CREF-LLM: Contrastive Representation Encoding with Frozen Language Models for Few-Shot Wind Power Forecasting

Jingdong Wang
School of Computer Science
Northeast Electric Power University
Jilin, China
707569380@qq.com

Zhichao Zhang*
School of Computer Science
Northeast Electric Power University
Jilin, China
1446513892@qq.com

Lina Zhou
Beijing Institute of Computer Technology and Application
Beijing, China
2189192916@qq.com

Fanqi Meng
School of Computer Science
Northeast Electric Power University
Jilin, China
meng8191@163.com

***Abstract*—Accurate wind power forecasting is critical for power system scheduling and electricity market operations. However, newly deployed wind farms face significant challenges due to limited historical data, as conventional forecasting methods typically require substantial training samples and tend to overfit in few-shot scenarios. Pre-trained large language models (LLMs) offer a promising solution—frozen LLM backbones can leverage pre-trained sequence modeling capabilities while avoiding overfitting to limited data. Yet, existing LLM-based methods primarily focus on general-purpose adaptation strategies without designs specific to wind power forecasting: they lack mechanisms for learning robust temporal representations from limited data, and fail to address the unique characteristics of wind power such as cross-variable dependencies and inherent periodicity. To address these limitations, we propose CREF-LLM, a framework that integrates a contrastive temporal encoder with a frozen language model backbone. Contrastive pre-training enables learning robust temporal representations without requiring external datasets, while multivariate joint encoding captures cross-variable dependencies and explicit temporal features model periodic patterns. Experiments on two real-world wind farms with different operational characteristics demonstrate that, with only 30 days of training data, our method achieves consistent superiority over state-of-the-art methods across all forecasting horizons.**

***Index Terms*—Wind power forecasting, few-shot learning, contrastive learning, large language models, time series**

I. Introduction

The global transition toward renewable energy has positioned wind power as a critical component of sustainable electricity generation, with global wind capacity reaching over 1,000 GW by the end of 2023 [1]. Accurate wind power forecasting is essential for grid integration, enabling operators to balance supply and demand, schedule reserves, and minimize curtailment. Among various forecasting horizons, ultra-short-term forecasting, spanning 1 to 4 hours ahead, is particularly critical for real-time dispatch and intraday market operations [2].

However, forecasting for newly deployed wind farms presents a significant challenge: limited historical data. While established wind farms may have years of operational records, new sites typically have only weeks to months of data available. This data scarcity is particularly problematic for deep learning methods, which have achieved remarkable success in time series forecasting [3] but often require substantial training data to learn effective representations. When applied to few-shot scenarios, these models tend to overfit, resulting in poor generalization performance.

The challenge is further compounded by the domain-specific nature of wind power generation. Accurate forecasting requires capturing the complex coupling among meteorological variables—where wind speed, temperature, pressure, and humidity interact in nonlinear ways that vary across locations. Additionally, wind power exhibits strong diurnal and seasonal periodicity, patterns that are difficult to learn implicitly from limited training data, motivating the use of explicit temporal features. With limited data, standard deep learning approaches struggle to simultaneously avoid overfitting while capturing these patterns, leading to suboptimal performance in practical deployment.

In this paper, we present a framework combining a contrastive temporal encoder with frozen language models for few-shot wind power forecasting. The contrastive encoder learns robust temporal representations from the site's own limited data without requiring external datasets, while the frozen language model avoids overfitting and leverages pre-trained sequence modeling capabilities. Our approach further incorporates multivariate joint encoding and explicit temporal features to capture wind power characteristics under data scarcity.

*Corresponding author: Zhichao Zhang.

The main contributions of this paper are as follows:

- We propose a framework that integrates a contrastive temporal encoder with a frozen language model. Specifically, we employ contrastive learning to pre-train the temporal encoder from limited data, then leverage the frozen language model for sequence-to-sequence modeling without fine-tuning, enabling effective forecasting with only 30 days of data.
- We demonstrate through comprehensive ablation studies that multivariate joint encoding and explicit temporal features are critical for wind power forecasting under data scarcity. Specifically, multivariate joint encoding captures cross-variable dependencies and proves to be the most essential component, followed by explicit temporal features for modeling the diurnal and seasonal periodicity inherent to wind power generation.
- We validate the proposed framework on two real-world wind farms with different operational characteristics, demonstrating consistent superiority over state-of-the-art methods across all forecasting horizons.

II. Related Work

Wind power forecasting research has evolved from classical statistical methods through deep learning architectures to recent explorations of pre-trained language models. This section reviews three research directions relevant to our work: general time series forecasting approaches, few-shot learning techniques that address data scarcity, and emerging methods that leverage large language models for temporal prediction.

A. Time Series Forecasting

Traditional statistical methods, particularly the Autoregressive Integrated Moving Average (ARIMA) family [4], have long served as the foundation for time series forecasting. While effective for linear and stationary patterns, these approaches struggle with the nonlinear dynamics inherent in complex systems such as wind power generation [5]. Deep learning has substantially advanced the field. Recurrent architectures, especially Long Short-Term Memory (LSTM) networks [6], effectively model sequential dependencies but suffer from gradient issues over long sequences. More recently, Transformer-based models have achieved state-of-the-art results: Informer [7] introduces ProbSparse self-attention for computational efficiency, PatchTST [8] segments sequences into patches for improved representation learning, and iTransformer [9] enhances multivariate modeling by treating variates as tokens. However, DLinear [10] demonstrates that simple linear models can achieve comparable performance, questioning whether architectural complexity is always necessary. More critically, these methods typically require substantial training data and tend to overfit when applied to few-shot scenarios.

B. Few-shot Learning for Time Series

To address data scarcity challenges, researchers have explored transfer learning and meta-learning approaches. Transfer learning methods pre-train models on data-rich sources and fine-tune for target tasks with limited samples. For wind power forecasting, Hu et al. [11] propose a shared hidden layer deep neural network that transfers knowledge from established wind farms to newly deployed sites. Meta-learning offers an alternative paradigm by learning to adapt quickly from minimal samples. The Model-Agnostic Meta-Learning (MAML) framework [12] enables models trained across diverse tasks to rapidly adapt to new domains. Chen et al. [13] further developed a meta-learning approach specifically for few-shot wind power forecasting, demonstrating that with only 30 days of training data, meta-learned models can achieve accuracy equivalent to months of supervised learning. However, these approaches typically require access to external datasets from similar domains for pre-training or meta-training. For newly deployed wind farms, such auxiliary data may not be readily available, and even when accessible, the distribution shift between source and target domains can significantly limit transfer effectiveness.

Self-supervised contrastive learning offers an alternative paradigm for representation learning without requiring external datasets. Methods such as TS-TCC [14] and CoST [15] demonstrate that contrastive learning can effectively learn robust temporal representations from unlabeled time series data. However, these methods have not been specifically designed for few-shot forecasting scenarios or integrated with pre-trained language models to leverage their sequence modeling capabilities.

C. LLM-based Time Series Forecasting

Recent works explore leveraging pre-trained language models for time series tasks. GPT4TS [16] demonstrates that frozen GPT-2 with only embedding layer fine-tuning achieves competitive forecasting performance while requiring minimal trainable parameters. Time-LLM [17] introduces input reprogramming to align time series with the text modality, enabling few-shot generalization. These LLM-based approaches show particular promise in data-limited scenarios by leveraging pre-trained sequence modeling capabilities without overfitting. However, existing methods primarily focus on general-purpose adaptation strategies without designs specific to wind power forecasting. They typically lack explicit mechanisms for capturing cross-variable dependencies or the diurnal and seasonal periodicity inherent to wind power generation. Building upon these advances, our framework combines a contrastive temporal encoder with a frozen language model backbone for learning site-specific temporal representations. Additionally, we introduce multivariate joint encoding and explicit temporal features specifically designed to address the unique characteristics of wind power forecasting under data scarcity.

III. Methodology

Wind power forecasting for newly deployed wind farms presents a significant challenge due to limited historical data. We propose a framework that combines a contrastive pre-trained encoder with a frozen language model backbone. Both contrastive pre-training and downstream forecasting are

performed using only the site's 30-day training split without requiring external datasets, which is crucial for newly deployed wind farms where historical data are limited. Fig. 1 illustrates the overall architecture.

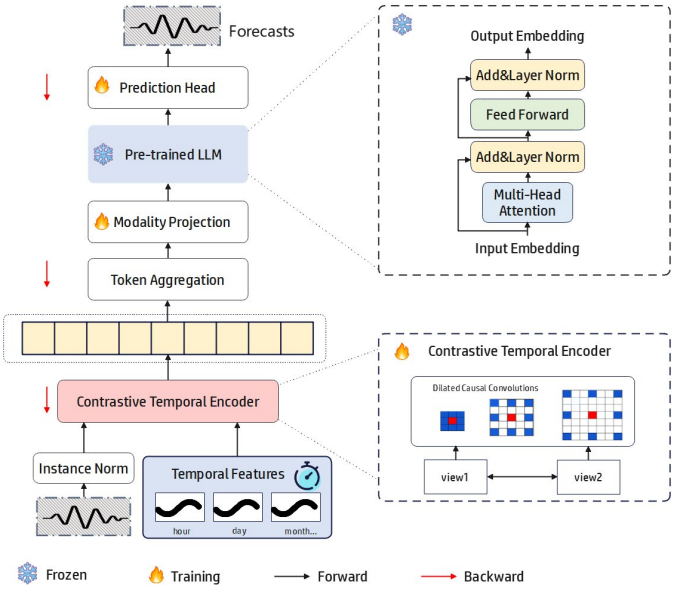


Fig. 1: Overall architecture of the proposed framework. The contrastive temporal encoder (pre-trained, then fine-tuned) processes multivariate inputs with temporal features. Token aggregation and modality projection transform encoded representations for the frozen LLM backbone.

The training consists of two stages. We first pre-train the temporal encoder on the unlabeled 30-day training split using hierarchical contrastive learning. In the forecasting stage, we fine-tune the encoder together with a lightweight projection layer and an MLP prediction head, while keeping the language model frozen. This design improves site adaptation under limited data while avoiding the instability of updating a large language model.

A. Input Representation

The input at each time step concatenates 9 meteorological variables and the historical wind power measurement, forming a 10-dimensional raw vector, and then appends 7 normalized temporal features (e.g., hour-of-day, day-of-year). Wind power generation is governed by complex interactions among meteorological factors: wind direction determines the effective wind speed at the turbine rotor, temperature affects air density and power conversion efficiency, and wind speeds at different heights reflect atmospheric stability. To capture these cross-variable dependencies, we adopt multivariate joint encoding, where all variables are processed together by a shared encoder, rather than the channel-independent approach that processes each variable separately.

Let $\mathbf{x}_t \in \mathbb{R}^{10}$ denote the raw input vector containing meteorological variables and historical power, and let $\mathbf{e}_t \in \mathbb{R}^7$ denote the normalized temporal features. Because wind power series are non-stationary, we apply instance normalization over the input window to stabilize the distribution before encoding:

$$\hat{\mathbf{x}}_t = \frac{\mathbf{x}_t - \boldsymbol{\mu}}{\boldsymbol{\sigma} + \epsilon} \tag{1}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are the mean and standard deviation computed over the input sequence. The normalized features are concatenated with temporal features to form the encoder input $\mathbf{z}_t = [\hat{\mathbf{x}}_t; \mathbf{e}_t] \in \mathbb{R}^{17}$.

B. Contrastive Temporal Encoder

The contrastive temporal encoder transforms the input sequence into time-dependent representations by modeling dependencies along the time axis. It employs a dilated causal convolutional architecture with exponentially increasing dilation rates. Specifically, the dilation rate at layer l is $d_l = 2^l$. Each convolutional block contains dilated causal convolutions with residual connections:

$$\mathbf{u}^{(l)} = f_{d_l}(\mathbf{u}^{(l-1)}) + \mathbf{u}^{(l-1)} \tag{2}$$

where f_{d_l} denotes two stacked dilated convolutions with dilation rate d_l and GELU activation. The exponentially increasing dilation enables the network to capture patterns at multiple temporal scales while maintaining a receptive field that covers the entire input sequence.

The encoder is pre-trained using a hierarchical contrastive loss. Given an input sequence, two overlapping views are generated through random cropping. The contrastive objective combines an instance-level loss that distinguishes different samples at each timestamp, and a temporal-level loss that captures the temporal structure within each sample. These losses are applied at multiple scales through a hierarchical structure: after computing the losses, the representations are downsampled by max pooling, and the losses are computed again at coarser scales until the temporal dimension reduces to 1:

$$\mathcal{L} = \frac{1}{D} \sum_{d=0}^{D-1} \left[\lambda \mathcal{L}_{\text{inst}}^{(d)} + (1-\lambda) \mathcal{L}_{\text{temp}}^{(d)} \right] \tag{3}$$

where D is the number of scales and λ balances the two components. After contrastive pre-training, the encoder is initialized with the learned weights and fine-tuned during the forecasting stage.

C. LLM-based Forecasting

To leverage the sequence modeling capabilities of pre-trained language models, the encoded sequence must be transformed into a token sequence compatible with the language model's input format. This transformation involves three steps: token aggregation, modality projection, and output prediction.

Token Aggregation. The encoded representations are segmented into non-overlapping segments, and each segment is aggregated through mean pooling to form a token. Let $\mathbf{r}_t$ denote the encoder output at time step t. The i-th token is computed as:

$$\mathbf{h}_i = \frac{1}{P} \sum_{t=(i-1)P+1}^{iP} \mathbf{r}_t \tag{4}$$

where P is the number of time steps per token.

Modality Projection. Each aggregated token is projected to match the language model's embedding space through a projection layer:

$$\mathbf{g}_i = \text{GELU}(\text{LayerNorm}(\mathbf{W}_g \cdot \mathbf{h}_i + \mathbf{b}_g)) \tag{5}$$

where $\mathbf{W}_g \in \mathbb{R}^{d_{\text{LM}} \times d_{\text{enc}}}$ is the projection matrix, $\mathbf{b}_g$ is the bias, d_{enc} is the encoder output dimension, and d_{LM} is the language model's hidden dimension.

Output Prediction. The projected tokens are fed into the frozen language model. Let $\mathbf{o}_i$ denote the language model output for the i-th token. Unlike approaches that only use the last token's output for prediction, we utilize all token outputs. A prediction head maps each output to future time steps:

$$\hat{\mathbf{y}}_i = \mathbf{W}_2(\text{GELU}(\mathbf{W}_1 \cdot \mathbf{o}_i + \mathbf{b}_1)) + \mathbf{b}_2 \tag{6}$$

where $\mathbf{W}_1 \in \mathbb{R}^{d_h \times d_{\text{LM}}}$ and $\mathbf{W}_2 \in \mathbb{R}^{P \times d_h}$ are weight matrices, d_h is the hidden dimension, and $\mathbf{b}_1$, $\mathbf{b}_2$ are biases. The predictions from all tokens are concatenated and denormalized using the input statistics to obtain the final forecast.

IV. Experiments

To demonstrate the effectiveness of our approach, we conduct comparative experiments on wind power forecasting between our method and state-of-the-art baselines under a few-shot setting. In addition, we perform ablation experiments to verify the contribution of joint encoding, temporal features, and contrastive pre-training. Two real-world wind farms with different data quality characteristics are used to assess the robustness and generalizability of the proposed framework.

A. Experimental Settings

Datasets. Site-1 (36 MW installed capacity) has a zero-value ratio of 28.9%, indicating many low/zero-power intervals and comparatively noisier measurements. Such patterns are often associated with curtailment, maintenance, or sensor issues in operational wind farms. Site-2 (96 MW installed capacity) has a zero-value ratio of 4.9%, representing higher-quality operational data. Both datasets are sampled at 15-minute intervals. We use 30 days for training, 7 days for validation, and 14 days for testing.

Prediction Horizons. We evaluate ultra-short-term prediction at 1, 2, 3, and 4 hours ahead. Additionally, we include 6-hour prediction to assess the potential for short-term forecasting.

Baselines. We compare with seven methods: DLinear [10], Informer [7], PatchTST [8], iTransformer [9], TimesNet [18], GPT4TS [16], and AutoTimes [19]. All methods use identical data splits and undergo hyperparameter tuning.

Evaluation Metrics. We use Mean Absolute Error (MAE) and Mean Squared Error (MSE) as evaluation metrics:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{7}$$

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{8}$$

where y_i and $\hat{y}_i$ denote the ground truth and predicted values, respectively, and n is the number of samples.

Implementation Details. The input sequence length is 96 time steps, corresponding to 24 hours at 15-minute intervals. The encoder has 10 convolutional layers with output dimension 320. The encoded sequence is segmented into 4 tokens, where each token aggregates 24 time steps, and projected to 768 dimensions to match GPT-2. For contrastive pre-training, we train for 200 epochs with batch size 8 and $\lambda = 0.5$. For forecasting, we use Adam optimizer with initial learning rate 0.001 and learning rate decay, training for 20 epochs with early stopping (patience = 5). All experiments use a fixed random seed of 2021 for reproducibility.

B. Main Results

Tables I and II present results across five prediction horizons. Our method achieves the best performance on all 10 tasks.

Our method consistently outperforms all baselines across all prediction horizons. Compared to GPT4TS, the strongest baseline, our method achieves MSE reductions of 26.9%, 26.4%, 18.7%, 14.0%, and 26.4% for 1h to 6h predictions on Site-1, and 24.0%, 26.5%, 37.1%, 18.1%, and 7.4% on Site-2. The Transformer-based methods (Informer, iTransformer, PatchTST, TimesNet) show substantially higher errors across all horizons, likely due to overfitting with limited training data.

Fig. 2 illustrates MSE trends across prediction horizons. Two observations emerge: (1) Our method and GPT4TS significantly outperform other baselines, which cluster together with MSE above 0.3 even at 1-hour horizon; (2) The gap between our method and GPT4TS widens as the horizon increases. On Site-1, this gap grows from 0.035 at 1h to 0.155 at 6h, demonstrating stronger long-range forecasting capability.

Fig. 3 visualizes predictions during a high-variance period where power rapidly transitions from near 0 to approximately 35 MW. Our method closely tracks the ground truth throughout the entire period, including both the low-power region and the rapid ramp-up. In contrast, GPT4TS produces negative predictions in the low-power region, which are physically infeasible since power output cannot be negative. Informer exhibits systematic overestimation, particularly in the high-power region.

C. Ablation Study

To verify the contribution of each component, ablation experiments are performed by systematically replacing or removing key design choices. Table III presents the results for 1-hour prediction on both sites.

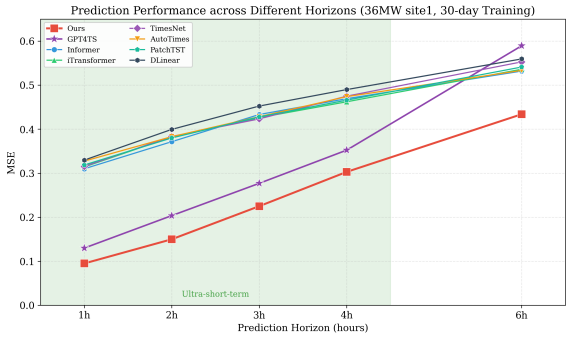
Joint encoding contributes the most, with a 35.6% MSE increase on Site-1 and 12.1% on Site-2 when replaced by channel-independent (CI) encoding. This empirically indicates that modeling cross-variable dependencies is important for wind power forecasting, since interactions among wind speed, wind direction, and air temperature affect aerodynamic power conversion.

TABLE I: Results on Site-1 (36 MW). Best results in **bold**.

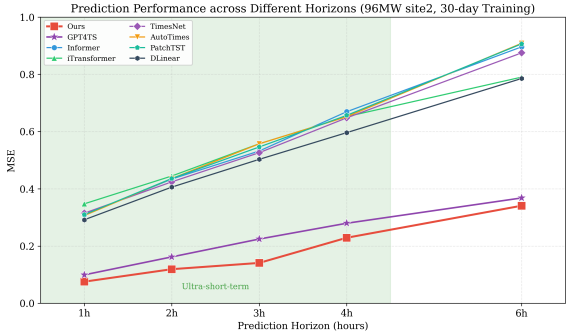
Horizon	Metric	**Ours**	GPT4TS	Informer	iTransformer	TimesNet	PatchTST	AutoTimes	DLinear
1h	MSE	**0.0950**	0.1299	0.3100	0.3141	0.3153	0.3186	0.3274	0.3297
1h	MAE	**0.2102**	0.2484	0.2938	0.2993	0.2939	0.2951	0.3081	0.3027
2h	MSE	**0.1499**	0.2036	0.3715	0.3824	0.3834	0.3800	0.3829	0.3994
2h	MAE	**0.2677**	0.3125	0.3404	0.3476	0.3530	0.3458	0.3463	0.3524
3h	MSE	**0.2250**	0.2769	0.4332	0.4260	0.4232	0.4278	0.4283	0.4524
3h	MAE	**0.3267**	0.3692	0.3931	0.3805	0.3761	0.3826	0.3816	0.3899
4h	MSE	**0.3029**	0.3523	0.4692	0.4621	0.4749	0.4658	0.4738	0.4898
4h	MAE	**0.3704**	0.4126	0.4101	0.4059	0.4183	0.4124	0.4129	0.4152
6h	MSE	**0.4338**	0.5892	0.5318	0.5360	0.5527	0.5410	0.5334	0.5594
6h	MAE	**0.4603**	0.5275	0.4534	0.4538	0.4663	0.4624	0.4511	0.4608

TABLE II: Results on Site-2 (96 MW). Best results in **bold**.

Horizon	Metric	**Ours**	GPT4TS	Informer	iTransformer	TimesNet	PatchTST	AutoTimes	DLinear
1h	MSE	**0.0755**	0.0993	0.3079	0.3475	0.3148	0.3095	0.3065	0.2917
1h	MAE	**0.1740**	0.2102	0.2548	0.2754	0.2593	0.2606	0.2486	0.2556
2h	MSE	**0.1191**	0.1620	0.4340	0.4446	0.4242	0.4358	0.4358	0.4058
2h	MAE	**0.2272**	0.2812	0.3095	0.3269	0.3153	0.3159	0.3099	0.3093
3h	MSE	**0.1410**	0.2244	0.5320	0.5573	0.5261	0.5460	0.5571	0.5028
3h	MAE	**0.2545**	0.3308	0.3647	0.3900	0.3659	0.3574	0.3618	0.3550
4h	MSE	**0.2289**	0.2796	0.6692	0.6521	0.6477	0.6564	0.6516	0.5961
4h	MAE	**0.3434**	0.3767	0.4206	0.4098	0.4124	0.4051	0.4062	0.3976
6h	MSE	**0.3408**	0.3680	0.8958	0.7901	0.8748	0.9062	0.9082	0.7853
6h	MAE	**0.4175**	0.4239	0.5291	0.4940	0.4929	0.4910	0.5033	0.4759



(a) Site-1 (36 MW)



(b) Site-2 (96 MW)

Fig. 2: MSE comparison across prediction horizons.

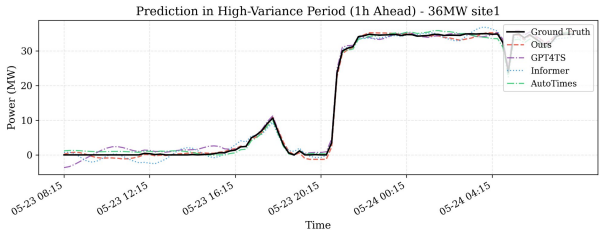


Fig. 3: Prediction visualization during a high-variance period (1h ahead, Site-1).

TABLE III: Ablation study results for 1-hour prediction.

Configuration	Site-1 MSE	Site-1 Δ%	Site-2 MSE	Site-2 Δ%
Full model	**0.095**	–	**0.072**	–
w/o Joint encoding (CI)	0.129	+35.6	0.081	+12.1
w/o Temporal features	0.115	+20.8	0.080	+10.1
w/o Contrastive pre-training	0.109	+14.2	0.076	+5.7

Temporal features contribute 20.8% on Site-1 and 10.1% on Site-2. Wind power exhibits strong diurnal and seasonal patterns that are difficult to learn from limited data alone.

Contrastive pre-training improves performance by 14.2% on Site-1 and 5.7% on Site-2. The larger gain on Site-1 (noisier data) indicates that pre-training provides stronger regularization under challenging conditions.

The larger gains on Site-1 suggest that these design choices

are especially beneficial under more challenging data conditions with frequent low/zero-power intervals and higher noise. The consistent improvements across both sites indicate that the proposed framework is robust across different data quality regimes.

V. Conclusion

We proposed a few-shot wind power forecasting framework that integrates a contrastive-pretrained temporal encoder with a frozen language model. Contrastive pre-training enables learning robust temporal representations from limited data, while the frozen language model backbone avoids overfitting and leverages pre-trained sequence modeling capabilities. Multivariate joint encoding proves essential for capturing cross-variable dependencies, and explicit temporal features further enhance performance by modeling periodic patterns. The approach generalizes well across sites with different operational characteristics, demonstrating its practical value for newly deployed wind farms where historical data remains limited.

References

[1] International Renewable Energy Agency, "Renewable Capacity Statistics 2024," IRENA, Abu Dhabi, 2024.

[2] S. Hanifi, X. Liu, Z. Lin, and S. Lotfian, "A critical review of wind power forecasting methods—Past, present and future," *Energies*, vol. 13, no. 15, p. 3764, 2020.

[3] Q. Wen, T. Zhou, C. Zhang, W. Chen, Z. Ma, J. Yan, and L. Sun, "Transformers in time series: A survey," in *Proc. IJCNN*, 2023, pp. 6778–6786.

[4] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*, 5th ed. Wiley, 2015.

[5] M. Elsaraiti and A. Merabet, "A comparative analysis of the ARIMA and LSTM predictive models and their effectiveness for predicting wind speed," *Energies*, vol. 14, no. 20, p. 6782, 2021.

[6] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[7] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proc. AAAI*, 2021, pp. 11106–11115.

[8] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," in *Proc. ICLR*, 2023.

[9] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long, "iTransformer: Inverted transformers are effective for time series forecasting," in *Proc. ICLR*, 2024.

[10] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" in *Proc. AAAI*, 2023, pp. 11121–11128.

[11] Q. Hu, R. Zhang, and Y. Zhou, "Transfer learning for short-term wind speed prediction with deep neural networks," *Renewable Energy*, vol. 85, pp. 83–95, 2016.

[12] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. ICML*, 2017, pp. 1126–1135.

[13] F. Chen, J. Yan, Y. Liu, Y. Yan, and L. B. Tjernberg, "A novel meta-learning approach for few-shot short-term wind power forecasting," *Applied Energy*, vol. 362, p. 122838, 2024.

[14] E. Eldele, M. Ragab, Z. Chen, M. Wu, C. K. Kwoh, X. Li, and C. Guan, "Time-series representation learning via temporal and contextual contrasting," in *Proc. IJCAI*, 2021, pp. 2352–2359.

[15] G. Woo, C. Liu, D. Sahoo, A. Kumar, and S. Hoi, "CoST: Contrastive learning of disentangled seasonal-trend representations for time series forecasting," in *Proc. ICLR*, 2022.

[16] T. Zhou, P. Niu, L. Sun, and R. Jin, "One fits all: Power general time series analysis by pretrained LM," in *Proc. NeurIPS*, 2023, pp. 43322–43355.

[17] M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi, P. Chen, Y. Liang, Y. Li, S. Pan, and Q. Wen, "Time-LLM: Time series forecasting by reprogramming large language models," in *Proc. ICLR*, 2024.

[18] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "TimesNet: Temporal 2D-variation modeling for general time series analysis," in *Proc. ICLR*, 2023.

[19] Y. Liu, G. Qin, X. Huang, J. Wang, and M. Long, "AutoTimes: Autoregressive time series forecasters via large language models," in *Proc. NeurIPS*, 2024.