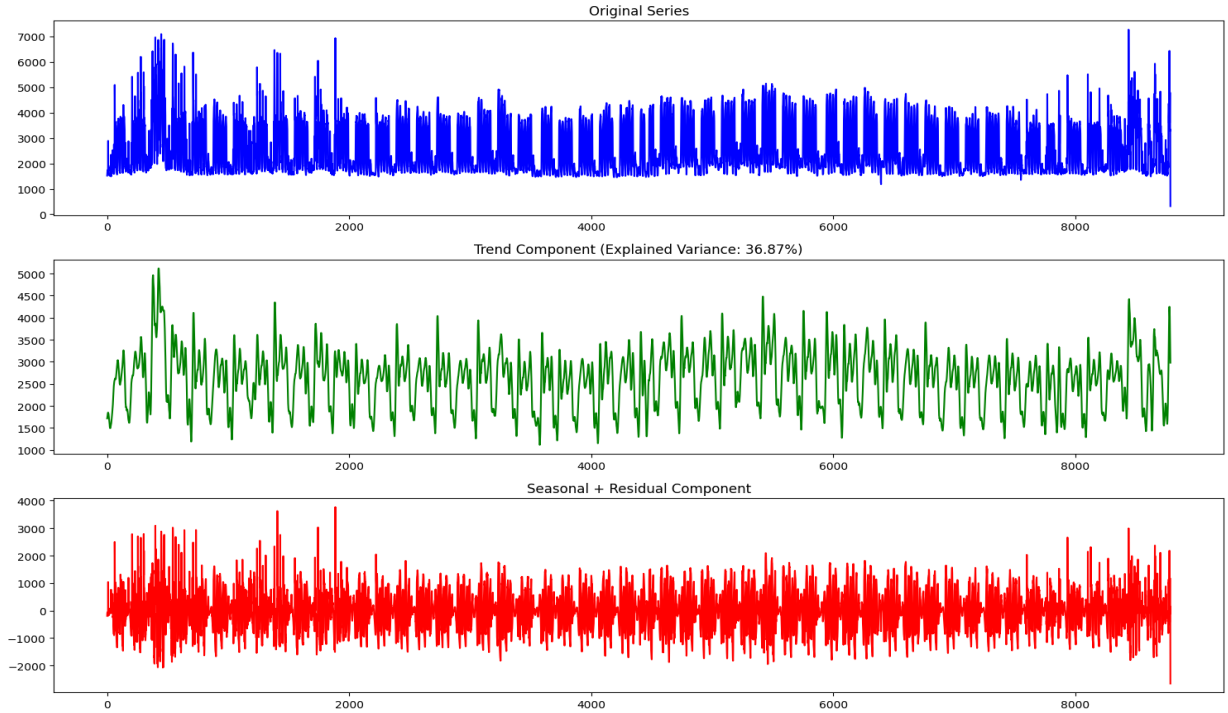
Table 3. Comparison of decomposition methods for time series forecasting.

Decomposition Method	Trend Block	Seasonal-Residual Block	Comm. Error ↓	Res. Error ↓
Additive	LSTM	MLP-Mixer	17.28	72.58
Wavelet	LSTM	MLP-Mixer	**15.99**	72.43

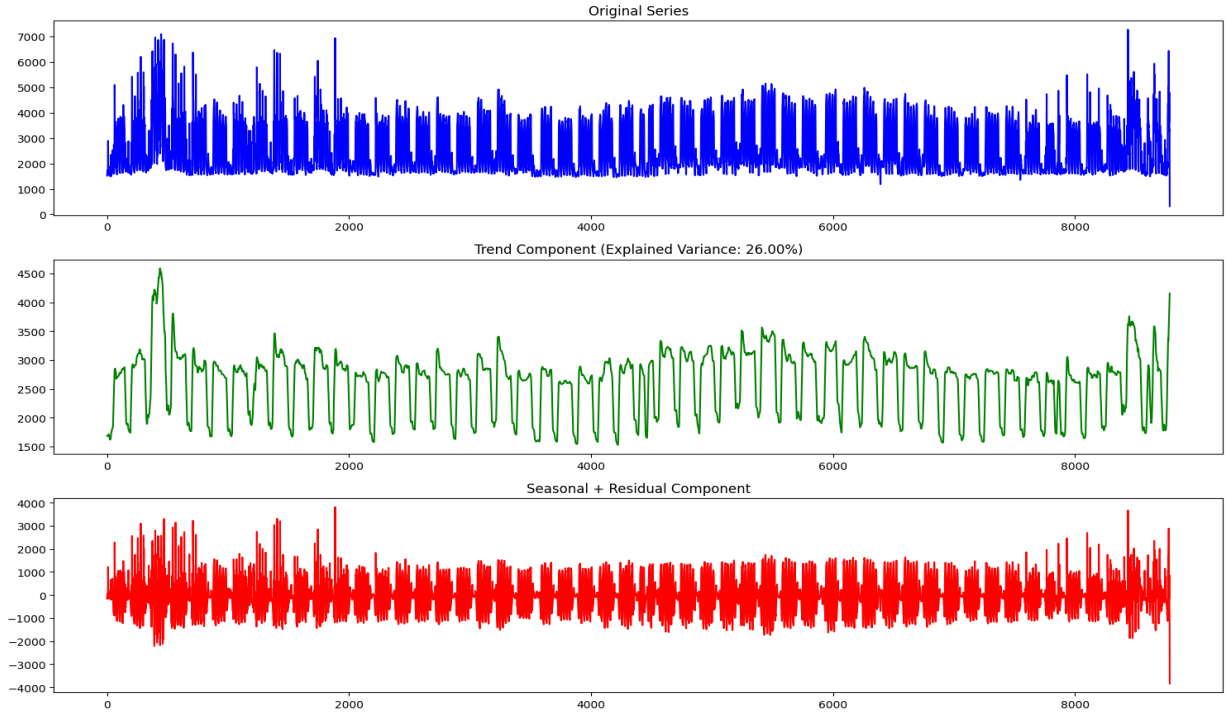
The results indicate that **Wavelet Decomposition** provides a lower average forecasting error for commercial buildings compared to Seasonal Decomposition. This improvement is attributed to the ability of wavelet transforms to capture a greater proportion of the variance in the low-frequency trend component of time series data.

Specifically, wavelets utilize scale and shift operations along with wave-like, decaying basis functions. This allows them to simultaneously localize signal features in both the time and frequency domains, making them especially well-suited for extracting slowly varying trend components. As a result, the trend block trained on wavelet-transformed data is better able to forecast long-term temporal structures.

See Fig. 4 for a comparative variance analysis of trend components of enernoc_14 building.



(a) Wavelet Decomposition of enernoc_14: Trend variance explained **36.87%**



(b) Additive Decomposition of enernoc_14: Trend variance explained **26.00%**

Figure 4. (a) Visualization of time series decomposition using wavelet transform. (b) Visualization of time series decomposition using additive decomposition.

G. Pre-training Details:

Our curated dataset (Appendix A) comprises diverse geographically regions, each containing multiple buildings, totaling 39,956. From each region, 38,956 buildings are used for model pretraining, and the remaining 1000 buildings, entirely unseen during training, are used exclusively for evaluation. This ensures no overlap at the building level in time series data between training and testing sets, preventing data leakage.

From-Scratch Models (TFS): Models such as N-BEATS and Tiny Time Mixers (TTM) are trained from scratch using the same pre-training and testing split as W-LSTMix. These models are subject to the same data availability, regional splits, and constraints, ensuring a fully equitable comparison.

Pre-trained TSFMs: Models like Lag-Llama, Chronos, and Moirai are leveraged in their publicly released pre-trained forms, typically trained on large-scale public or proprietary datasets different from ours.

For fine-tuning evaluations, all models are adapted at the building level, where the first 6 months of each unseen test building are used for fine-tuning, and the remaining period is used for testing. This simulates a realistic deployment scenario with limited adaptation data.

Acknowledgments

This work was supported by a research grant from the AI & Robotics Technology Park (ARTPARK) at the Indian Institute of Science. The authors gratefully acknowledge the dataset providers for their efforts in making the data publicly available and maintaining high-quality resources for this study.