

A Load Forecasting Method Based on Large Language Model Considering Numerical-Text Separation[#]

Bing Sun^{1*}, Junwei Yang¹, Peng Li¹, Hao Yu¹, Haoran Ji¹, Guanyu Song¹, Jinli Zhao¹

¹ School of Electrical and Information Engineering, Tianjin University

ABSTRACT

To address the challenges of nonlinearity, non-stationarity, and multimodal fusion in power load forecasting under the new power system environment, this paper proposes a load forecasting method based on large language models that considers the separation of numerical and textual data. The method constructs a dual-channel architecture for numerical and textual separation processing, using a frozen large language model encoder to extract textual features and a multi-layer perceptron (MLP) to extract numerical features. After feature concatenation, the input is fed into the frozen large language model decoder, and the results are finally output through a numerical prediction head. By freezing the core weights of the large language model and introducing only a small number of trainable layers, the method retains the model's strong semantic understanding capability while reducing computational overhead. A case study based on real load data from a large city in northern China demonstrates that the proposed NTSForecast method outperforms mainstream time series forecasting models such as Informer, Autoformer, and PatchTST in various evaluation metrics, including mean square error (MSE) and mean absolute error (MAE), validating the effectiveness and superiority of this method.

Keywords: load forecasting; large language models; numerical-text separation; dual-channel architecture; multimodal fusion

1. INTRODUCTION

Power load forecasting, as a core technology for grid dispatching and demand-side management, directly impacts the safe and economic operation of power systems and the efficient integration of renewable energy [6]. With China's accelerated progress toward its "dual carbon" goals and the development of a new power system, large-scale grid integration of intermittent energy sources like distributed photovoltaics and wind power, along with rapid growth

in new loads such as electric vehicles and energy storage devices, has created highly uncertain characteristics on both the supply and demand sides of the power system. Meanwhile, frequent extreme weather events, deepening power market reforms, and user-side flexible resources participating in grid interactions have made load curves increasingly complex. Traditional forecasting methods based on numerical statistics struggle to effectively capture these nonlinear and non-stationary dynamic characteristics [8]. These challenges demand enhanced generalization capabilities, robustness, and interpretability from forecasting models, urgently requiring the introduction of new modeling approaches to improve prediction accuracy and reliability.

However, modern power load forecasting is fundamentally a typical multimodal fusion problem. Beyond the load curve itself, heterogeneous information such as task description texts and meteorological data (e.g., temperature, humidity, wind speed) all contain critical predictive signals [8]. By jointly learning the temporal dynamics of load data and semantic information from textual and visual modalities, multimodal fusion techniques can significantly enhance prediction robustness and accuracy in scenarios like extreme weather events and major incidents [2,3]. Current multimodal forecasting for power loads still faces several technical challenges: First, the challenge of heterogeneous modal alignment. The continuity of numerical load sequences and the discreteness of textual modalities create an inherent feature space gap, making traditional linear mappings or simple tokenization inadequate for preserving key temporal features and physical constraints [15].

To address this, recent studies propose structure-guided cross-modal alignment frameworks that learn state transition matrices from meteorological texts via hidden Markov models, then inject these as prior knowledge into temporal representation learning, effectively improving modal alignment fidelity [4]. Second, the difficulty in constructing multimodal

[#] This is a paper for the 17th International Conference on Applied Energy (ICAE2025), December 8-12, 2025, Bangkok, Thailand.

datasets. The key challenge lies in effectively integrating numerical and textual data during sample set construction to ensure sufficient input information. Third, lightweight deployment and real-time performance. Power grid dispatching requires strict timeliness for load forecasting, necessitating multi-region predictions within 5-15 minutes. However, the massive parameter size and computational overhead of large language models limit their deployment on edge computing nodes [11]. To address this, researchers have explored parameter-efficient fine-tuning techniques [9], model compression and knowledge distillation methods, as well as lightweight adaptation schemes based on prompt learning [13].

However, while these models enhance computational speed, they also incur certain precision losses. This paper proposes a lightweight numerical text separation prediction method based on large language models. By introducing a small number of trainable additional layers while keeping the core weights of large language models frozen, this method achieves the function of separating numerical features from textual features, thereby significantly improving the load prediction performance.

2. DATA PROCESSING AND EVALUATION INDICATORS

2.1 Data Sources

This paper selects load data from a large city in northern China for analysis. To protect privacy, the original power data has been preprocessed to eliminate dimensions. The original load data contains missing and abnormal phenomena in some periods, which are processed by interpolation to complete the data resolution to a 15-minute interval. The original meteorological data includes temperature, humidity, air pressure, wind direction, and wind speed.

2.2 Feature Analysis and Sample Set Construction

The Pearson correlation coefficient is used to analyze the meteorological factors affecting the load, and the analysis results are shown in the figure below.

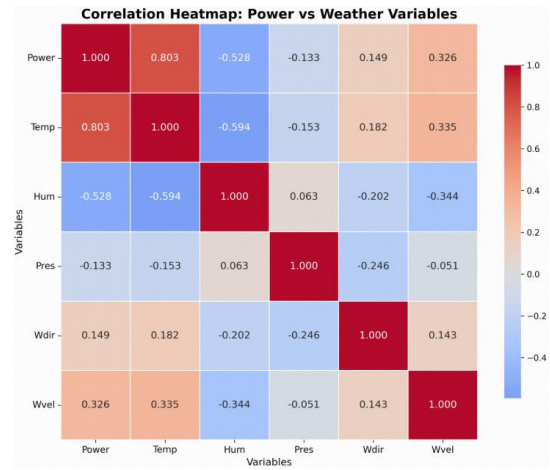


Fig. 1 Pearson correlation analysis of meteorological factors

As can be seen, temperature, humidity, and wind speed are the meteorological factors that most significantly impact load, hence they are selected as the meteorological dimension features for constructing the load prediction sample set.

The autocorrelation function (ACF) test was performed on the time series of line load rate. ACF curves with different lag durations were calculated, with a unit lag duration of 15 minutes, as shown in the figure below.

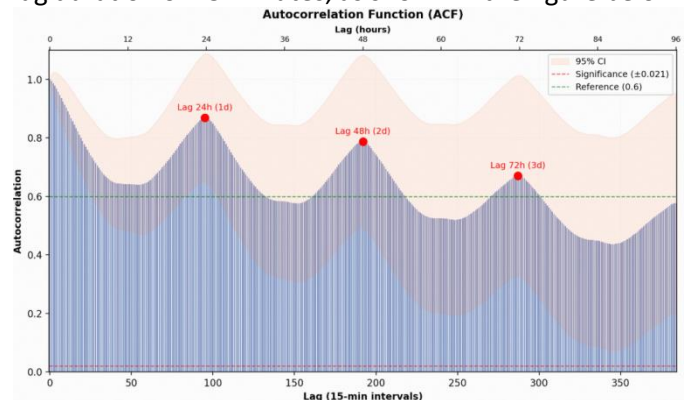


Fig. 2 Analysis of load power autocorrelation

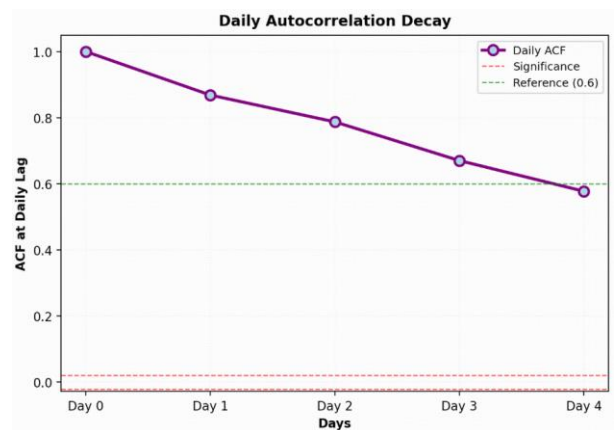


Fig. 3 load autocorrelation decay map

The graph shows that the lag time reaches a peak every 24 hours, indicating a significant daily cycle in the load. The autocorrelation coefficients for the first three days are all above 0.6, suggesting a strong correlation. However, on the fourth day, the autocorrelation coefficient drops below 0.6, and the confidence interval becomes wider, indicating a weaker correlation. Therefore, the data from this day is not included in the sample input.

The model sample set is constructed as shown in the table below. The left side represents the expected input of the model, while the right side shows the model's output. The overall input is a 12-dimensional sample set with one-dimensional output.

Tab. 1 Model sample set construction

Input Data	Expected Output
Load at T-2 on D-3 Load at T-2 on D-2 Load at T-2 on D-1	Load at time T on day D
Load at T-1 of D-3 Load at T-1 of D-2 Load at T-1 of D-1	
Load at T time on D-3 Load at T time on D-2 Load at T time on D-1	
Temperature at time T on day D Humidity at time T on day D Wind speed at time T on day D	

3. NUMERICAL-TEXT DUAL-CHANNEL MODEL

The sample set for model training is illustrated below, divided into three fields: the text section is blue, and the numerical section is black. The "task_description" field contains a pure text description of the task background. The "weather_features" field represents three-dimensional meteorological features selected through feature extraction. The "load_history" field contains historical load data selected via autocorrelation coefficient analysis.

```
{
  "task_description": "You are an assistant for load forecasting. Your task is to predict the load at D day T time. To do this, you are given: (1) weather conditions (temperature, humidity, and wind speed) at D day T time, and (2) historical load values at T, T-15 minutes, and T-30 minutes from D-1, D-2, and D-3 days.",
  "weather_features": [
    {"label": "D day T time temperature", "value": 27.2},
    {"label": "D day T time humidity", "value": 91.375},
    {"label": "D day T time wind speed", "value": 0.9125}
  ],
  "load_history": [
    {"label": "D-3 day T-30min load", "value": 1.917925887},
    {"label": "D-3 day T-15min load", "value": 1.919695107},
    {"label": "D-3 day T load", "value": 1.92435312},
    {"label": "D-2 day T-30min load", "value": 2.018880987},
    {"label": "D-2 day T-15min load", "value": 2.02704078},
    {"label": "D-2 day T load", "value": 1.99821514},
    {"label": "D-1 day T-30min load", "value": 2.000767707},
    {"label": "D-1 day T-15min load", "value": 1.974033947},
    {"label": "D-1 day T load", "value": 1.970773}
  ]
}
```

Fig. 4 The dataset used by the model

To better integrate prior textual information, we designed a dual-channel architecture for load forecasting based on digital-text separation. Specifically, this architecture is tailored for ultra-short-term load forecasting. In the input layer, digital and textual data are separated, then extracted through distinct channels to obtain features, which are subsequently merged in the fusion layer. The workflow diagram is illustrated below.

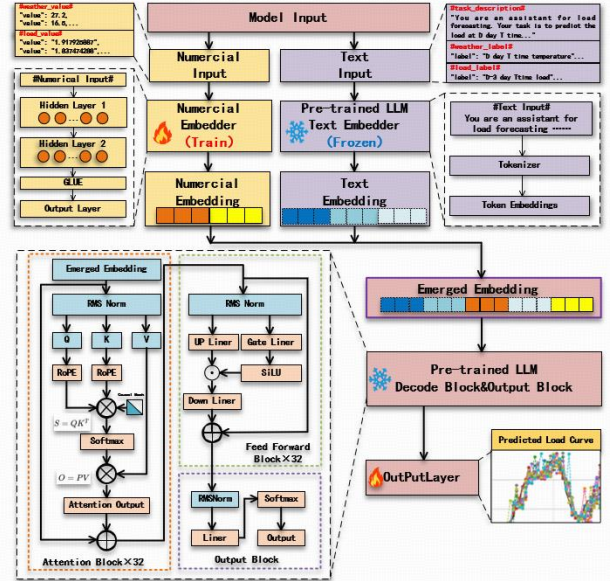


Fig. 5 Architectures of Numerical-Text Dual-Channel Model

For input text, it undergoes word segmentation and tokenization sequentially through the frozen large language model encoder, mapping the text into the semantic space of the large language model. The process can be described as follows:

$$\text{Tokenizer: } \mathcal{X} \rightarrow \mathcal{V}^n \quad (1)$$

Where: \mathcal{X} represents the input text space (e.g., a numeric sequence), \mathcal{V}^n denotes the model's vocabulary, and n indicates the sequence length after segmentation.

The word segmentation tool then splits the text sequence into basic units, retrieves their integer indices

from a lookup table, and establishes the following mapping:

$$\text{Input String} \mapsto [t_1, t_2, \dots, t_n] \quad (2)$$

Furthermore, the discrete index sequence is transformed into a vector representation through the embedding layer within the model, ultimately forming the model input tensor.

$$X = [E(t_1), E(t_2), \dots, E(t_n)] \in \mathbb{R}^{n \times d} \quad (3)$$

Here, d denotes the learning dimension, and E is the embeddable matrix.

For input numbers, we process them through an MLP network with L layers. The computation process from layer L to layer $L+1$ is as follows:

$$z^{(l+1)} = W^{(l)} a^{(l)} + b^{(l)} \quad (4)$$

$$a^{(l+1)} = \sigma(z^{(l+1)}) \quad (5)$$

Here, $W^{(l)}$ denotes the weight matrix from layer L to layer $L+1$, $b^{(l)}$ is the bias vector of layer L , $z^{(l+1)}$ represents the pre-activation value of layer $L+1$, and $a^{(l+1)}$ is the activation output of layer L . To enhance the model's expressive power and ensure stable gradients and training, the GLUE activation function is adopted.

The concatenated features are fed into the decoding layer of the large language model, where the weight parameters are frozen to preserve the model's language capabilities.

The data stream is processed by the large model to output the hidden layer states. Since the large model has not encountered the fused feature inputs during the fine-tuning phase, it cannot yet generate load prediction values. Therefore, this paper introduces a numerical prediction head after the output layer of the large language model to align the model's output with the task requirements.

For evaluating the model's prediction performance, the following four metrics are calculated, including absolute and relative errors: Mean Square Error (MSE):

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (6)$$

Fig. 2 Comparison Table of Prediction Performance of Different Models

Model	MAE	MSE	MAPE (%)	SMAPE (%)
Informer	0.120	0.031	4.942	4.689
DLinear	0.181	0.054	6.665	6.799
Autoformer	0.215	0.064	8.025	8.238
PatchTST	0.214	0.075	7.448	7.831
FEDformer	0.251	0.098	9.011	9.484
NTSForecast (ours)	0.102	0.019	3.667	3.580

mean absolute error (MAE) :

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (7)$$

Mean Percentage Error (MPE):

$$MPE = \frac{1}{n} \sum_{i=1}^n \frac{y_i - \hat{y}_i}{y_i} \times 100\% \quad (8)$$

Mean Absolute Percentage Error (SMAPE):

$$SMAPE = \frac{1}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{\frac{|y_i| + |\hat{y}_i|}{2}} \times 100\% \quad (9)$$

Here, y_i represents the predicted load value, \hat{y}_i represents the actual load value, and n denotes the number of samples for model prediction.

4. CASE ANALYSIS

An analysis was conducted on the actual load of a large city in northern China, selecting the paradigm models of time series forecasting in recent years, including Autoformer, Dlinear, PatchTST, Informer, and FEDformer, and comparing them with the method proposed in this paper. As shown in the figure below, the method proposed in this paper can achieve good load forecasting results.

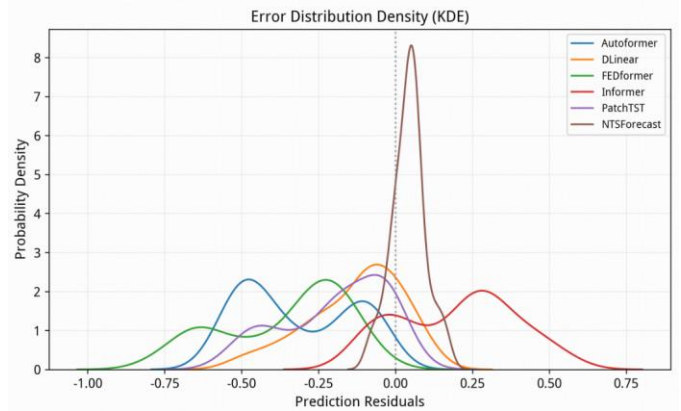


Fig. 6 Error distribution plots of different models

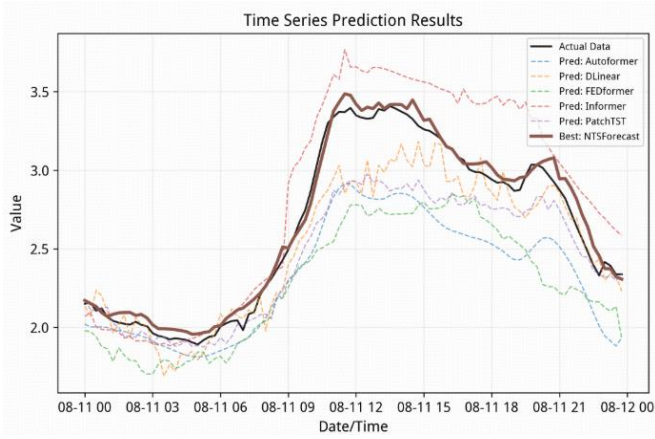


Fig. 7 Prediction diagram of data in steady state

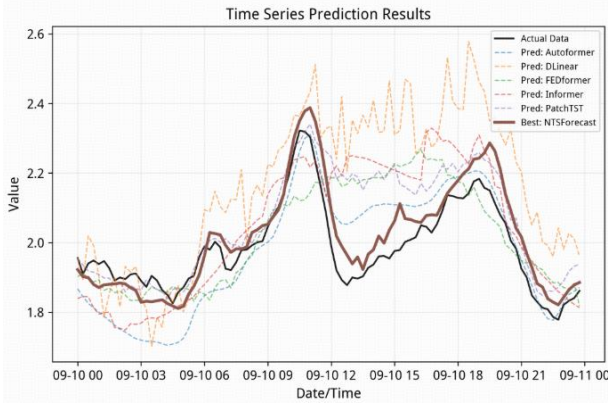


Fig. 8 Prediction chart for data fluctuations

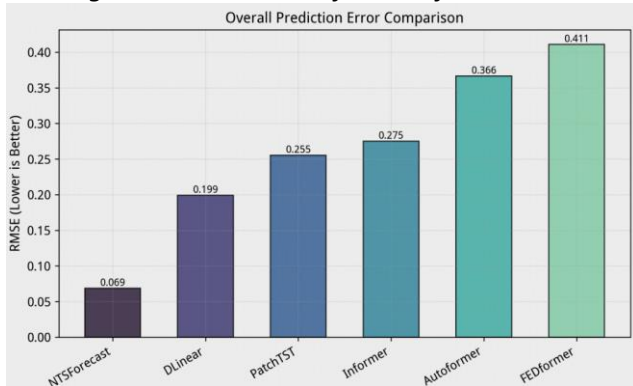


Fig. 9 Comparison of prediction performance across different models

5. CONCLUSION

This paper presents a load forecasting method for digital text separation. By freezing the core parameters of large language models and introducing only a few fine-tuning layers, the method achieves the separate extraction and fusion of digital and text features. Tested on actual power load data, it demonstrates excellent forecasting performance.

ACKNOWLEDGEMENT

This project was supported by the National Natural Science Foundation of China (NSFC), with the project number 52307132.

REFERENCE

- [1] Jin M, Wang S, Ma L, et al. Time-LLM: Time Series Forecasting by Reprogramming Large Language Models. ICLR 2024.
- [2] Wang Z, Li Y, Li K, Li D. Time-VLM: Exploring Multimodal Vision-Language Models for Augmented Time Series Forecasting. arXiv:2502.04395, 2025.
- [3] Chen L, Wang Z, Liu Y. MLLM4TS: Leveraging Vision and Multimodal Language Models for General Time-Series Analysis. arXiv:2510.07513, 2025.
- [4] Sun S, Chen Y. Enhancing LLMs for Time Series Forecasting via Structure-Guided Cross-Modal Alignment. arXiv:2505.13175, 2025.
- [5] Liu C, Xu Q, Miao H, et al. TimeCMA: Towards LLM-Empowered Multivariate Time Series Forecasting via Cross-Modality Alignment. AAI 2025.
- [6] Zhang X, Chowdhury RR, Gupta RK, Shang J. Large Language Models for Time Series: A Survey. arXiv:2402.03182, 2024.
- [7] Xiao C, Zhou J, Xiao Y, et al. TimeFound: A Foundation Model for Time Series Forecasting. arXiv, 2025.
- [8] Liu C, Zhou S, Xu Q, et al. Towards Cross-Modality Modeling for Time Series Analytics: A Survey in the LLM Era. arXiv, 2025.
- [9] Goswami M, Szafer J, Choudhary A, et al. MOMENT: A Family of Open Time-series Foundation Models. arXiv, 2024.
- [10] Ansari AF, Stella L, Turkmen AC, et al. Chronos: Learning the Language of Time Series. TMLR, 2024.
- [11] Tang H, Zhang C, Jin M, et al. Time Series Forecasting with LLMs: Understanding and Enhancing Model Capabilities. arXiv, 2025.
- [12] Chen L, Wang Z, Liu Y. ChatTime: A Unified Multimodal Time Series Foundation Model Bridging Numerical and Textual Data. AAI 2025.
- [13] Xue H, Salim FD. PromptCast: A New Prompt-Based Learning Paradigm for Time Series Forecasting. IEEE TKDE, 2023.
- [14] Zhou B, Kwon S, Zhang X, et al. One Fits All: Power General Time Series Analysis by Pretrained LM. NeurIPS 2023.
- [15] Zheng L, Dong C, Zhang W, et al. Revisiting Large Language Model for Time Series Analysis through Modality Alignment. arXiv:2410.12326, 2024.