A General Framework for Load Forecasting based on Pre-trained Large Language Model

Mingyang Gao^a, Suyang Zhou^a, Wei Gu^a, Zhi Wu^a, Haiquan Liu^a, Aihua Zhou^b

 $^a Southeast\ University,\ Nanjing,\ Jiangsu,\ China$ $^b China\ Electric\ Power\ Research\ Institute,\ Nanjing,\ Jiangsu,\ China$

Abstract

Accurate load forecasting is crucial for maintaining the power balance between generators and consumers, particularly with the increasing integration of renewable energy sources, which introduce significant intermittent volatility. With the advancement of data-driven methods, machine learning and deep learning models have become the predominant approaches for load forecasting tasks. In recent years, pre-trained large language models (LLMs) have achieved significant progress, demonstrating superior performance across various fields. This paper proposes a load forecasting method based on LLMs, offering not only precise predictive capabilities but also broad and flexible applicability. Additionally, a data modeling method is introduced to effectively transform load sequence data into natural language suitable for LLM training. Furthermore, a data enhancement strategy is designed to mitigate the impact of LLM hallucinations on forecasting results. The effectiveness of the proposed method is validated using two real-world datasets. Compared to existing methods, our approach demonstrates state-of-the-art performance across all validation metrics.

Keywords: Load forecasting, pre-trained language model, deep learning

1. Introduction

1.1. Motivation

Load forecasting has been playing an important role in maintaining the stability of the modern power system [1, 2]. With accurate forecasting, the power system can optimize the integration of variable renewable energy sources [3, 4]. As the advancement of data-driven algorithms, machine learning and deep learning-based methods have become the predominant approaches for load forecasting tasks, owing to their exceptional performance [5].

More recently, large language models (LLMs) have emerged, demonstrating strong accuracy and flexibility across various research in natural language process (NLP) tasks [6, 7]. The attention mechanism in LLMs models have been proven to be effective in capturing the long-range dependencies in time series data, which is beneficial for load forecasting tasks [8]. However, there is currently no research utilizing LLMs for load forecasting tasks. The challenge lies in effectively considering the characteristics of electrical loads in conjunction with the superior knowledge comprehension and reasoning capabilities of LLMs to enhance the accuracy and reliability of predictions. This remains a promising issue that requires further investigation.

1.2. Literature Review

Load forecasting is a task within the domain of time series forecasting, and various networks have been continuously adapted to promote the prediction accuracy, such as eXtreme Gradient Boosting (XGBoost) [9], Long Short-Term Memory (LSTM) [10] and Transformers [11]. Notably, the Transformer architecture and its subsequent variants have demonstrated superior performance. Informer [12] proposes the Prob-Sparse self-attention mechanism, which reduces time complexity while maintaining performance. It employs self-attention distilling to manage long input sequences effectively. Autoformer [13] introduces an autocorrelation mechanism by utilizing the pre-processing convention of series decomposition, transforming it

into a fundamental component within models. Fedformer [14] introduces an approach that integrates Transformer models with seasonal-trend decomposition techniques and leverage the typically sparse representation of time series data in the Fourier transform. Additionally, in DLinear [15], researchers argue that accuracy is not primarily determined by the network architecture. Instead, the decomposition and processing of data significantly enhance the accuracy of the predictive models. By employing a simple fully-connected network with decomposition techniques , they are able to achieve similarly satisfactory prediction accuracy.

Consequently, beyond the model itself, the quality and distribution of data used for training above models also make a difference to the forecasting accuracy. To ensure the model performs optimally on general task, data modeling and feature engineering strategies designed for specific model are often proposed at the same time [16]. Reference [17] adapts a XGBoost-based scheme for electricity load forecasting through increasing number of features available and converting daily electricity load information into weekly load information. In [18, 19], a residential load forecasting framework based on the LSTM is described with an customer-wise level data analysis.

Some ongoing research are already applying LLMs on time series forecasting tasks and obtain competitive forecasting results. Reference [20] introduces a prompt-based learning paradigm for time series forecasting based on LLMs and shows superior performance across three distinct scenarios. However, data missing is observed during the prediction process. This issue arises from the hallucination problem inherent in LLMs [21, 22]. In load forecasting tasks, the hallucination may lead to extremely inaccurate predictions or missing values in the output sequence, but there is few research on how to effectively solve the problem. Reference [23] also employs LLMs as the predictor, but they keeps the parameter of LLMs static and completes the training by updating the forward reprogramming layer. Reference [24] leverages the reasoning ability of LLMs for accurate wind speed forecasting with spatio-temporal information.

1.3. Contributions and Paper Organization

In order to resolve the above deficiencies, this paper proposes a load forecasting framework based on pre-trained LLMs, leveraging its flexibility and generalizability to achieve more accurate results on multi-time-scale and multi-scenario datasets. Also, the paper introduces a dataset modeling method that enables LLMs to perform effectively.

The specific contributions of this research are as follows:

- 1. We propose a general and flexible load forecasting method based on pre-trained LLMs. The proposed method can be applied to multi-timescale and multi-scenario load forecasting tasks.
- 2. A dataset formulation method that combine language with statistical information is introduced to better leverage the predictive capabilities of LLMs.
- 3. A data enhancement method is devised for solving the hallucination problems of LLMs by separating numerical sequence with language descriptions.
- 4. The effectiveness of the proposed method is validated across open-sources and real-world load fore-casting datasets with different time scales. Compared with existing load forecasting methods, the superiority and adaptability of the proposed framework is clearly proved.

The paper is structured as follows: Section 2 outlines the dataset modeling approach for load forecasting tasks utilizing pre-trained LLMs. Section 3 details the proposed load forecasting framework, encompassing the backbone models employed, training strategies, and evaluation metrics. Section 4 presents case studies to validate the effectiveness of the proposed methods. Finally, Section 5 concludes the paper and discusses potential directions for future research.

2. Dataset Formulation and Enhancement

In this section, we present a method for creating datasets for language models. Starting with converting numerical data into textual data, we will detail the approach through which we can effectively used the data in. Moreover, a technique to address the hallucination phenomenon is also introduced. To emphasize, the proposed dataset modeling method is applicable to all load forecasting tasks based on language models.

Table 1: The example of dataset based on proposed method

| Input Data | $L \times d$ | Example | $\operatorname{Ground-truth}$ | | |
|------------|-------------------------|--|--|--|--|
| X_{text} | ELFD: 7×1 ICLD: 24×1 | The electricity consumption of each day is as follows, 29979,29415,27958,25579,28112,29664,29516kWh. What is the daily consumption of next week? | The electricity consumption of each day is as follows, 22992,21895,26303,28286, 28727,26488,24839kWh. | | |
| X_{ts} | ELFD: 7×1 ICLD: 24×1 | The electricity consumption of each day is as follows, 29979,29415,27958,25579,28112,29664, 29516kWh. The maximum value is 32123, the minimum value is 20321, the average value is 28603. What is the daily consumption of next week? | The electricity consumption of each day is as follows,22992,21895,26303,28286, 28727,26488,24839kWh. | | |
| X_{ets} | ELFD: 7×1 ICLD: 24×1 | The electricity consumption of day one is 29979, the electricity consumption of day two is 29415, the electricity consumption of day three is 27958, the electricity consumption of day four is 25579, the electricity consumption of day five is 28112, the electricity consumption of day six is 29664, the electricity consumption of day seven is 29516. The maximum value is 32123, the minimun value is 20321, the average value is 28603. What is the daily consumption of next week? | The electricity consumption of day one is 22992, the electricity consumption of day two is 21895, the electricity consumption of day three is 26303, the electricity consumption of day four is 28286, the electricity consumption of day five is 28727, the electricity consumption of day six is 26488, the electricity consumption of day seven is 24839. | | |

For datasets in other languages, we use Google Translate to generate corresponding input data and Ground-truth in the identical format.

2.1. Combine Language with Statistical Information

In common load forecasting tasks, historical load data are always employed as the input for forecasting. The input data are typically modelled into a continuous sequence $X \in \mathbb{R}^{L \times d}$, where L and d represents the length and dimension of the sequence, respectively. Since data is required to be input in text format for language models, we propose a dataset modeling method that convert numerical sequence into natural language expression X_{text} as described below:

$$X_{text} = \mathbb{S}(X) = \{ \mathbb{S}(x_1) \dots \mathbb{S}(x_i) \dots \mathbb{S}(x_n) \} \qquad \text{for } 1 \le i \le n$$
 (1)

where x_i is the *i*-th data in the input sequence, \mathbb{R} represents the set of real number, \mathbb{S} stands for the transformation from real number to text.

Additionally, to further exploit the advantages of textual expression and inspired by references [15], which demonstrate that data decomposition and processing significantly enhance predictive model accuracy, we introduce statistical information X_{stat} to enhance the feature dimension of the input data, denoted as X_{ts} .

The statistical information includes maximum, minimum, and average values to cover the local and global. Specifically, we use the maximum and minimum values within the range of N_{obs} steps before the predicted time to model global features, and represent the local features with the average value of the input sequence.

$$\begin{cases}
X_{ts} = \{X_{text}, X_{stat}\} \\
X_{stat} = \{Max(X_{obs}), Min(X_{obs}), Average(X)\}
\end{cases}$$
(2)

where X_{ts} represents the input with statistical information, X_{stat} is the statistical information with language descriptions including the maximum, minimum and average value of X, X_{obs} demonstrate the historical load data within the range of N_{obs} time-steps before the predicted time.

2.2. Separate Numerical Sequence with Language

The causes of hallucination in load forecasting tasks, such as missing data or generating extra data, can be attributed to two primary aspects: 1) during the conversion of numerical data to textual descriptions, the lengths of loaded data stored in string format exhibit inconsistency. 2) the pre-training parameters of LLMs are derived from training on natural language, thereby lacking the capability to effectively recognize purely numerical values.

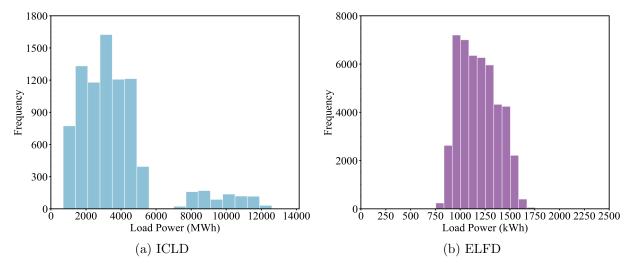


Fig. 1: Distribution of involved load forecasting datasets

Taking advantages of LLM's sensitivity to language descriptions, this section proposes a data enhancement method that separates numerical data with textual information. The enhanced input dataset X_{ets} is constructed based on X_{text}^* as shown below:

$$\begin{cases} X_{ets} = \{X_{text}^*, X_{stat}\} \\ X_{text}^* = \{(t_1, x_1) \dots (t_i, x_i) \dots (t_n, x_n)\} & for \ 1 \le i \le n \\ X_{stat} = \{Max(X_{obs}), Min(X_{obs}), Average(X)\} \end{cases}$$
 (3)

where X_{text}^* is the textual expression of the numerical sequence with time information, t_i is the *i*-th corresponding time-steps in textual expression of x_i .

Given that the output data from LLMs also exists in text form, the textual ground-truth is necessary for the training process consequently. Following the same process for each input format of X, we generate the corresponding ground-truth Y_{gt} .

2.3. Dataset for Forecasting

To evaluate the generality and accuracy of the proposed methods in load forecasting tasks, we selected the following two real-world datasets at different time scales to perform our research.

Industrial Clients Load Dataset (ICLD): This real-world dataset comprises around 9000 daily load data on the electricity consumption of the 10 industrial clients from June 1st, 2018 to June 25th, 2021. All data is collected from a real-world city-level power system in east China. The time length of the training/validation/test set is 24/6/6 months, respectively. The average and standard deviation value of ICLD is 3695.10 and 2334.11.

Electricity Load Forecasting Dataset (ELFD): This is an open-source dataset available on Kaggle¹, covering over 40,000 hourly load data for the Panama region from January 31st, 2015 to June 10th, 2020. The time length of the training/validation/test set is 48/12/6 months, respectively. The average and standard deviation value of ICLD is 1184.82 and 192.26.

The distribution of two dataset is visualized in Figure 1. Dataset with detailed examples under the strategies established in this section is shown in Table 1. The effectiveness of proposed methods above are validated in Section 4.

 $^{^{1}} kaggle.com/datasets/saurabhshahane/electricity-load-forecasting/datasets/saurabhshahane/electricity-load-forecasting/datasets/saurabhshahane/electricity-load-forecasting/datasets/saurabhshahane/electricity-load-forecasting/datasets/saurabhshahane/electricity-load-forecasting/datasets/saurabhshahane/electricity-load-forecasting/datasets/saurabhshahane/electricity-load-forecasting/datasets/saurabhshahane/electricity-load-forecasting/datasets/saurabhshahane/electricity-load-forecasting/datasets/saurabhshahane/electricity-load-forecasting/datasets/saurabhshahane/electricity-load-forecasting/datasets/saurabhshahane/electricity-load-forecasting/datasets/saurabhshahane/electricity-load-forecasting/datasets/saurabhshahane/electricity-load-forecasting/datasets/saurabhshahane/electricity-load-forecasting/datasets/saurabhshahane/electricity-load-forecasting/saurabhshahane/electricity-load-fore$

3. Proposed Framework

In this section, we provide a detailed introduction to the basic structure of our proposed prediction framework and the LLMs used to complete the prediction task. Additionally, we detail training methods for different LLMs and list the metrics used to evaluate their forecasting results.

3.1. Multi-Head Attention Mechanism within LLMs

The Multi-Head Attention mechanism comprises three primary components: linear projections, scaled dot-product attention, and concatenation followed by linear transformation, which enables the model to process input data from multiple perspectives simultaneously, enhancing its ability to capture complex patterns and relationships within the sequence.

1)Linear Projections: For each attention head, the input vectors are linearly projected into three distinct spaces to create queries Q_i , keys K_i , and values V_i , allowing each head to focus on different aspects of the input data.

$$Q_i = QW_i^Q K_i = KW_i^K V_i = VW_i^V (4)$$

where W_i^Q , W_i^K , and W_i^V are the linear projection matrices for each head.

2)Scaled Dot-Product Attention: Each attention head computes attention scores and passed through a softmax function to obtain attention weights, which are used to compute a weighted sum of the value vectors.

$$head_i = \text{Attention}(Q_i, K_i, V_i) = \text{softmax}(\frac{Q_i K_i^T}{\sqrt{d_k}}) V_i$$
 (5)

where $head_i$ denotes the output of the *i*-th attention head, d_k is the dimension of the key vectors and $1/\sqrt{d_k}$ stands for the scaling factor.

3)Concatenation and Linear Transformation: The outputs from all attention heads are concatenated and passed through a final linear transformation, which combines the diverse information captured by each head into a single output representation.

$$MultiHead(Q_i, K_i, V_i) = Concat(head_1, ..., head_h)W^O$$
(6)

where W^O is the linear transformation matrix for the output of network, h is the number of attention heads.

3.2. LLMs for Load Forecasting

LLMs could be structurally categorized into three types:

Encoder-Only Models: Represented by BERT [25], these models learn bidirectional context encoders through masked language modeling. The training objective involves randomly masking parts of the text and predicting the masked words. This architecture is mainly suitable for tasks that do not require sequence generation but instead need to encode and process input, such as text classification and sentiment analysis.

Decoder-Only Models: Represented by GPT [26] and BLOOM [27], these models are typically used for sequence generation tasks and known as generative model. It generates sequences directly from the input and perform unsupervised pre-training. However, they require tremendous training data to improve the quality and diversity of generated text.

Encoder-Decoder Models: Represented by T5 [28] and BART [29], these models use an encoder to process the input sequence, extracting features and semantic information, and a decoder to generate the corresponding output sequence. Known as sequence-to-sequence model, it experts in handling the relationship between input and output sequences, improving accuracy in tasks like machine translation and dialogue generation.

Depending on the characteristics of load forecasting tasks, we primarily considers LLMs based on Decoder-only and Encoder-Decoder architectures. In light of the models discussed in references [20, 24], we selected several open-source LLMs for load forecasting. The configurations for these models are presented in Table 2. Furthermore, LLMs trained in different languages are selected to verify whether the forecasting result is influenced by natural language expression.

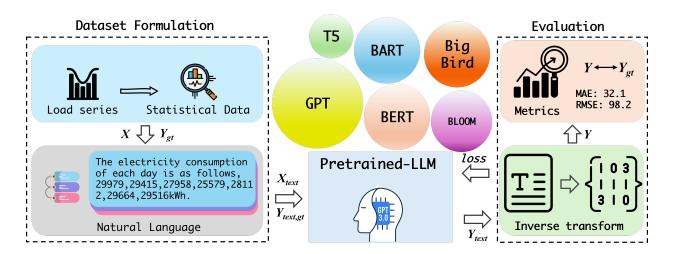


Fig. 2: The architecture of the proposed framework.

| Table 2: | The LLN | I Models | used in | the Pro | posed Framework |
|----------|---------|----------|---------|---------|-----------------|
|----------|---------|----------|---------|---------|-----------------|

| Model | Pre-trained Language | Access Key | Model Size | |
|--------------------------|----------------------|--|------------|--|
| GPT2 | English | openai-community/gpt2 | 548MB | |
| BART | English | facebook/bart-base | 558MB | |
| BART-CN | Chinese | ${\rm fnlp/bart\text{-}base\text{-}chinese}$ | 561MB | |
| T5 | English | google-t5/t5-base | 892MB | |
| Mengzi-T5 | Chinese | Langboat/mengzi-t5-base | 990MB | |
| $\operatorname{BigBird}$ | English | google/bigbird-pegasus- large-arxiv | 2.3GB | |
| BLOOM | English | bigscience/bloom-1b7 | 3.4GB | |
| BLOOM-CN | Chinese | Langboat/bloom-1b4-zh | 5.6GB | |

3.3. Training Strategies for LLMs

3.3.1. Parameter-Efficient Fine-Tuning (PEFT)

Large language models pre-trained for general tasks, encode a comprehensive understanding of knowledge within their pre-trained parameters. Consequently, training these models completely on specialized datasets will destroy the distribution pattern of pre-trained parameters, reducing their feasibility in text comprehension. Therefore, we adoption PEFT method with the Low-Rank Adaptation of Large Language Models (LoRA) technique to delicately fine-tune model parameters [30]. In this method, we use low-rank decomposition to simulate parameter changes based on the original model's parameter distribution, thereby indirectly training a large model with a minimal number of parameters. We process the selected parameter matrix $W_{d\times k}$ from the original model as follows:

$$W_{d \times k} = U_{d \times r} \cdot V_{r \times k} \qquad r \ll d, k \tag{7}$$

where r is the Low-rank coefficient, U and V are the low-rank matrices.

In our research, the parameters selected for PEFT are linear transform layers and attention layers. The total amount of trainable parameters takes up to 10% of the original model.

3.3.2. Full Parameter Training

Fully parameterized training method under our proposed framework is also served to train LLMs with the proposed dataset. While this approach trades off original problem-solving capabilities for the utilization of pre-trained parameters, it demonstrates notable efficacy in load forecasting tasks.

3.4. Evaluation Method and Metrics

For the model's prediction results, we mostly care about the accuracy of the numerical sequence within natural language. According to the format setting of ground-truth in Section 2, we can easily extract the data sequence from the text, with which we can calculate the forecasting accuracy to analyse the performance of the model.

Hallucination Rate is proposed to evaluate the hallucination in the forecasting results. Together with the Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), three metrics are served as evaluation metrics in our research and defined as follows,

$$\begin{cases}
H = n_h/N \\
MAE = \frac{1}{N} \sum_{i=1}^{N} |Y_i - Y_{i,gt}| \\
RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (Y_i - Y_{i,gt})^2}
\end{cases}$$
(8)

where N is the number of samples, Y_i is the *i*-th predicted result, $Y_{i,gt}$ is the corresponding ground-truth, $H \in (0,1)$ is the Hallucination Rate and n_h is the number of hallucination samples.

The framework of our work is depicted in Figure 2. To fully leverage the pre-trained parameters in large models, we adopt diverse training approaches for various LLMs, aiming to achieve optimal prediction results while maintaining training efficiency.

Table 3: Hyperparameters of Proposed and Comparison Methods

| Method | Hyperparameters | Value |
|----------|--------------------------|-----------|
| | batch size | 32 |
| | Learning rate | $5e^{-5}$ |
| | Input length of ICLD | 7 |
| | Output length of ICLD | 7 |
| LLM | Input length of ELFD | 24 |
| | Output length of ELFD | 24 |
| | LoRA coefficient | 8 |
| | LoRA alpha | 32 |
| | LoRA dropout | 0.1 |
| | Number of estimators | 160 |
| XGBoost | Learning rate | 0.001 |
| | Max depth | 10 |
| | Number of layers | 10 |
| | Hidden size | 128 |
| LSTM | Dropout rate | 0.2 |
| | Batch size | 32 |
| | Learning rate | 0.001 |
| | Number of heads | 8 |
| | Moving average step | 12 |
| X-former | Number of Enc/Dec layers | 2/1 |
| | Batch size | 32 |
| | Learning rate | 0.001 |
| | Kernel size | 25 |
| Dlinear | Individual | 0 |
| Dimear | Batch size | 32 |
| | Learning rate | 0.001 |

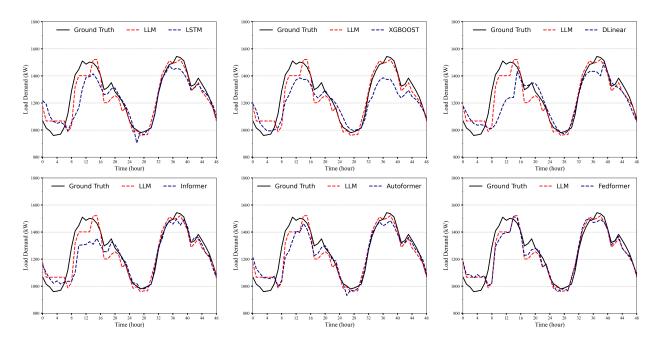


Fig. 3: Forecasting results comparison of the proposed method with other models on ELFD.

4. Case study

In this section, we will validate the effectiveness of the proposed methods. Firstly, we will demonstrate the physical environment and hyperparameter configurations employed during training. Secondly, we will apply the forecasting framework to two datasets introduced in Section 2. Mengzi-T5 model as a representation of LLMs, will undergo an in-depth evaluation of its performance and statistical outcomes compared with traditional methods. Furthermore, the various LLMs mentioned in Section 3 will be tested to confirm their capabilities in the prediction task.

4.1. Parameters Configuration

Our model is implemented using PyTorch and Transformers from HuggingFace, with all experiments conducted on NVIDIA 4090-24G GPUs. All of our models can be accessed with the Access Key in Table 2 from HuggingFace Model Hub [31]. The hyperparameters of the proposed framework and comparison methods are shown in Table 3.

4.2. Case1: Forecasting Results of Different Time-scale Datasets

In the validation sets of ICLD and ELFD, with lengths of 6 and 12 months respectively, we calculate the Hallucination Rate, MAE and RMSE of the predicted data. The hallucination of LLMs may result in missing or excess issues with the results. To ensure the calculation of metrics, we address this problem as follows: 1)missing data is handled by supplementing it with zeros, 2)additional data is removed to keep the same length of all output sequences. We employ GPT2 model for our framework and compare it with traditional methods including XGBoost, LSTM, Informer, Autoformer, Fedformer and DLinear.

As depicted in Table 4, our method shows state-of-the-art performance compared to all traditional prediction methods. The LLM, LLM-ts, and LLM-ets methods are only different in the input data format corresponding to X_{text} , X_{ts} , and X_{ets} . The prediction results suggest that integrating long-vision statistical information into the data enhances prediction accuracy. To provide a intuitional presentation, forecasting curves of the our framework compared with other methods are visualized in Figure 3.

Notably, the red data in Table 4 highlights the impact of the hallucination problem within forecasts, particularly in the ICLD dataset, where it results in a significantly higher RMSE than normal values. We

confirm that this issue arises from a missing value in the predicted data sequence. With X_{ets} as the input data, the hallucination rate is reduced to zero, and the MAE and RMSE are also significantly improved.

The forecasting result demonstrates that without preprocessing the original data, the predictive capability of LLM is under-explored. With the proposed method in Section 2, LLMs can effectively eliminate hallucination and improve the forecasting accuracy. In contrast to traditional model data preprocessing methods, the approach based on pre-trained LLMs leverages decomposition in a more straightforward and efficient way by simply incorporating statistical information into the prompts.

Table 4: Comparison between Different Methods and Proposed Framework

| Method | Input Data | ICLD | | | ELFD | | |
|-------------|------------|--------------------|-------------|---------|--------------------|--------|--------|
| | Inpat Data | Hallucination Rate | MAE | RMSE | Hallucination Rate | MAE | RMSE |
| XGBoost | X | 0 | 130.50 | 219.54 | 0 | 83.35 | 113.68 |
| LSTM | X | 0 | 172.44 | 294.69 | 0 | 85.08 | 117.52 |
| Informer | X | 0 | 92.71 | 167.68 | 0 | 63.15 | 78.51 |
| Autoformer | X | 0 | 87.89 | 150.18 | 0 | 61.15 | 79.89 |
| Fedformer | X | 0 | 82.01 | 126.31 | 0 | 57.15 | 74.37 |
| Dlinear | X | 0 | 99.40 | 189.73 | 0 | 74.25 | 106.51 |
| $_{ m LLM}$ | X_{text} | 0.035 | 264.53 | 2987.19 | 0.085 | 203.25 | 457.62 |
| LLM-ts | X_{ts} | 0.022 | 239.60 | 2182.43 | 0.016 | 116.05 | 364.31 |
| LLM-ets | X_{ets} | 0 | $\bf 80.62$ | 112.59 | 0 | 52.44 | 73.03 |

4.3. Case2: Forecasting Results based on Different LLMs Models

We validated the generality of our approach on LLMs with different backbones given in Table 2. As shown in Table 5, the proposed prediction framework consistently achieves low MAE and RMSE across different LLMs, and the GPT2 model shows the best prediction performance on both datasets.

Additionally, we conducted comparative experiments using two state-of-the-art LLM framework, GPT-4 and Claude 3.5. For these models, we obtained predictions by inputting prompts without any additional training. The results indicate that LLMs demonstrate a certain level of predictive capability even without being trained on specific datasets. However, even the most advanced commercially available LLMs, when not fine-tuned on task-specific datasets, underperform compared to models with fewer parameters. This result further validates the efficacy of our proposed methodology.

Table 5: Comparison between Different LLM backbones

| Model | IC | LD | ELFD | |
|-------------------------------|--------|--------|-------|--------|
| Model | MAE | RMSE | MAE | RMSE |
| GPT40 (Without training) | 151.58 | 178.05 | 86.27 | 95.76 |
| Claude 3.5 (Without training) | 103.80 | 151.92 | 82.61 | 100.14 |
| GPT2 | 80.62 | 112.59 | 52.44 | 73.03 |
| T5 | 84.59 | 139.05 | 59.65 | 80.23 |
| Mengzi-T5 | 80.10 | 104.01 | 60.01 | 83.87 |
| BART | 91.58 | 216.3 | 66.27 | 92.19 |
| BART-CN | 93.37 | 159.02 | 67.31 | 90.51 |
| $\operatorname{BigBird}$ | 102.93 | 187.23 | 65.16 | 88.47 |
| BLOOM | 99.17 | 238.01 | 62.25 | 95.98 |
| BLOOM-CN | 109.74 | 247.08 | 62.31 | 98.05 |

5. Conclusion

In this paper, a general and flexible load forecasting framework based on pre-trained language models is proposed. The following conclusions can be drawn:

- 1. A dataset formulation approach is established to convert sequence-formatted data into natural language to facilitate LLM training and language descriptions of statistical information is integrated for broaden the input feature dimension.
- 2. A data enhancement method is accordingly proposed to address the hallucination problem of LLMs in load prediction tasks. With the proper separation of numerical sequence and language descriptions, the hallucination rate is significantly reduced to 0%.
- 3. The comprehensive predictive performance of our method is validated on two real-world datasets. The MAE is reduced to 80.10 and 52.44 on ICLD and ELFD respectively, demonstrating superior prediction accuracy over existing methods.

In future work, we aim to apply larger language models on load prediction problems. We will focus on establishing datasets and developing training methods suitable for large language models, ensuring reliable load prediction while maximizing the utilization of pre-trained parameters.

CRediT Authorship Contribution Statement

Mingyang Gao: Conceptualization, Methodology, Software, Formal analysis, Writing - Original draft preparation. Suyang Zhou: Conceptualization, Investigation, Supervision, Writing - Reviewing and Editing, Funding acquisition. Wei Gu: Resources, Validation, Data curation, Project administration. Zhi Wu: Supervision, Writing - Reviewing and Editing. Haiquan Liu: Supervision. Aihua Zhou Supervision.

Data availability

Public datasets are used. The data can be accessed with the following URL: https://www.kaggle.com/datasets/saurabhshahane/electricity-load-forecasting/data.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work is supported by the Science and Technology Project of State Grid under the grant number 5700-202458232A-1-1-ZN (Corresponding author: Suyang Zhou).

References

- [1] P. Kundur, Power system stability, Power system stability and control 10 (2007) 7–1.
- [2] Y. Wang, C. Chen, J. Wang, R. Baldick, Research on resilience of power systems under natural disasters—a review, IEEE Transactions on Power Systems 31 (2) (2016) 1604–1613. doi:10.1109/TPWRS.2015.2429656.
- [3] A. S. Brouwer, M. Van Den Broek, A. Seebregts, A. Faaij, Impacts of large-scale intermittent renewable energy sources on electricity systems, and how these can be modeled, Renewable and Sustainable Energy Reviews 33 (2014) 443–466.
- [4] K. Guerra, P. Haro, R. Gutiérrez, A. Gómez-Barea, Facing the high share of variable renewable energy in the power system: Flexibility and stability requirements, Applied Energy 310 (2022) 118561.
- [5] M. Cai, M. Pipattanasomporn, S. Rahman, Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques, Applied energy 236 (2019) 1078–1088.
- [6] H. Wang, J. Li, H. Wu, E. Hovy, Y. Sun, Pre-trained language models and their applications, Engineering (2022).
- [7] B. Min, H. Ross, E. Sulem, A. P. B. Veyseh, T. H. Nguyen, O. Sainz, E. Agirre, I. Heintz, D. Roth, Recent advances in natural language processing via large pre-trained language models: A survey, ACM Computing Surveys 56 (2) (2023) 1–40.
- [8] Z. Fazlipour, E. Mashhour, M. Joorabian, A deep model for short-term load forecasting applying a stacked autoencoder based on lstm supported by a multi-stage attention mechanism, Applied Energy 327 (2022) 120063.
- [9] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016, pp. 785–794.

- [10] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (8) (1997) 1735–1780.
- [11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).
- [12] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, W. Zhang, Informer: Beyond efficient transformer for long sequence time-series forecasting, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 35, 2021, pp. 11106–11115.
- [13] H. Wu, J. Xu, J. Wang, M. Long, Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting, Advances in neural information processing systems 34 (2021) 22419–22430.
- [14] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, R. Jin, Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting, in: International conference on machine learning, PMLR, 2022, pp. 27268–27286.
- [15] A. Zeng, M. Chen, L. Zhang, Q. Xu, Are transformers effective for time series forecasting?, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 37, 2023, pp. 11121–11128.
- [16] H. Liu, C. Chen, Data processing strategies in wind energy forecasting models and applications: A comprehensive review, Applied Energy 249 (2019) 392–408.
- [17] R. A. Abbasi, N. Javaid, M. N. J. Ghuman, Z. A. Khan, S. Ur Rehman, Amanullah, Short term load forecasting using xgboost, in: Web, Artificial Intelligence and Network Applications: Proceedings of the Workshops of the 33rd International Conference on Advanced Information Networking and Applications (WAINA-2019) 33, Springer, 2019, pp. 1120–1131.
- [18] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, Y. Zhang, Short-term residential load forecasting based on lstm recurrent neural network, IEEE transactions on smart grid 10 (1) (2017) 841–851.
- [19] W. Kong, Z. Y. Dong, D. J. Hill, F. Luo, Y. Xu, Short-term residential load forecasting based on resident behaviour learning, IEEE Transactions on power systems 33 (1) (2017) 1087–1088.
- [20] H. Xue, F. D. Salim, Promptcast: A new prompt-based learning paradigm for time series forecasting, IEEE Transactions on Knowledge and Data Engineering (2023).
- [21] Z. Xu, S. Jain, M. Kankanhalli, Hallucination is inevitable: An innate limitation of large language models, arXiv preprint arXiv:2401.11817 (2024).
- [22] H. Ye, T. Liu, A. Zhang, W. Hua, W. Jia, Cognitive mirage: A review of hallucinations in large language models, arXiv preprint arXiv:2309.06794 (2023).
- [23] M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y.-F. Li, S. Pan, et al., Time-llm: Time series forecasting by reprogramming large language models, arXiv preprint arXiv:2310.01728 (2023).
- [24] T. Wu, Q. Ling, Stellm: Spatio-temporal enhanced pre-trained large language model for wind speed forecasting, Applied Energy 375 (2024) 124034. doi:https://doi.org/10.1016/j.apenergy.2024.124034. URL https://www.sciencedirect.com/science/article/pii/S030626192401417X
- [25] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, CoRR abs/1810.04805 (2018). arXiv:1810.04805. URL http://arxiv.org/abs/1810.04805
- [26] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language models are unsupervised multitask learners, 2019. URL https://api.semanticscholar.org/CorpusID:160025533
- [27] T. L. Scao, A. Fan, C. Akiki, Bloom: A 176b-parameter open-access multilingual language model, ArXiv abs/2211.05100 (2022).
 - URL https://api.semanticscholar.org/CorpusID:253420279
- [28] Z. Zhang, H. Zhang, K. Chen, Y. Guo, J. Hua, Y. Wang, M. Zhou, Mengzi: Towards lightweight yet ingenious pre-trained models for chinese (2021). arXiv:2110.06696.
- [29] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, CoRR abs/1910.13461 (2019). arXiv:1910.13461. URL http://arxiv.org/abs/1910.13461
- [30] N. Ding, Y. Qin, G. Yang, F. Wei, Z. Yang, Y. Su, S. Hu, Y. Chen, C.-M. Chan, W. Chen, et al., Parameter-efficient fine-tuning of large-scale pre-trained language models, Nature Machine Intelligence 5 (3) (2023) 220–235.
- [31] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Brew, Huggingface's transformers: State-of-the-art natural language processing, CoRR abs/1910.03771 (2019). arXiv:1910.
 - URL http://arxiv.org/abs/1910.03771