HyperLoad: A Cross-Modality Enhanced Large Language Model-Based Framework for Green Data Center Cooling Load Prediction

Haoyu Jiang[1], Boan Qu[2], Junjie Zhu[1], Fanjie Zeng[2], Xiaojie Lin[1]*, Wei Zhong[1,3]

[1]College of Energy Engineering, Zhejiang University, Hangzhou, China
[2]Polytechnic Institute, Zhejiang University, Hangzhou, China
[3]Shanghai Institute for Advanced Study, Zhejiang University, Shanghai, China
{haoyu.jiang, boan.qu, junjie_zhu, zengfanjie, xiaojie.lin, wzhong}@zju.edu.cn

Abstract

The rapid growth of artificial intelligence is exponentially escalating computational demand, inflating data center energy use and carbon emissions, and spurring rapid deployment of green data centers to relieve resource and environmental stress. Achieving sub-minute orchestration of renewables, storage, and loads, while minimizing PUE and lifecycle carbon intensity, hinges on accurate load forecasting. However, existing methods struggle to address small-sample scenarios caused by cold start, load distortion, multi-source data fragmentation, and distribution shifts in green data centers. We introduce HyperLoad, a cross-modality framework that exploits pre-trained large language models (LLMs) to overcome data scarcity. In the Cross-Modality Knowledge Alignment phase, textual priors and time-series data are mapped to a common latent space, maximizing the utility of prior knowledge. In the Multi-Scale Feature Modeling phase, domain-aligned priors are injected through adaptive prefix-tuning, enabling rapid scenario adaptation, while an Enhanced Global Interaction Attention mechanism captures cross-device temporal dependencies. The public DCData dataset is released for benchmarking. Under both data sufficient and data scarce settings, HyperLoad consistently surpasses state-of-the-art (SOTA) baselines, demonstrating its practicality for sustainable green data center management.

Introduction

The rapid progress of artificial intelligence has caused computational demand to grow exponentially, doubling every 5.7 months since 2010, far exceeding hardware performance improvement (Giattino and Samborska 2025). Data from the Association for Computer Operations Management (AFCOM 2024) show that average rack power density in data centers increased from 6.1 kW in 2016 to 12 kW in 2024, imposing heavy stress on power and cooling systems. Traditional data centers, featuring "high-reliability, high-redundancy" designs and long-term high-load operation, further aggravate energy use and carbon emissions. By 2024, global data center electricity consumption reached 415 TWh, about 1.5% of total global demand (IEA 2024). Amid growing energy and environmental pressures, green data centers have become vital to reducing the ICT sector's carbon footprint. Green data centers should serve as flexible load nodes that are deeply integrated with the power grid, enabling coordinated scheduling with photovoltaic, wind, and energy storage systems across time scales from seconds to minutes. This interactive framework can sustain Power Usage Effectiveness below 1.3 and reduce lifecycle carbon emissions by over 30%.

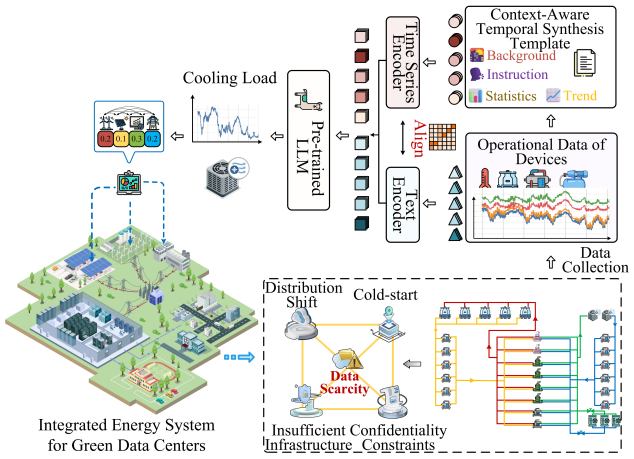

Figure 1: Schematic diagram of HyperLoad performing green data center cooling load prediction tasks.

Field surveys reveal that conventional data centers depend on fixed-load control without dynamic regulation, making cooling systems unable to respond to real-time conditions, causing significant energy waste (Radovanovic et al. 2022). Cooling accounts for more than 25% of total electricity use, the largest auxiliary load (IEA 2024). Green data centers emphasize smart operation, where accurate cooling load forecasting is key to optimizing energy allocation, integrating renewables, and ensuring reliability. Precise forecasting aligns cooling load with variable renewable outputs, reducing curtailment, storage redundancy, and overall costs (Han et al. 2025). Yet emerging facilities face severe data scarcity. Limited operational history constrains model training, renewable integration alters data distributions, and multi-source data from logs and environmental monitoring are incomplete due

*Corresponding author.

to privacy and compliance barriers (Figure 1).

Current research on data center cooling load forecasting still relies heavily on engineers' empirical rules, which depend on subjective judgment and lack quantitative rigor and replicability (Zhang et al. 2021). Thermodynamic physical models can represent system mechanisms but are costly to deploy due to complex parameter measurement, calibration, and numerical computation (Lin et al. 2022). With the rise of deep learning, models such as RNNs, CNNs, and Transformers have been introduced, markedly improving forecasting accuracy (Lu et al. 2022) (Figure 2). RNN-based models (e.g., NGCU (Wang et al. 2022), SegRNN (Lin et al. 2023)) capture short-term temporal relations but suffer from information decay over long sequences. CNN-based models (e.g., LightTS (Zhang et al. 2022), TSMixer (Chen et al. 2023)) extract local features efficiently and support parallel computation but cannot preserve temporal order, weakening sequence dependency. Transformer-based models (e.g., Informer (Zhou et al. 2021)) leverage attention mechanisms to learn temporal patterns, yet load variations depend on multiple heterogeneous factors, making single-modality modeling inadequate. Despite strong benchmark performance, these models require large, task-specific datasets for end-to-end training, and their learned representations are constrained by data distribution. When applied to green data centers with small samples, distribution shifts, and sparse labels, they often overfit and lose generalization, hindering fine-grained load prediction and intelligent operation.

Transformer-based LLMs such as GPT (OpenAI 2023) and LLaMA (Touvron et al. 2023) are trained through self-supervised learning on massive text corpora, acquiring generalized representations that enable high-accuracy inference with minimal task-specific data. Building on their cross-domain success (Jiang et al. 2025; Bi et al. 2024), recent research has explored applying LLMs to time-series forecasting by leveraging their sequence-pattern recognition and contextual-understanding abilities to address data scarcity. Models such as PromptCast (Xue and Salim 2024) and LF-PLM (Gao et al. 2024) focus on enhancing time-series feature representation, yet limited contextual reasoning reduces prediction accuracy. Time-LLM (Jin et al. 2024) and FPE-LLM (Qiu et al. 2024) incorporate multi-modal fusion to guide LLM encoding, but directly processing raw text and temporal data ignores distributional and semantic disparities, weakening the value of textual priors. To better capture periodic variations, TEMPO (Cao et al. 2024) applies STL decomposition to separate trend, seasonal, and residual components, though performance declines under complex seasonal dynamics (Bogahawatte et al. 2024). Furthermore, green data centers exhibit nonlinear, dynamically evolving feature interactions among devices. Most existing methods fail to model these couplings or adaptively adjust feature weights, constraining their effectiveness in capturing intricate dependencies within complex operational datasets.

To address the foregoing challenges, we propose a cross-modality enhanced LLM-based load forecasting framework termed HyperLoad, which integrates cross-modality knowledge alignment with multi-scale feature modeling to enable efficient prediction in complex, dynamic settings (Figure 1).

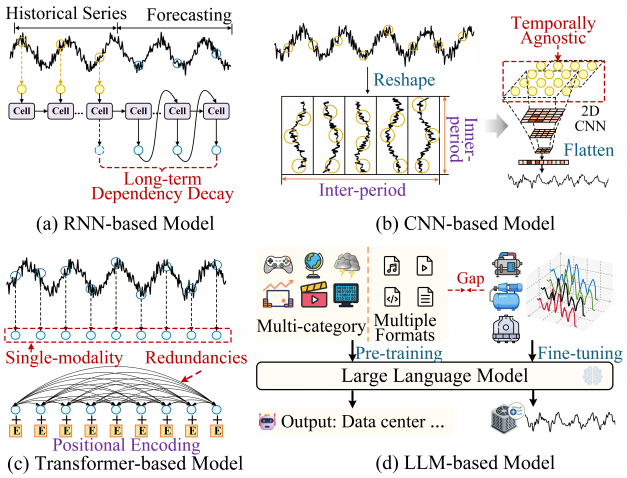

Figure 2: Limitations of time-series forecasting algorithms.

The methodology unfolds in two phases:

1. *Cross-Modality Knowledge Alignment phase*.

The Knowledge-Aligned Representation Integration (KARI) strategy is employed to dynamically adjust both the time series encoder and the text encoder, projecting text prior knowledge together with the load data features into a shared embedding space. The alignment enhances cross-modalities consistency, enabling LLMs to better leverage textual knowledge for temporal reasoning.

2. *Multi-Scale Feature Modeling phase*.

We propose an Adaptive Domain-specific Prefix Tuning (ADPT) strategy that leverages the text encoder's cross-modal representational capacity acquired in the first phase to encode text prior knowledge and embed it as prefixes. Enables LLMs to efficiently adapt to new scenarios with limited data while capturing underlying load fluctuation mechanisms. To address device dependencies, we design an Enhanced Global Interaction Attention (EGIA) mechanism that further models cross-variable coupling, ensuring robustness under sample sparsity and distribution shifts. Finally, by integrating text priors with temporal features, the model harnesses the trend comprehension and reasoning capabilities learned from large-scale distributed data during pre-training to achieve accurate cooling load forecasting.

The major contributions can be summarized as follows:

- We construct the dataset for data center cooling load forecasting, DCData, comprising key parameters to provide a standardized, systematic foundation for model development and evaluation.
- For the first time, we apply the trend-understanding and reasoning capabilities learned by LLMs from diverse distributions to the domain of green data center load forecasting, aiming to improve prediction accuracy and reduce the demand for large amounts of data.
- We introduce a KARI strategy that reshapes the text encoder feature space to narrow distributional gaps between text priors and time-series data, enhancing prior knowledge utilization. Building on this, we propose an ADPT

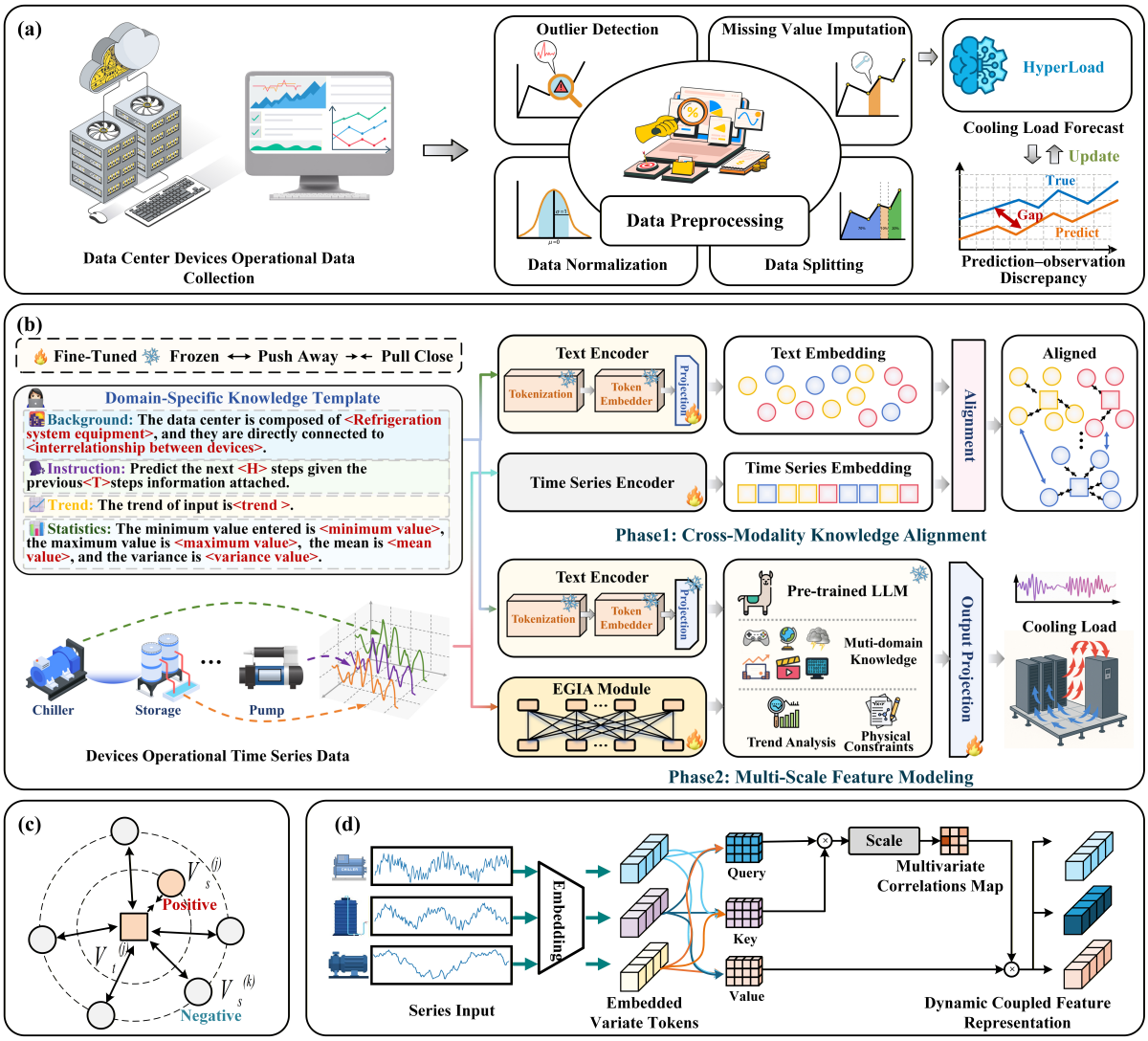

Figure 3: (a) Data collection and training process. (b) Framework of HyperLoad. (c) Schematic illustration of the KARI loss. (□ represents text modality, ○ represents time-series modality.) (d) Schematic illustration of the EGIA mechanism.

strategy that encodes critical green data center background as learnable prefix vectors, enabling rapid task adaptation without extensive retraining.

- We propose an EGIA mechanism designed to efficiently capture the coupling relationships and global patterns across variables, thereby enhancing the model's ability to model key dependencies among devices.
- Through experiments on DCData, HyperLoad outperforms SOTA baselines and demonstrates high accuracy and stability even under data scarcity.

Methodology

This section provides an overview of the proposed HyperLoad cooling load forecasting framework built on LLMs, as illustrated in Figure 3(b). HyperLoad comprises two phases:

1. *Cross-Modality Knowledge Alignment phase*: Using the KARI strategy to create a unified representation that merges textual prior knowledge with cooling load features.

2. *Multi-Scale Feature Modeling phase*: Through an ADPT strategy, domain prior knowledge is injected into the LLMs, guiding the model to apply pre-training knowledge to green data center scenarios, thereby enabling precise encoding of load features. An EGIA mechanism further extracts multi-scale, fine-grained patterns and cross-variable dependencies from the time series. Ultimately, by harnessing the pre-trained LLMs' capacities for trend analysis and logical reasoning, we achieve accurate cooling load forecasting.

Problem Definition

In the scenario of green data center cooling load forecasting, the historical observation data are denoted as $\mathcal{X} \in R^{T \times M}$, where T represents the length of the sampling period and M the number of monitored indicators. Given L consecu-

tive observations $x^{(j)} = (x_1, x_2, \ldots, x_L) \subset \mathcal{X}$, our objective is to predict the cooling load for the next K time steps, denoted as $\hat{y}^{(j)} \in R^{K\times 1}$. The model parameters are optimized by minimizing the prediction error between $\hat{y}^{(j)}$ and the ground-truth values $y^{(j)}$.

Cross-Modality Input Construction

Construction of Time-Series Modality Inputs. Given the high dimensionality, multi-scale characteristics, and non-stationarity of data in green data center environments, coupled with substantial distributional variability across devices, we apply reversible instance normalization to each feature column of $x^{(j)}$ to standardize them to zero mean and unit variance, thereby mitigating distributional heterogeneity. For the i-th feature column $x_i^{(j)} = \{x_{i,n}^{(j)}\}_{n=1}^{L} \in R^{L\times 1}$, compute its mean $\mu_i^{(j)}$ and standard deviation $\sigma_i^{(j)}$:

$$\mu_i^{(j)} = \frac{1}{L}\sum_{n=1}^{L} x_{i,n}^{(j)}, \qquad \sigma_i^{(j)} = \sqrt{\frac{1}{L}\sum_{n=1}^{L}\big(x_{i,n}^{(j)} - \mu_i^{(j)}\big)^2}. \quad (1)$$

For the n-th observation $x_{i,n}^{(j)}$ in this column, the normalized value is obtained as:

$$\bar{x}_{i,n}^{(j)} = \frac{x_{i,n}^{(j)} - \mu_i^{(j)}}{\sigma_i^{(j)}}. \quad (2)$$

The final normalized form of $x^{(j)}$ can thus be denoted as $\mathcal{P}^{(j)} = \big(\bar{x}_1^{(j)}, \bar{x}_2^{(j)}, \ldots, \bar{x}_M^{(j)}\big)$.

Construction of Text Modality Inputs. Building on the template paradigm introduced in CLIP (Radford et al. 2021), we construct a *Context-Aware Temporal Synthesis Template* for data center operations that serves as a priori embeddings for time-series data. By aligning text and time series representations within a shared latent space, we reinforce semantic coherence across the two modalities and steer the model toward capturing the intricate relationships among data center contexts, inter-device synergies, and load dynamics.

First, we construct a domain knowledge base $\mathcal{T}_b^{(j)}$, which encompasses domain knowledge of data centers, including:

A. Background: Provide an overview of the data center system's constituent modules together with their interdependencies, enabling the model to grasp the mechanisms of their coordinated operation.

B. Instruction: State the task specification unambiguously so that the model can correctly identify the prediction target and delineate the task boundaries.

Subsequently, we transform sequence $\mathcal{P}^{(j)}$ into its text counterpart $\mathcal{T}_s^{(j)}$, furnishing a structured input for later cross-modality alignment:

C. Trend: Describe the trend exhibited by the data segment, capturing long-term patterns and oscillations in cooling load to help model apprehend the dynamics of the series.

D. Statistics: Statistically analyze the data segment to obtain its extrema, mean, and variance, describing its distribution and fluctuations.

By concatenating the domain knowledge base $\mathcal{T}_b^{(j)}$ with the textual representation $\mathcal{T}_s^{(j)}$ of the time-series data, we obtain the *Context-Aware Temporal Synthesis Template* ($\mathcal{T}_{CATS}^{(j)}$) for sequence $\mathcal{P}^{(j)}$:

$$\mathcal{T}_{CATS}^{(j)} = \big[\,\mathcal{T}_b^{(j)},\ \mathcal{T}_s^{(j)}\big]. \quad (3)$$

Cross-Modality Knowledge Alignment Phase

During the Cross-Modality Knowledge Alignment phase, we introduce a KARI strategy that dynamically adjusts both the time series encoder and the text encoder. This strategy alleviates the distributional mismatch between the text and time-series modalities, thereby enabling more effective utilization of textual prior knowledge. After training, the text encoder is frozen and subsequently reused for all later training and inference phases.

Time-series modality feature encoding process. $\mathcal{P}^{(j)}$ is passed through the time series encoder ($\mathcal{F}_{\mathrm{TS}}(\cdot)$), which first embeds each variate into token-level representations (series representations of each token are extracted by the shared feed-forward network) and subsequently integrates them to derive the global time-series feature $\mathcal{V}_s^{(j)}$:

$$\mathcal{V}_s^{(j)} = \mathcal{F}_{\mathrm{TS}}\big(\mathcal{P}^{(j)}\big), \qquad \mathcal{V}_s^{(j)} \in R^d, \quad (4)$$

where d denotes the dimensionality of the features.

Text modality feature encoding process. The sequence $\mathcal{T}_{CATS}^{(j)}$ is processed by the text encoder to yield the text feature $\mathcal{V}_t^{(j)}$. The detailed procedure is as follows:

First, $\mathcal{T}_{CATS}^{(j)}$ is transformed into a discrete token sequence ($\mathcal{N}^{(j)} = (t_1^{(j)}, t_2^{(j)}, \ldots, t_n^{(j)})$) through tokenization and token embedding. On this basis, a feed-forward network ($\mathcal{E}$) maps each token $t_i^{(j)}$ to a learnable embedding $e_i^{(j)}$, thereby yielding the embedding sequence:

$$\mathcal{D}^{(j)} = \mathcal{E}\big(\mathcal{N}^{(j)}\big) = (e_1^{(j)}, e_2^{(j)}, \ldots, e_n^{(j)}), \quad (5)$$

a global representation $\mathcal{V}_t^{(j)} \in R^d$ is then extracted from the encoded sequence for downstream cross-modal alignment.

KARI Strategy. In the field of green data center load forecasting, the text modality (including load feature descriptions, domain knowledge bases, and task specifications) and the time-series modality (expressed as dynamic numerical sequences) differ fundamentally in both data distribution and semantic space. The former encodes structured knowledge through discrete symbols, whereas the latter represents complex system dynamics via continuous signals. This modal heterogeneity leads to a representational mismatch during feature-level fusion, thereby weakening the guiding role of textual prior knowledge in modeling load data.

To address the above challenge, this study proposes a KARI loss ($\mathcal{L}_{\mathcal{KARI}}$) that projects the feature representations of the time-series $\mathcal{V}_s^{(j)}$ and text $\mathcal{V}_t^{(j)}$ modalities into a unified shared latent space (see Figure 3(c)). In doing so, it ensures that the features produced by the text encoder become distributionally compatible with the time-series load data, allowing text priors to supply effective inductive support to the load encoding process and enhancing the model's

understanding of the mechanisms driving load fluctuations. $\mathcal{L}_{\mathrm{KARI}}$ is defined as follows:

$$\mathcal{L}_{\mathcal{KART}} = -\log \frac{\exp\big(\mathcal{C}\big(\mathcal{V}_t^{(j)}, \mathcal{V}_s^{(j)}\big)/\tau\big)}{\sum_{k=1}^{\mathcal{B}} \exp\big(\mathcal{C}\big(\mathcal{V}_t^{(j)}, \mathcal{V}_s^{(k)}\big)/\tau\big)}, \tag{6}$$

where $\mathcal{C}(\cdot, \cdot)$ denotes cosine similarity and τ the temperature (fixed to 0.05), $\mathcal{B}$ denotes batch size. The $\mathcal{L}_{\mathcal{KART}}$ minimizes the distance between each sequence feature and its corresponding text feature, thereby projecting them into a shared feature space and achieving cross-modality semantic alignment with efficient feature representations.

Multi-Scale Feature Modeling Phase

During the multi-scale feature modeling stage, we introduce an ADPT strategy and an EGIA mechanism. ADPT strategy injects domain prior knowledge into the LLMs in an efficient manner, guiding it to fully leverage knowledge accumulated during pre-training so as to encode load features with higher precision; EGIA mechanism strengthens the model's capacity to capture multi-scale fine-grained patterns and complex cross-variable dependencies. Ultimately, by exploiting the LLMs' strengths in trend analysis and reasoning, the framework achieves efficient prediction of cooling load.

ADPT Strategy. We first use the text encoder trained in the Cross-Modality Knowledge Alignment phase to encode the *Context-Aware Temporal Synthesis Template* adaptively constructed for each data patch. The resulting template encoding $\mathcal{V}_T^{(j)}$ is then used as a prefix and concatenated with the corresponding time-series encoding, and combined sequence is fed into LLMs. The objectives of this strategy are:

- **Knowledge transfer:** Guide the LLMs to apply the trend-understanding and reasoning capabilities accumulated during pre-training to problems in the data center.
- **Contextual understanding:** Enhance the LLMs' grasp of the intrinsic relationships between the coordinated operation of devices and fluctuations in cooling load.

EGIA Mechanism. In green data center scenarios, multiple environmental variables (e.g., inlet and outlet cooling-water temperature, air humidity, and equipment load) exhibit pronounced, temporally evolving coupling. Existing approaches primarily emphasize modeling temporal dependencies while overlooking inter-device collaboration and the adaptive allocation of informational weights, thereby limiting their effectiveness for load data with nonlinear and dynamically evolving characteristics. To address this issue, we design an EGIA mechanism that jointly learns multivariate sequence representations and their adaptive correlations, thereby elucidating the impact of device collaboration on cooling load fluctuations. The EGIA mechanism is illustrated in Figure 3(d). Specifically, we first embed the time series of each device-level variable in $\mathcal{P}^{(j)}$ as an independent token to obtain the feature representation $\mathcal{S}^{(j)}$:

$$\mathcal{S}^{(j)} = \mathrm{Embedding}\big(\mathcal{P}^{(j)}\big), \qquad \mathcal{S}^{(j)} \in R^{M \times d}. \tag{7}$$

The embedded features are linearly projected into query ($\mathcal{Q}^{(j)}$), key ($\mathcal{K}^{(j)}$), and value ($\mathcal{H}^{(j)}$) representations via three learnable projection matrices $\mathcal{W}_Q, \mathcal{W}_K, \mathcal{W}_H \in R^{d \times d}$:

$$\mathcal{Q}^{(j)} = \mathcal{S}^{(j)}\mathcal{W}_Q, \mathcal{K}^{(j)} = \mathcal{S}^{(j)}\mathcal{W}_K, \mathcal{H}^{(j)} = \mathcal{S}^{(j)}\mathcal{W}_H, \tag{8}$$

Subsequently, we model the correlations among variables, such that highly correlated variables receive greater weights in their subsequent interaction with the value representation:

$$\mathcal{V}_E^{(j)} = \mathcal{A}ttn\Big(\mathcal{Q}^{(j)}, \mathcal{K}^{(j)}, \mathcal{H}^{(j)}\Big) = \mathrm{Softmax}\Big(\frac{\mathcal{Q}^{(j)}\mathcal{K}^{(j)\top}}{\sqrt{d}}\Big)\mathcal{H}^{(j)}. \tag{9}$$

Pre-trained LLM-based Backbone. We adopt LLaMA-7B as the backbone model of HyperLoad. To maintain its pre-trained capacity for trend inference and reasoning, all parameters aside from the output prediction head are frozen while training. Fine-tuning procedure proceeds as follows:

First, we apply the ADPT strategy to concatenate the multimodal features $\mathcal{V}_T^{(j)}$ and $\mathcal{V}_E^{(j)}$, constructing the input:

$$\mathcal{V}_{in}^{(j)} = [\mathcal{V}_T^{(j)}, \mathcal{V}_E^{(j)}]. \tag{10}$$

Subsequently, $\mathcal{V}_{in}^{(j)}$ is fed into the pre-trained LLM $\big(\mathcal{O}_\theta^{\mathrm{LLM}}\big)$ to obtain the encoded representation, and the output is obtained through a trainable linear projection layer ($\mathcal{F}_{proj}()$):

$$\mathcal{V}_{out}^{(j)} = \mathcal{F}_{proj}\big(\mathcal{O}_\theta^{\mathrm{LLM}}\big(\mathcal{V}_{in}^{(j)}\big)\big). \tag{11}$$

Finally, the output is inverse-normalized to obtain the final prediction $\hat{y}^{(j)}$. We adopt the MSE as the training objective to quantify the discrepancy between the predicted values $\hat{y}^{(j)}$ and the ground-truth values $y^{(j)}$, and perform gradient updates accordingly:

$$\mathcal{L}_{MSE} = \frac{1}{K}\sum_{k=1}^{K}\Big(\hat{y}_k^{(j)} - y_k^{(j)}\Big)^2. \tag{12}$$

Experiments

Experimental Setup

High-quality public datasets are scarce because of privacy rules and limited infrastructure. To bridge this gap and evaluate HyperLoad, we collected 5-minute interval data from a data center in Dongguan spanning October to December, 2024. The dataset comprises 41 variables, including: outdoor temperature and humidity (2 groups); inlet/outlet temperatures of cooling water (9 cooling pumps); inlet/outlet temperatures of chilled water (9 chiller pumps); cooling load. After removing segments affected by commissioning and equipment faults, and applying a series of preprocessing steps (see Figure 3(a)), we obtained the DCData dataset (contains 13,438 records). It is chronologically partitioned training, validation, and test sets in a 7:1:2 ratio. We assess HyperLoad in two data-availability settings:

- **Data sufficient:** Model is trained on full training set to characterize the upper bound of achievable performance.
- **Data scarce:** Model is trained using 50% and 25% of training set (partitioned in chronological order), to simulate limited-data conditions and assess its potential applicability in green data centers.

Implemented in PyTorch and trained on a single NVIDIA A100 GPU, the model employed an input sequence length of 96 and forecasting length of 12, 24, 48, and 96 steps. Key hyper-parameters included an initial learning rate of 7×10^{-4}, batch size $\mathcal{B}$ = 64, and an 8 layer LLaMA backbone, with optimization via Adam.

Training Data (Percent)	Methods	PL = 12		PL = 24		PL = 48		PL = 96	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
100%	Autoformer (Wu et al. 2021)	0.0384	0.1307	0.0518	0.1613	0.1383	0.2613	0.0656	0.1944
	FreTS (Yi et al. 2023)	0.0065	0.0572	0.0093	0.0678	0.0166	0.0942	0.0270	0.1270
	iTransformer (Liu et al. 2024)	0.0065	0.0589	0.0108	0.0722	0.0153	0.0854	0.0203	0.0977
	LightTS (Zhang et al. 2022)	0.0070	0.0618	0.0103	0.0738	0.0240	0.1196	0.0410	0.1431
	N_Transformer (Liu et al. 2022b)	0.0096	0.0703	0.0130	0.0978	0.0144	0.0865	0.0240	0.1109
	PatchTST (Nie et al. 2023)	0.0075	0.0652	0.0105	0.0686	0.0145	0.0839	0.0293	0.1005
	PAttn (Tan et al. 2024)	0.0061	0.0569	0.0094	0.0692	0.0158	0.0883	0.0218	0.1008
	SCINet (Liu et al. 2022a)	0.0063	0.0579	0.0088	0.0661	0.0311	0.0783	0.0271	0.1025
	Transformer (Vaswani et al. 2017)	0.0872	0.1816	0.1563	0.3460	0.1043	0.2781	0.1496	0.2292
	TSMixer (Chen et al. 2023)	0.0134	0.0870	0.0229	0.1301	0.0510	0.1793	0.2822	0.4680
	TimesNet (Wu et al. 2023)	0.0062	0.0565	0.0098	0.0708	0.0155	0.0879	0.0308	0.1262
	HyperLoad	**0.0055**	**0.0539**	**0.0084**	**0.0637**	**0.0124**	**0.0748**	**0.0192**	**0.0931**
50%	Autoformer (Wu et al. 2021)	0.0315	0.1370	0.0381	0.1470	0.0434	0.1685	0.0691	0.2070
	FreTS (Yi et al. 2023)	0.0071	0.0595	0.0096	0.0683	0.0134	0.0796	0.0208	0.0997
	iTransformer (Liu et al. 2024)	0.0067	0.0592	0.0118	0.0761	0.0139	0.0809	0.0211	0.0991
	LightTS (Zhang et al. 2022)	0.0077	0.0659	0.0181	0.1074	0.0280	0.1334	0.0510	0.1878
	N_Transformer (Liu et al. 2022b)	0.0075	0.0624	0.0101	0.0705	0.0154	0.0866	0.0223	0.1050
	PatchTST (Nie et al. 2023)	0.0062	0.0565	0.0090	0.0669	0.0149	0.0836	0.0227	0.1046
	PAttn (Tan et al. 2024)	0.0068	0.0599	0.0093	0.0682	0.0135	0.0812	0.0220	0.1020
	SCINet (Liu et al. 2022a)	0.0067	0.0593	0.0096	0.0691	0.0139	0.0807	0.0217	0.1020
	Transformer (Vaswani et al. 2017)	0.1323	0.3184	0.5614	0.7100	1.0982	0.9328	1.8995	1.2011
	TSMixer (Chen et al. 2023)	0.0402	0.1564	0.0991	0.2255	0.3436	0.4792	0.6154	0.6494
	HyperLoad	**0.0058**	**0.0544**	**0.0087**	**0.0638**	**0.0128**	**0.0757**	**0.0200**	**0.0950**
25%	Autoformer (Wu et al. 2021)	0.1171	0.2413	0.1368	0.2547	0.1519	0.2944	0.1782	0.3519
	FreTS (Yi et al. 2023)	0.0238	0.1203	0.0351	0.1469	0.0454	0.1617	0.0760	0.2029
	iTransformer (Liu et al. 2024)	0.0177	0.0978	0.0227	0.1082	0.0299	0.1227	0.0454	0.1499
	LightTS (Zhang et al. 2022)	0.0202	0.1087	0.0514	0.1867	0.1025	0.2702	0.2133	0.3826
	N_Transformer (Liu et al. 2022b)	0.0168	0.0927	0.0269	0.1153	0.1078	0.2592	0.0937	0.2185
	PatchTST (Nie et al. 2023)	0.0241	0.1011	0.0325	0.1305	0.0305	0.1220	0.0540	0.1663
	PAttn (Tan et al. 2024)	0.0203	0.1051	0.0268	0.1179	0.0360	0.1358	0.0499	0.1583
	SCINet (Liu et al. 2022a)	0.0302	0.1269	0.0300	0.1306	0.0424	0.1470	0.0561	0.1700
	Transformer (Vaswani et al. 2017)	2.4365	1.4666	4.1371	1.9232	4.4457	1.9520	6.006	2.3439
	TSMixer (Chen et al. 2023)	0.5039	0.6698	0.8543	0.8881	3.4351	1.6968	5.8421	2.0656
	HyperLoad	**0.0140**	**0.0839**	**0.0197**	**0.0969**	**0.0294**	**0.1170**	**0.0442**	**0.1435**

Table 1: Prediction performance using 100%, 50%, and 25% of the training data.

Experimental Results

Model Performance in the Data Sufficient Setting. First, we evaluate all models with input length of 96 and prediction lengths (PL) of 12, 24, 48, and 96. The results are summarized in top of Table 1, with the best scores highlighted in bold. HyperLoad consistently outperforms all SOTA baselines across all forecast lengths, underscoring its advantage for cooling load forecasting. Specifically, at the 96-step prediction length, HyperLoad reduces MAE by 52.1%, 26.7%, 4.7%, 34.9%, 16.1%, 7.4%, 7.6%, 9.2%, 59.4%, and 26.2% relative to Autoformer, FreTS, iTransformer, LightTS, N_Transformer, PatchTST, PAttn, SCINet, Transformer, and TimesNet, respectively. Corresponding MSE reductions are 70.7%, 28.9%, 5.4%, 53.2%, 20.0%, 34.5%, 11.9%, 29.1%, 87.2%, and 37.7%, respectively.

In experiments with both input and prediction sequences fixed at 96 time steps, HyperLoad was evaluated over horizons of 1, 32, 64, and 96 (S-Table 1). It achieved the highest accuracy at all horizons. Visualizations (Figure 4, S-Figures 1–4) show that HyperLoad consistently follows target trends more closely than other models, confirming its superior and stable forecasting performance across different time steps, demonstrates its ability to accurately perform predictions at varying horizons according to practical requirements.

Model Performance in the Data Scarce Setting. Owing to the challenges imposed by data scarcity, the performance of all models declined. Nevertheless, HyperLoad capitalizes on the prior knowledge accumulated during pre-training to make accurate predictions of cold-start data center load demand with only a limited quantity of data, and its performance is correspondingly less affected. Concretely, across the four forecasting lengths examined, HyperLoad outperforms SOTA baselines in the data scarce setting where 50% of the training data are provided (middle of Table 1), reducing the MSE by 6.5%, 3.3%, 4.5 %, and 3.9% and the MAE by 3.7%, 4.6%, 4.9%, and 4.1%, respectively. Under an extreme data scarce setting with only 25% of the training data (bottom of Table 1), the corresponding reductions in MSE are 16.7%, 13.2%, 1.7%, and 2.6%, while MAE decreases by 9.5%, 10.5%, 4.1%, and 4.3%, respectively.

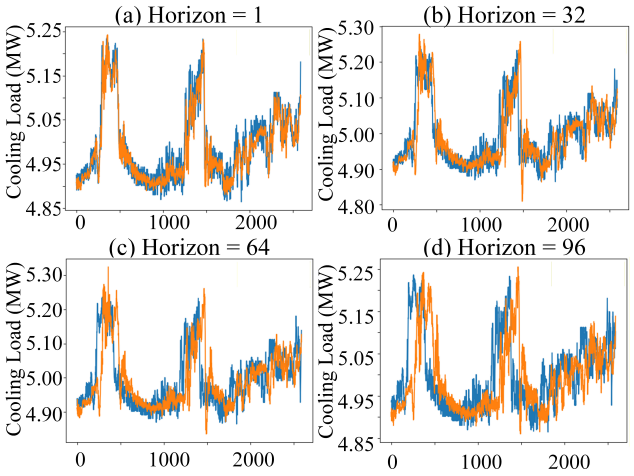

Figure 4: Prediction results for different horizons. (Blue lines: ground truth; orange lines: predictions.)

Additionally, we evaluated each method with both the input and prediction sequence lengths fixed at 96 across four forecasting horizons(H = 1, 32, 64, 96). The corresponding results are shown in Figure 5. Under all settings, HyperLoad demonstrates consistently strong predictive performance and achieves the best overall results. This shows that HyperLoad effectively transfers LLMs' trend-understanding and reasoning capabilities to data center load forecasting, mitigating data scarcity in real-world settings.

Ablation Study. To validate the effectiveness of the HyperLoad framework, we conducted ablation experiments on the full training dataset using an input length of 96 and an output length of 96. The results are listed in Table 2.

Variants	MSE	MAE
w/o ADPT	0.0220	0.1006
w/o EGIA	0.0211	0.0981
w/o KARI	0.0194	0.0941
HyperLoad	**0.0192**	**0.0931**

Table 2: Results of the ablation study.

- **ADPT:** By incorporating text knowledge via cross-modal fusion, the model's MSE and MAE losses are reduced by 12.7% and 7.5%, respectively. This demonstrates that embedding domain-specific background information effectively facilitates the LLMs' adaptation to data center context, enabling it to leverage pre-training knowledge to encode the complex latent patterns and trends present in multivariate time-series data from data center operations.
- **KARI:** Implementing a knowledge-alignment strategy yields additional reductions of 1.0% and 1.1% in the model's MSE and MAE, respectively. This indicates that improving distributional consistency across modalities can effectively reduce cross-modal semantic bias and improves the utilization of textual priors.

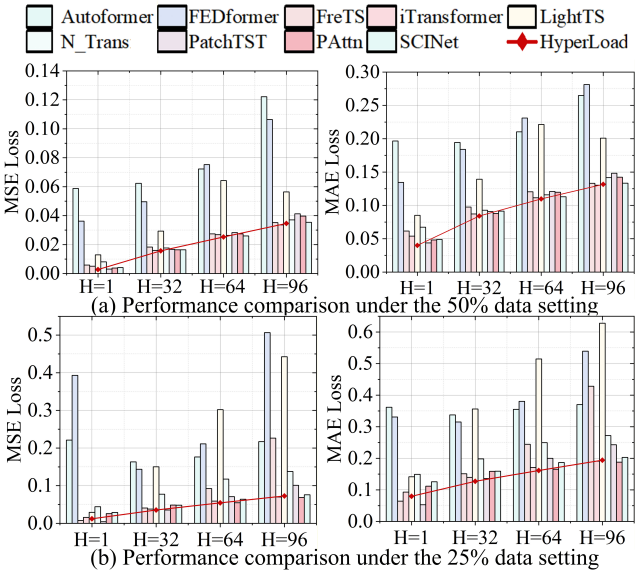

Figure 5: Model performance in the data scarce setting.

- **EGIA:** By integrating the EGIA mechanism, the model's MSE and MAE were reduced by 9.0% and 5.1%, respectively, demonstrating that this module can effectively capture the dynamic interrelationships between devices, aid the LLMs in understanding the underlying drivers of load fluctuations, and thereby enhance the accuracy and stability of cooling load forecasting.
- **Backbone:** In Table 3, we compare the training time and predictive performance of three backbone models (BERT, GPT-2, and LLaMA). LLaMA attains the highest accuracy, with a per-iteration runtime of 0.3928 s, which falls between BERT (0.4289 s) and GPT-2 (0.1943 s), thus effectively balancing speed and precision.

Backbones	Time (s / iter)	MSE	MAE
BERT	0.4289	0.0217	0.1019
GPT2	**0.1943**	0.0224	0.1012
LLaMA	0.3928	**0.0192**	**0.0931**

Table 3: Ablation studies of backbone models.

Conclusion

This paper introduces HyperLoad, a framework for forecasting cooling load demand in green data centers. By synergistically combining cross-modal knowledge alignment with multi-scale feature modeling, HyperLoad transfers the long-range dependency modeling and contextual understanding capabilities acquired by LLMs during pre-training to the cooling load prediction task, allowing rapid adaptation to new scenarios and high-accuracy forecasts even under limited data. Experiments show that HyperLoad significantly outperforms mainstream baselines across multiple forecast horizons in both data sufficient and data scarce settings.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No.52576234) and the National Key R & D Program of China (Grant No.2023YFE0108600). This work is also supported by the State Key Laboratory of Alternate Electrical Power System with Renewable Energy Sources (Grant No.LAPS25016).

References

AFCOM. 2024. State of the Data Center Report 2024. https://www.datacenterknowledge.com/cooling/data-center-rack-density-has-doubled-and-it-s-still-not-enough.

Bi, Z.; Zhang, N.; Xue, Y.; Ou, Y.; Ji, D.; Zheng, G.; and Chen, H. 2024. OceanGPT: A Large Language Model for Ocean Science Tasks. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 3357–3372.

Bogahawatte, J.; Seneviratne, S.; Perera, M.; and Halgamuge, S. 2024. Rethinking Time Series Forecasting with LLMs via Nearest Neighbor Contrastive Learning. ArXiv preprint arXiv:2412.04806.

Cao, D.; Jia, F.; Arik, S. ; Pfister, T.; Zheng, Y.; Ye, W.; and Liu, Y. 2024. TEMPO: Prompt-Based Generative Pre-Trained Transformer for Time Series Forecasting. In *Proceedings of the International Conference on Learning Representations*.

Chen, S. A.; Li, C. L.; Yoder, N.; Arik, S. O.; and Pfister, T. 2023. TSMixer: An All-MLP Architecture for Time Series Forecast-ing. *Transactions on Machine Learning Research*.

Gao, M.; Zhou, S.; Gu, W.; Wu, Z.; Hu, Z.; Zhu, H.; and Liu, H. 2024. LFPLM: A General and Flexible Load Forecasting Framework based on Pre-trained Language Model. ArXiv preprint arXiv:2406.11336.

Giattino, C.; and Samborska, V. 2025. Since 2010, the training computation of notable AI systems has doubled every six months. https://ourworldindata.org/data-insights/since-2010-the-training-computation-of-notable-ai-systems-has-doubled-every-six-months.

Han, J.; Han, K.; Han, T.; Wang, Y.; Han, Y.; and Lin, J. 2025. Data-Driven Distributionally Robust Optimization of Low-Carbon Data Center Energy Systems Considering Multi-Task Response and Renewable Energy Uncertainty. *Journal of Building Engineering*, 102: 111937.

IEA. 2024. Energy demand from AI. https://www.iea.org/reports/energy-and-ai/energy-demand-from-ai.

Jiang, H.; Cheng, Z. Q.; Moreira, G.; Zhu, J.; Sun, J.; Ren, B.; He, J. Y.; Dai, Q.; and Hua, X. S. 2025. UCDR-Adapter: Exploring Adaptation of Pre-Trained Vision-Language Models for Universal Cross-Domain Retrieval. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 5429–5438.

Jin, M.; Wang, S.; Ma, L.; Chu, Z.; Zhang, J. Y.; Shi, X.; Chen, P. Y.; Liang, Y.; Li, Y. F.; Pan, S.; et al. 2024. Time-LLM: Time Series Forecasting by Reprogramming Large Language Models. In *Proceedings of the International Conference on Learning Representations*.

Lin, J.; Lin, W.; Lin, W.; Wang, J.; and Jiang, H. 2022. Thermal Prediction for Air-Cooled Data Center Using Data-Driven-Based Model. *Applied Thermal Engineering*, 217: 119207.

Lin, S.; Lin, W.; Wu, W.; Zhao, F.; Mo, R.; and Zhang, H. 2023. SegRNN: Segment Recurrent Neural Network for Long-Term Time Series Forecasting. ArXiv preprint arXiv:2308.11200.

Liu, M.; Zeng, A.; Chen, M.; Xu, Z.; Lai, Q.; Ma, L.; and Xu, Q. 2022a. Scinet: Time series modeling and forecasting with sample convolution and interaction. In *Proceedings of the Conference on Neural Information Processing Systems*, 5816–5828.

Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; and Long, M. 2024. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. In *Proceedings of the International Conference on Learning Representations*.

Liu, Y.; Wu, H.; Wang, J.; and Long, M. 2022b. Non-Stationary Transformers: Exploring the Stationarity in Time Series Forecasting. In *Proceedings of the Conference on Neural Information Processing Systems*, 9881–9893.

Lu, F.; Chai, Z.; Huang, J.; Liu, S.; and Wang, C. 2022. Load Forecasting for Data Centers Based on Time Series. In *Proceedings of the International Conference on Applied Machine Learning*, 372–377.

Nie, Y.; Nguyen, N. H.; Sinthong, P.; and Kalagnanam, J. 2023. A Time Series Is Worth 64 Words: Long-Term Forecasting with Transformers. In *Proceedings of the International Conference on Learning Representations*.

OpenAI. 2023. GPT-4 Technical Report. ArXiv preprint arXiv:2303.08774.

Qiu, Z.; Li, C.; Wang, Z.; Mo, H.; Xie, R.; Chen, G.; and Dong, Z. 2024. FPE-LLM: Highly Intelligent Time-Series Forecasting and Language Interaction LLM in Energy Systems. ArXiv preprint arXiv:2411.00852.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning Transferable Visual Models from Natural Language Supervision. In *Proceedings of the International Conference on Machine Learning*, 8748–8763.

Radovanovic, A.; Chen, B.; Talukdar, S.; Roy, B.; Duarte, A.; and Shahbazi, M. 2022. Power modeling for effective datacenter planning and compute management. *IEEE Transactions on Smart Grid*, 13: 1611–1621.

Tan, M.; Merrill, M. A.; Gupta, V.; Althoff, T.; and Hartvigsen, T. 2024. Are Language Models Actually Useful for Time Series Forecasting? In *Proceedings of the Conference on Neural Information Processing Systems*, 60162–60191.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M. A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. LLaMA: Open and Efficient Foundation Language Models. ArXiv preprint arXiv:2302.13971.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention Is All You Need. In *Proceedings of the Conference on Neural Information Processing Systems*, 5999–6009.

Wang, J.; Li, X.; Li, J.; Sun, Q.; and Wang, H. 2022. NGCU: A New RNN Model for Time-Series Data Prediction. *Big Data Research*, 27: 100296.

Wu, H.; Hu, T.; Liu, Y.; Zhou, H.; Wang, J.; and Long, M. 2023. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *Proceedings of the International Conference on Learning Representations*.

Wu, H.; Xu, J.; Wang, J.; and Long, M. 2021. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. In *Proceedings of the Conference on Neural Information Processing Systems*, 22419–22430.

Xue, H.; and Salim, F. D. 2024. PromptCast: A New Prompt-Based Learning Paradigm for Time Series Forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 36: 6851–6864.

Yi, K.; Zhang, Q.; Fan, W.; Wang, S.; Wang, P.; He, H.; An, N.; Lian, D.; Cao, L.; and Niu, Z. 2023. Frequency-Domain MLPs Are More Effective Learners in Time Series Forecasting. In *Proceedings of the Conference on Neural Information Processing Systems*, 76656–76679.

Zhang, Q.; Meng, Z.; Hong, X.; Zhan, Y.; Liu, J.; Dong, J.; Bai, T.; Niu, J.; and Deen, M. J. 2021. A Survey on Data Center Cooling Systems: Technology, Power Consumption Modeling and Control Strategy Optimization. *Journal of Systems Architecture*, 119: 102253.

Zhang, T.; Zhang, Y.; Cao, W.; Bian, J.; Yi, X.; Zheng, S.; and Li, J. 2022. Less Is More: Fast Multivariate Time Series Forecasting with Light Sampling-Oriented MLP Structures. ArXiv preprint arXiv:2207.01186.

Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; and Zhang, W. 2021. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 11106–11115.