PMU-driven Feeder-level Load Forecasting Using Large Language Model

Haoyu Liu, Zhe Gao, Feifei Bai
School of Electrical Engineering and Computer Science
The University of Queensland
Brisbane, Australia
uqhliu38@uq.edu.au

Abstract—Load forecasting is essential for core power-system operations such as Distributed Energy Resource (DER) scheduling, power-purchase planning, and congestion management. While a lot of recent work has focused on household-level and bulk-system forecasting, distribution-feeder-level forecasting is increasingly critical due to high renewable penetration, behind-the-meter photovoltaic (PV), and rapid load diversity at the grid edge. However, existing approaches struggle to (i) fuse heterogeneous high-frequency measurements with exogenous context (weather, calendar), (ii) generalize across feeders without expensive retraining, and (iii) remain lightweight enough for edge deployment. To address such gaps, a phasor measurement unit (PMU)-driven feeder-level forecasting framework that tokenizes inputs including hourly-aggregated PMU phasors, spectral descriptors, and external features is developed. These features are assembled into a sequence processed by a pre-trained large language model (LLM). The approach leverages long-context priors and parameter-efficient tuning to achieve strong few-/zero-shot performance. The proposed model also achieves a 40.1% relative reduction in Mean Absolute Percentage Error (MAPE) and a 43.7% relative reduction in Normalized Root Mean Square Error (NRMSE) compared with existing methods, while still maintaining high accuracy during edge deployment on a Jetson device with low overhead.

Keywords—Feeder-level load forecasting, PMU, large language model, parameter-efficient tuning, edge intelligence.

I. INTRODUCTION

Ultra-short-term load forecasting (USTLF) with horizons on the order of minutes is of great significance to modern power system operation [1]. Unlike long-term planning or even day-ahead scheduling, USTLF directly informs real-time control actions where operators must continuously balance supply and demand under rapid demand fluctuations and uncertain DER behavior. At this granularity, even small forecasting errors can propagate into frequency deviations, voltage instabilities, or inefficient dispatch, translating into direct reliability and economic risks [2]. There are several reasons why USTLF is critical: Accurate USTLF allows system operators to react quickly when load changes occur. It supports faster Automatic Generation Control and reserve activation, which helps keep system frequency stable. It also reduces the risk of transformer or feeder overloads by enabling early Volt/VAR control and fast demand response before equipment limits are reached. In addition, high-resolution forecasts improve visibility of short transients and sudden load jumps, allowing operators to take preventive actions rather than waiting for alarms. This capability is increasingly important as rooftop PV and Electrical Vehicle (EV) charging introduce rapid fluctuations: net load can swing sharply within just a few minutes when clouds pass or when clusters of EVs begin charging. By reducing the need for emergency interventions and improving situational awareness, USTLF helps maintain a more stable grid while also lowering real-time balancing costs.

Electrical load forecasting has been extensively studied across multiple methodological paradigms. Early work focused on statistical methods, including autoregressive integrated moving average (ARIMA), seasonal ARIMA, exponential smoothing, and state-space models [3]. These approaches are computationally efficient and interpretable, but they rely heavily on stationarity assumptions, meaning they expect load patterns to remain statistically consistent over time. However, real feeder-level demand often changes with DER adoption, weather shifts, or customer behavior, breaking these assumptions and degrading model performance. As a result, these statistical approaches struggle to capture nonlinear and rapidly changing load dynamics, particularly during sharp ramps, special events, or disturbances.

Machine learning enabled more flexible representations of nonlinear dynamics. Support Vector Regression [4] and tree ensembles such as Random Forests [5] and Gradient Boosting (XGBoost [6], LightGBM) have demonstrated strong performance when sufficient historical data and engineered features are available [7]. These methods improve upon statistical models by capturing feature interactions, but they are still dependent on extensive lag-based and calendar feature engineering. Moreover, they have limited capacity for long temporal dependencies and do not natively model spatio-temporal correlations among feeders or buildings, which are increasingly important under DER proliferation.

To overcome these limitations, deep learning approaches have been widely adopted. Recurrent Neural Networks, particularly Long Short-Term Memory and Gated Recurrent Units [8, 9], have been shown to capture temporal

979-8-3315-8138-1

dependencies effectively, improving short-term residential and feeder-level forecasting accuracy. Temporal Convolutional Networks [10] and one-dimensional convolutional neural networks [11] have also been used to detect local motifs in demand sequences. More recently, spatio-temporal graph deep learning has emerged, with models such as the Spatio-Temporal Graph Convolutional Network (STGCN) [12] and Graph WaveNet (GW) [13] exploiting the spatial structure of distribution systems alongside temporal load evolution. Transformer-based architectures further extended forecasting horizons, with models like Autoformer [14] and PatchTST [15] achieving state-of-the-art performance in long-horizon multivariate time-series forecasting. Despite these advances, deep learning methods remain highly data-intensive, often struggle under domain shift (e.g., feeder-level DER adoption patterns).

A new line of research explores the use of pre-trained large language models (LLMs) for time-series forecasting. [16] recently proposed *BlackInter*, a black-box tuning inductive adapter that empowers pre-trained LLMs for building-level load forecasting. BlackInter leverages the generalization capacity of frozen decoder-only LLMs (e.g., LLaMA-7B [17]), with input embeddings augmented by spatial neighbor aggregation and dynamic mode decomposition features. It optimizes only lightweight adapters via a Mix-Adam strategy that avoids full back-propagation, thereby reducing memory costs and enabling few-shot and zero-shot transfer across different building populations. Experimental results demonstrated significant improvements when historical data is scarce, validating the potential of LLMs for inductive load forecasting. Nevertheless, notable gaps remain. Current LLM-based studies are restricted to building-level demand using advanced metering infrastructure data, without incorporating physics-aware features from PMUs. The tokenization strategy focuses on generic patch embeddings and does not explicitly encode temporal operations, spectral descriptors, or feeder-specific physical constraints. Furthermore, most LLM-based forecasters have not addressed edge deployment feasibility, where memory-efficient quantization and compact sequence design are essential for real-time feeder-level applications.

To address these gaps, a PMU-driven, LLM-based feeder forecaster is proposed. The contributions include: (i) tokenizes heterogeneous inputs including hourly PMU aggregates, spectral descriptors, and exogenous features into a unified time-encoded sequence, (ii) adapts pre-trained decoder-only LLMs via frozen backbones or LoRA adapters to support robust few-/zero-shot transfer across feeders, and (iii) supports efficient edge deployment through compact token design and weight-only quantization. This enables accurate and edge-ready load forecasting.

II. Methodology

This section shows the overall workflow, and then describes features, sequence assembly and tokenization, backbone and adaptation, learning objectives and complexity/edge deployment. The workflow is shown in Fig. 1. Heterogeneous inputs including aggregated PMU phasors, spectral disturbance descriptors and exogenous features are encoded into a unified token sequence and processed by a frozen or LoRA-adapted LLM backbone for 5-minute-ahead prediction. This workflow is deployed across feeder subsystems, where each local area (represented by PMU-enabled recloser sites) is equipped with synchrophasor sensing and edge computing capability. The trained model is executed on the edge device to provide real-time feeder-level predictions.

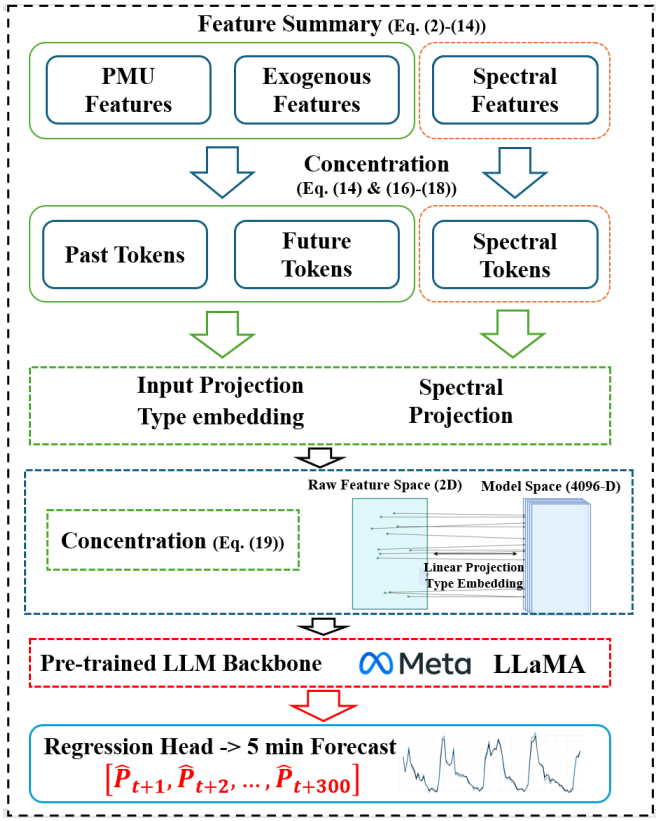

Fig. 1 Overall Workflow

A. Notation and Setup

Let s index seconds. The past window is $W_t = \{t - L + 1, \dots, t\}$ with $L = 900$ and the horizon is $\{t + 1, \dots, t + H\}$ with $H = 300$. Raw PMU channels are sampled at 50 Hz. They are aggregated to 1 Hz by averaging all samples with each $[s, s + 1)$ as shown in (1):

$$\bar{x}[s] = \frac{1}{|\mathcal{B}_s|} \sum_{n \in \mathcal{B}_s} x[n], \qquad \mathcal{B}_s = \{n : n \in [s, s + 1)\} \tag{1}$$

B. PMU-Derived Features on the 15min window

From per-phase phasors at 1 Hz at index s, the per-sample active power is calculated as (2) and (3):

$$\bar{P}_\phi[s] = \overline{U_\phi^{MAG}[s] I_\phi^{MAG}[s] \cos\left(\angle U_\phi[s] - \angle I_\phi[s]\right)} \tag{2}$$

$$\bar{P}[s] = \sum_{\phi \in \{A,B,C\}} \bar{P}_\phi[s] \tag{3}$$

Within the past window W_t, the second-level statistics used as features are computed in (4)(5)(6).:

$$P_{W_t}^{mean} = mean_{s\in W_t}\bar{P}[s] \quad (4)$$

$$P_{W_t}^{max} = max_{s\in W_t}\bar{P}[s] \quad (5)$$

$$P_{W_t}^{min} = min_{s\in W_t}\bar{P}[s] \quad (6)$$

The three phase voltage and current unbalance percentages are derived as shown in (7) and (8):

$$V_{u,W_s}\% = \frac{max_\phi \overline{U_\phi^{MAG}} - min_\phi \overline{U_\phi^{MAG}}}{\frac{1}{3}\sum_\phi \overline{U_\phi^{MAG}}} \times 100\% \quad (7)$$

$$I_{u,W_s}\% = \frac{max_\phi \overline{I_\phi^{MAG}} - min_\phi \overline{I_\phi^{MAG}}}{\frac{1}{3}\sum_\phi \overline{I_\phi^{MAG}}} \times 100\% \quad (8)$$

Denote ROCOF as $\overline{\dot{f}(s)}$. The within-hour intensity is derived by a robust quantile and exceedance count as shown in (9) and (10):

$$\dot{f}_{W_t}^{p95} = q_{0.95}\left(\left|\overline{\dot{f}(s)}\right|\right)_{s\in W_t} \quad (9)$$

$$\varepsilon_{W_t}^{\dot{f}}(0.5) = \sum_{n\in\mathcal{H}_t} 1\left\{\left|\dot{f}(n)\right| > 0.5\frac{Hz}{s}\right\} \quad (10)$$

Then the corresponding hourly token is shown in (11):

$$ROCOFToken_t = \left[\dot{f}_{W_t}^{p95}, \varepsilon_{W_t}^{\dot{f}}(0.5)\right] \quad (11)$$

A spectral fingerprint of fast disturbances is derived. Let $y[s] = \bar{P}[s] - \overline{\bar{P}}$ where $s \in W_t$, and $Y[k]$ be the rFFT magnitudes with $Y[0] = 0$. The top-1 peak $(\tilde{A}_1, f_1)$ and band-energy ratios (slow/low/high) are taken as shown in (12) and (13):

$$E_b = \frac{\sum_{k\in b}|Y[k]|}{\sum_{k>0}|Y[k]|}, b \in [0.02,0.2], [0.2,0.5] \quad (12)$$

$$f_{centroid} = \frac{\sum_{k>0} f_k|Y[k]|}{\sum_{k>0}|Y[k]|}, \quad f_k = \frac{k}{L}\,Hz \quad (13)$$

Then the high-frequency based spectral token is shown in (14):

$$SpecToken_{W_t}^{hi} = \left[\tilde{A}_1, f_1, E_{0.02-2}, E_{0.2-2}, f_{centroid}\right] \quad (14)$$

C. Exogenous Features

Weather features include ambient temperature T_t, relative humidity H_t. Calendar encodings use sinusoidal hour-of-day and month-of-year, plus binary flags for weekend, holiday, and school terms.

D. Input Vectors and Tokenization

For each second s in the window and horizon, a feature vector x_s that combines aggregated PMU statistics, high-frequency disturbance tokens, and exogenous context is derived as shown in (16):

$$x_s = \left[P_{W_s}^{min}, V_{u,W_s}\%, \dots, SpecToken_{W_s}^{hi}, weather, \dots\right] \quad (16)$$

The **past** tokens carry the $observed$target and exogenous features as shown in (17):

$$z_s^{past} = [\bar{P}[s], x_s], \qquad s \in W_t \quad (17)$$

Let PMU-derived features be x_{pmu}. The future tokens expose only exogenous features with a zero placeholder for the (unknown) target as shown in (18):

$$z_s^{fut} = \left[0, x_s \backslash x_{pmu}\right], s = t+1, \dots, t+H \quad (18)$$

All continuous features are standardized (fit on the training split). A single **GlobalSpecToken** computed from the entire 15 min history is prepended as a global descriptor. The final input sequence is shown in (19):

$$Input = \left[GlobalSpecToken; z_{t-L+1:t}^{past}; z_{t+1:t+H}^{fut}\right] \quad (19)$$

which is followed by a learnable linear projection to the LLM embedding space with type embeddings (past/future).

E. Backbone and Parameter-Efficient Adaptation

A decoder-only pre-trained LLM is used (e.g., GPT-/LLaMA-family). Two regimes are compared: (i) *Frozen* backbone (only the input/output projections and the regression head are trained), and (ii) *LoRA*-adapted backbone with low-rank updates inserted into attention/MLP projections. LoRA adds $O(rd)$parameters per adapted matrix (rank $r \ll d$), enabling few-shot feeder-specific adaptation under tight compute budgets.

F. Training Objectives

Let $\hat{\bar{P}}[s]$ be the model outputs at the future token positions. The primary loss is shown in (20):

$$\mathcal{L}_{MSE} = \frac{1}{H}\sum_{h=1}^{H}\left(\bar{P}[t+h] - \hat{\bar{P}}[t+h]\right)^2 \quad (20)$$

G. Complexity and Edge Deployment

The token length is $N = L + H + 1$. Transformer layers scale as $O(N^2 d)$; this is tractable with a compact backbone and parameter-efficient tuning. For edge deployment, weight-only int8 quantization is applied to the frozen backbone while keeping the projection/head (and LoRA adapters if present) in fp16/fp32, preserving accuracy with low memory/latency. When tighter budgets are required, a strided past (e.g., 2–5 s stride for a subset of history) can reduce N while keeping the $H = 300$ future seconds at full 1 Hz resolution.

III. Case Study

A. Setup and Data

The database is built from real PMU measurements collected by 50 Power Reclosers with embedded PMU functionalities installed across distribution networks in Queensland, Australia as shown in Fig. 2. From this database, three representative PMUs are selected: two are used for model training, while the remaining one is reserved for zero-shot and few-shot evaluation.

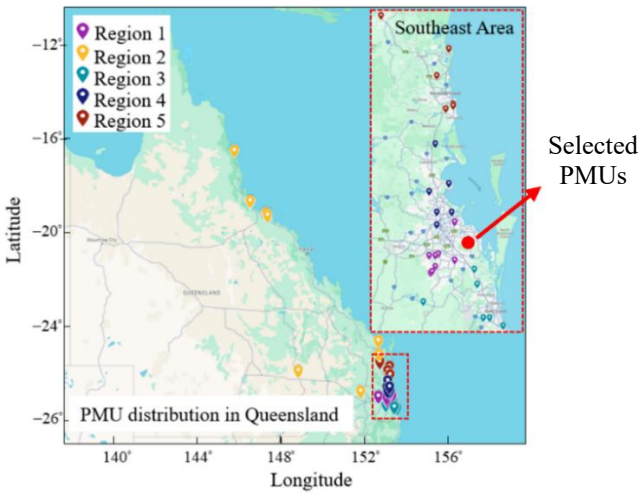

Fig. 2 Distribution of installed PMUs in Queensland, Australia

5min-ahead forecasting is evaluated with hourly PMU aggregates (Section II), rFFT spectral tokens, and weather/calendar features. A chronological split holds out 15% for testing; validation is 15% of the remaining training timeline. Two regimes are considered: **Zero-shot** and **Few-shot**.

The hyperparameters used during the model setup and training process are shown in TABLE I.

TABLE I
PARAMETERS OF MODEL SETUP AND TRAINING PROCESS

Model/Optimization	
Backbone	decoder-only LLM (frozen / LoRA)
Embedding dim	linear projection to hidden size 4096
LoRA rank r	8 (top layers adapted)
Optimizer	AdamW ($\beta_1 = 0.9, \beta_2 = 0.999$)
LR / Scheduler	0.0003 / cosine decay (warmup 5 epochs)
Batch / Epochs	32 / 50 (early stop patience 8)
Dropout	0.1 (projection & head)
Quantization	int8 weight-only (backbone), fp16 adapters
Inference	parallel head on Future tokens

B. Metrics

MAPE and NRMSE are chosen as the evaluation metrics, shown in (21) and (22):

$$MAPE\ [\%] = \frac{100}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{\max(|y_i|, \epsilon)}\right| \quad (21)$$

$$NRMSE\ [\%] = 100 \times \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}}{\bar{y}} \quad (22)$$

where $\epsilon = 10^{-3}$ and $\bar{y}$ is the mean load.

C. Model Accuracy Comparison

In this section, the forecasting performance of the proposed PMU–LLM framework against benchmark methods is evaluated. To provide an intuitive understanding, four representative test days are selected from the held-out dataset, covering different operating conditions such as smooth weekday profiles, sharp morning/evening ramps, and highly volatile weekend patterns. For each day, the actual feeder load is compared with forecasts generated by classical baselines, deep learning baselines, and the proposed approach.

As shown in Fig. 3, baseline models show noticeably lower fidelity to the measured curves, they tend to misalign peak timing and amplitudes. In contrast, the proposed PMU–LLM method achieves closer alignment with both the magnitude and timing of load variations across all scenarios. The qualitative results are shown in TABLE II, indicating the improved fit of the proposed model relative to baselines.

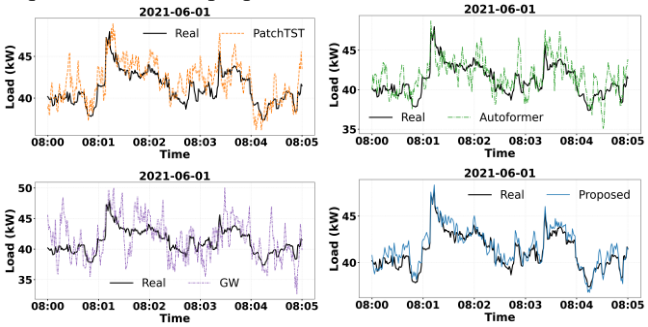

Fig. 3 Load forecasting curves with different models

TABLE II
BASELINE MODELS COMPARISON

Models	MAPE	NRMSE
PatchTST	3.27	4.21
Autoformer	3.14	3.89
GW	4.79	5.22
Proposed – LLM (LoRA)	1.88	2.19

D. Generalization Testing

In this section, the generalization performance of different models under both zero-shot and few-shot regimes is evaluated. The test feeder is unseen during training, representing a realistic domain shift scenario. Zero-shot results indicate how well a model can directly transfer, while few-shot results reflect its adaptability with a limited amount of feeder-specific data.

As shown in TABLE III, deep learning baselines (PatchTST, Autoformer, and GW) show limited transferability, with error reductions only after few-shot tuning. By contrast, the proposed PMU–LLM framework achieves substantially lower errors in both regimes. Notably, the ablation between a fully frozen backbone and LoRA-adapted backbone reveals that LoRA consistently improves generalization, cutting errors by about 20% compared to the frozen case. This demonstrates that the combination of PMU-driven tokens and parameter-efficient adaptation not only outperforms baselines but also scales effectively across feeders with minimal additional data.

TABLE III
ZERO-SHOT AND FEW-SHOT TESTING

Models	Zero-shot MAPE	Zero-shot NRMSE	Few-shot MAPE	Few-shot NRMSE

PatchTST	5.5	6.2	4.0	4.8
Autoformer	6.5	7.1	5.0	5.8
GW	8.8	9.3	7.5	8.0
Proposed – LLM (Frozen)	3.9	4.5	3.1	3.4
Proposed – LLM (LoRA)	3.2	3.7	2.4	3.1

E. Edge Deployment

In this section, the feasibility of deploying the proposed PMU–LLM forecaster on a resource-constrained edge platform is examined. Specifically, the NVIDIA Jetson Orin Nano is targeted, a compact substation-class device equipped with ARM CPUs and an integrated GPU but limited to 8 GB memory. While large language models typically exceed the memory and latency budgets of such hardware, weight-only int8 quantization is applied to compress the backbone while keeping the projection layers and LoRA adapters in half precision.

Fig. 4 presents representative real versus predicted load curves after edge deployment. Although quantization introduces a moderate drop in accuracy compared to the full-precision results, the overall forecasting quality remains satisfactory: peak timing and magnitudes are preserved, and error levels (MAPE = 3.42% and NRMSE = 4.33%) remain within acceptable operational margins. To visualize the results more clearly, the 5-minute-ahead forecasts are aggregated into hourly averages to form a full-day curve, which smooths out high-frequency fluctuations and highlights the overall trend. Under this hourly representation, the predicted trajectory appears stable and well aligned with the real load, further demonstrating that the proposed model can be effectively deployed at the feeder edge, maintaining robust accuracy under practical hardware constraints.

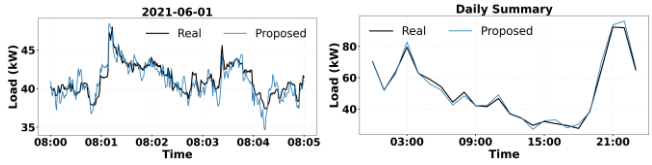

Fig. 4 Load forecasting curves with model deployed on NVIDIA Jetson

IV. Conclusion

This work proposed a PMU-driven feeder-level load forecasting framework that integrates physics-aware features with pre-trained large language models, achieving accurate, generalizable, and edge-ready forecasting. By tokenizing 1-Hz PMU aggregates, spectral descriptors, and exogenous context into unified sequences, and adapting decoder-only LLMs via frozen backbones or LoRA, the method consistently outperforms classical and deep learning baselines. Case studies on real Queensland PMU data demonstrated strong zero-/few-shot generalization across feeders, while edge deployment on a Jetson Orin Nano confirmed the feasibility of real-time execution under tight hardware constraints. These results highlight the potential of coupling synchrophasor data with LLMs to deliver scalable and operationally practical feeder-level forecasting, with future extensions toward probabilistic and multi-step forecasting under high DER penetration

Acknowledgment

This work is partially supported by the Department of Climate Change, Energy, the Environment and Water under the International Clean Innovation Researcher Networks (ICIRN) program (grant number ICIRN000077). This work is also partially supported by NOJA Power by providing the required synchrophasor data.

References

[1] Y. H. Lin, H. S. Tang, T. Y. Shen, and C. H. Hsia, "A Smart Home Energy Management System Utilizing Neurocomputing-Based Time-Series Load Modeling and Forecasting Facilitated by Energy Decomposition for Smart Home Automation," *IEEE Access,* vol. 10, pp. 116747-116765, 2022.

[2] J. Yan *et al.*, "Frequency-domain decomposition and deep learning based solar PV power ultra-short-term forecasting model," *IEEE Transactions on Industry Applications,* vol. 57, no. 4, pp. 3282-3295, 2021.

[3] C.-M. Lee and C.-N. Ko, "Short-term load forecasting using lifting scheme and ARIMA models," *Expert Systems with Applications,* vol. 38, no. 5, pp. 5902-5911, 2011/05/01/ 2011.

[4] K. Yun, R. Luck, P. J. Mago, and H. Cho, "Building hourly thermal load prediction using an indexed ARX model," *Energy and Buildings,* vol. 54, pp. 225-233, 2012/11/01/ 2012.

[5] S. Deng, X. Dong, L. Tao, J. Wang, Y. He, and D. Yue, "Multi-type load forecasting model based on random forest and density clustering with the influence of noise and load patterns," *Energy,* vol. 307, p. 132635, 2024/10/30/ 2024.

[6] X. Deng *et al.*, "Bagging–XGBoost algorithm based extreme weather identification and short-term load forecasting model," *Energy Reports,* vol. 8, pp. 8661-8674, 2022/11/01/ 2022.

[7] R. K. Jain, K. M. Smith, P. J. Culligan, and J. E. Taylor, "Forecasting energy consumption of multi-family residential buildings using support vector regression: Investigating the impact of temporal and spatial monitoring granularity on performance accuracy," *Applied Energy,* vol. 123, pp. 168-178, 2014/06/15/ 2014.

[8] M. Abumohsen, A. Y. Owda, and M. Owda, "Electrical Load Forecasting Using LSTM, GRU, and RNN Algorithms," *Energies,* vol. 16, no. 5, p. 2283, 2023.

[9] M. A. Majeed, S. Phichaisawat, F. Asghar, and U. Hussan, "Data-Driven Optimized Load Forecasting: An LSTM-Based RNN Approach for Smart Grids," *IEEE Access,* vol. 13, pp. 99018-99031, 2025.

[10] Y. Wang *et al.*, "Short-Term Load Forecasting for Industrial Customers Based on TCN-LightGBM," *IEEE Transactions on Power Systems,* vol. 36, no. 3, pp. 1984-1997, 2021.

[11] C. Wang, X. Li, Y. Shi, W. Jiang, Q. Song, and X. Li, "Load forecasting method based on CNN and extended LSTM," *Energy Reports,* vol. 12, pp. 2452-2461, 2024/12/01/ 2024.

[12] J. Cao *et al.*, "A short-term load forecasting method for integrated community energy system based on STGCN," *Electric Power Systems Research,* vol. 232, p. 110265, 2024/07/01/ 2024.

[13] W. Liu, X. Huang, S. Ji, D. Zhang, and S. Zhang, "A multi-scale graph tide model with enhanced graph structure learning and spatiotemporal decoupling for load forecasting," *Expert Systems with Applications,* vol. 295, p. 128839, 2026/01/01/ 2026.

[14] Y. Jiang *et al.*, "Very short-term residential load forecasting based on deep-autoformer," *Applied Energy,* vol. 328, p. 120120, 2022/12/15/ 2022.

[15] Z. Qu, Y. Meng, X. Hou, R. Chi, Y. Ai, and Z. Wu, "Integrated energy short-term multivariate load forecasting based on PatchTST secondary decoupling reconstruction for progressive layered extraction multi-task learning network," *Expert Systems with Applications,* vol. 269, p. 126446, 2025/04/15/ 2025.

[16] Y. Zhou and M. Wang, "Empower Pre-Trained Large Language Models for Building-Level Load Forecasting," *IEEE Transactions on Power Systems,* vol. 40, no. 5, pp. 4220-4232, 2025.

[17] H. Touvron *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971,* 2023.