

Article

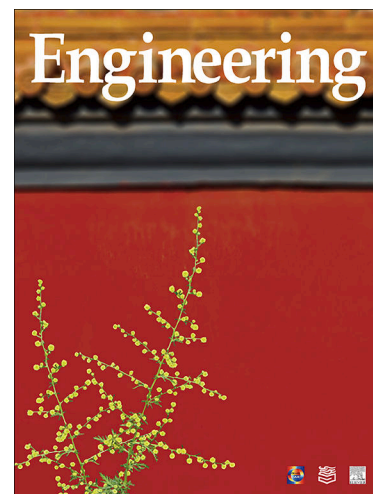
News-Driven Load Forecasting: Generative Agents and Large Language Models for Unstructured Data and Event Analysis

Xinlei Wang, Jinjin Gu, Jing Qiu, Guolong Liu, Xinlei Cai, Jinzhou Zhu, Yanli Liu, Zhaoyang Dong, Junhua Zhao

PII: S2095-8099(26)00175-X
DOI: <https://doi.org/10.1016/j.eng.2026.02.031>
Reference: ENG 2301

To appear in: *Engineering*

Received Date: 26 August 2024
Revised Date: 15 January 2026
Accepted Date: 6 February 2026



Please cite this article as: X. Wang, J. Gu, J. Qiu, G. Liu, X. Cai, J. Zhu, Y. Liu, Z. Dong, J. Zhao, News-Driven Load Forecasting: Generative Agents and Large Language Models for Unstructured Data and Event Analysis, *Engineering* (2026), doi: <https://doi.org/10.1016/j.eng.2026.02.031>

This is a PDF of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability. This version will undergo additional copyediting, typesetting and review before it is published in its final form. As such, this version is no longer the Accepted Manuscript, but it is not yet the definitive Version of Record; we are providing this early version to give early visibility of the article. Please note that Elsevier's sharing policy for the Published Journal Article applies to this version, see: <https://www.elsevier.com/about/policies-and-standards/sharing#4-published-journal-article>. Please also note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

Research

Next Gen-AI Enabled New Type of Power System—Article

News-Driven Load Forecasting: Generative Agents and Large Language Models for Unstructured Data and Event Analysis

Xinlei Wang ^a, Jinjin Gu ^a, Jing Qiu ^{a,*}, Guolong Liu ^{b,c}, Xinlei Cai ^d, Jinzhou Zhu ^d, Yanli Liu ^{e,*}, Zhaoyang Dong ^f, Junhua Zhao ^{b,c,*}

^a School of Electrical and Computer Engineering, The University of Sydney, Sydney, NSW 2006, Australia

^b School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen 518172, China

^c Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS), Shenzhen 518100, China

^d Guangdong Power Grid Dispatch Control Center, Guangzhou 510000, China

^e Key Laboratory of Smart Grid of Ministry of Education, Tianjin University, Tianjin 300072, China

^f Department of Electrical Engineering, City University of Hong Kong, Hong Kong 999077, China

* Corresponding authors.

E-mail addresses: jeremy.qiu@sydney.edu.au (J. Qiu), yanliliu@tju.edu.cn (Y. Liu), zhaojunhua@cuhk.edu.cn (J. Zhao).

ARTICLE INFO

Article history:

Received

Revised

Accepted

Available online

Keywords

Large language model

Load forecasting

Time series forecasting

Power systems

ABSTRACT

This study proposes a novel approach, intelligent text-analytic load forecasting (ITA-LF), to address the short-term load forecasting (STLF) problem by utilizing large language models (LLMs) and generative agents. This emphasizes the challenges faced by traditional forecasting methods in adapting to rapid changes and complex patterns in energy consumption, particularly during unexpected social events. It processes diverse unstructured data (e.g., historical loads, news, calendar dates, and weather), fine-tuning an LLM to enhance prediction accuracy and adaptability. An LLM-based agent with

reasoning capabilities is introduced to select and understand relevant news, demonstrating the model's ability to integrate diverse information for more precise forecasting. Our results surpass all baseline models in predictive accuracy, indicating that LLMs excel in managing the complexities of load forecasting patterns. This innovative approach not only improves forecasting accuracy but also indicates potential shifts in STLF paradigms by integrating unstructured data through advanced artificial intelligence (AI) techniques.

1. Introduction

Short-term load forecasting (STLF) [1] is crucial for the efficient operation, planning, and control of power systems, ensuring the balance between energy demand and supply. However, demand fluctuations are an inherent nature of power systems and cannot be completely captured by historical load data. A variety of factors, including weather conditions, specific dates, and unpredictable social and economic events like industrial shutdowns or unexpected public gatherings, significantly affect load demand. Moreover, the representation of such information can be highly unstructured and may come from diverse sources across different modalities. Existing STLF methods face challenges in processing unstructured data and adapting to random, real-time events, which limits their flexibility and undermines their forecasting accuracy. In this paper, we propose a novel STLF method called intelligent text-analytic load forecasting (ITA-LF) model that leverages the capabilities of pre-trained large language models (LLMs) [2]. It can unify various unstructured information from multiple modalities and data sources in a textual format and intelligently analyzes the information to achieve higher accuracy in load forecasting. Another LLM-based intelligent agent is designed to select load-related news information and provide analysis to assist in load forecasting. Our method represents a significant shift in STLF paradigms.

Current STLF methods have evolved significantly, moving from basic statistical techniques like regression models [3], autoregressive integrated moving average (ARIMA) [4], and exponential smoothing [5] to machine learning strategies such as support vector machine (SVM) [6], random forests [7], and neural network [8]. Deep learning has emerged as a dominant force in the analysis of complex data, leveraging advanced models such as convolutional neural networks (CNNs) [9] for spatial data processing and recurrent neural networks (RNNs) [10], including long short-term memory (LSTM) networks [11], for time-series analysis. Additionally, the integration of attention mechanisms [12,13] has significantly enhanced model performance and interpretability. These methods have made great progress in learning complex nonlinear relationships in load data. The existing methodologies in load forecasting are confronted with several challenges. Firstly, these methods tend to rely on rigid patterns, hindering their ability to effectively process unstructured information, particularly complex unstructured textual data. Secondly, they are constrained by the limited scope of historical data they can analyze. Efficient load forecasting necessitates the integration of extensive knowledge about the energy system, which is not often reflected in historical datasets. Lastly, these approaches are limited to making predictions through non-linear data mapping, unable to discern and understand the underlying causal factors and dynamic shifts that influence these data points. This means that while traditional methods can provide accurate predictions in some cases, they may not be able to adapt to environmental changes or understand emerging patterns and trends. These challenges highlight the limitations of existing approaches. We need new STLF research paradigms and methods to deal more effectively with these complexities.

There are ongoing efforts to integrate textual information into time series forecasting across multiple domains, such as finance [14–16], energy [17–19], entertainment [20], pandemics [21], and tourism [22]. The textual information sources are mainly from social media platforms (e.g., Facebook and Twitter), public reports, and news updates. Most previous studies only counted keyword frequencies [18] or treated events as dummy variable inputs without understanding the nuanced meanings of the text. Some research has tried to advance to include more context but still in a way that simplifies the content into a few analyzable features. For instance, Bai et al. [17] utilized traditional natural language processing (NLP) [23] and random forest methods to forecast energy demand. They selected features such as word frequencies, public sentiments, topic distributions, and word embeddings. However, traditional NLP methods face several challenges in processing complex language tasks associated with large-scale full-article news. Firstly, these methods require extensive manual feature engineering, which is labor-intensive and limited by feature quality. Secondly, they struggle with capturing long-text dependencies and lack a contextual understanding, often only processing information at the sentence or fragment level without grasping the broader document context. Additionally, traditional NLP techniques have scalability issues with large datasets and lack flexibility in multi-task learning, necessitating separate designs for each task. In contrast, LLMs excel in understanding complex textual content and contextual relationships, significantly enhancing prediction accuracy and efficiency while reducing manual intervention through automated feature extraction.

In this work, we propose that language models offer a unified framework for STLF problems, facilitating data assimilation from diverse sources and modalities. At the same time, time series prediction can be considered as a conditional generation problem of text sequences: Given a text containing historical data and unstructured supporting information, the prediction results are then turned into coherent output via text generation techniques. This is also in line with the general paradigm of NLP and general intelligence technology represented by LLMs. LLMs are advanced artificial intelligence (AI) models designed to comprehend and produce text that mimics human language. These models undergo training on vast datasets, equipping them with the ability to navigate a broad spectrum of language-related tasks. They possess an extensive reservoir of common knowledge gleaned from their extensive training material, alongside robust reasoning skills. This allows them to solve problems and make inferences within the given context. More importantly, the capabilities of LLMs can also be extended to other fields, including computer vision [24], audio analysis [25], and time series analysis [26–28]. Several studies have explored the performance of

training LLMs on time series datasets covering various domain knowledge [27,28]. LLMs are also used in some industry applications, such as finance [29], transportation [30,31], and public health [32]. LLMs offer innovative perspectives for enhancing the accuracy of time series predictions, making them well-suited for addressing the aforementioned challenges in STLF problems.

LLMs can contribute to STLF research from three aspects:

Flexibility. LLMs excel in parsing and interpreting unstructured data. By leveraging LLMs, STLF methods can incorporate real-time, unstructured data, including weather conditions, dates, and daily news. Among them, news offers insights into events such as natural disasters, changes in economic conditions, or public health emergencies that greatly impact electricity demand. Traditional time series forecasting models overlook textual information, which is crucial for accurately understanding and predicting fluctuations in power system demands.

Comprehensiveness. Pre-trained LLMs possess the capability to assimilate a broad spectrum of knowledge encompassing social, economic, and technological advancements. This is accomplished through the analysis of extensive training datasets. By harnessing this knowledge, LLMs are adept at integrating data from diverse sources and conducting cross-domain analyses to construct a holistic view, thereby enhancing forecasting accuracy. By introducing rich text information and applying fine-tuning technology, using pre-trained LLMs for load forecasting can significantly improve STLF models so that they can not only identify and understand underlying factors driving load variations but also achieve more accurate prediction results.

Intelligence. The LLMs-based AI agent leverages extensive knowledge learned in pre-training to enhance the selection process of information relevant to load forecasting. These agents are adept at pinpointing potential factors affecting load changes and demonstrate sophisticated reasoning in unseen scenarios. This capability allows them to filter out irrelevant news, reduce noise interference, and manage redundant details that appear in lengthy texts. As a result, these agents are more proficient in assessing the implications of news on energy usage patterns. This AI technology improves the accuracy and efficiency of load forecasting, adeptly addressing the complex challenges associated with interpreting the direct impact of news on energy demand.

In this paper, we propose ITA-LF, a novel STLF framework that integrates diverse unstructured inputs into LLMs, offering a shift in how time series and textual data can be jointly modeled. Our main contributions are summarized as follows:

- (1) We directly incorporate textual data as input to a fine-tuned, pre-trained LLM. Leveraging the LLM's modality adaptability and embedded world knowledge, our method enables context-aware forecasting that extends beyond traditional numerical modeling, and demonstrates reasoning abilities under previously unseen scenarios.
- (2) We construct a comprehensive dataset for LLM finetuning, including half-hourly load, daily weather, calendar indicators (e.g., holidays, weekdays), and region-specific news headlines across Australian sub-regions.
- (3) To address the challenge of long and noisy textual inputs, we design the LLM-based agent that performs content filtering via chain-of-thought prompting, selecting news items most relevant to regional load patterns.
- (4) ITA-LF consistently outperforms strong baselines across multiple metrics. The model shows enhanced robustness to sudden changes and greater responsiveness to external events, confirming the value of LLMs in capturing complex, real-time multimodal dependencies in load forecasting.

2. Literature review

2.1. Short-term load forecasting

STLF is evolving with advancements in data processing technologies. Initial approaches primarily utilized traditional statistical methods. Traditional models like regression [3], ARIMA [4], and exponential smoothing [33] were utilized due to their simplicity, but they are only effective in capturing linear trends and seasonality in load data. With the advent of machine learning and deep learning, more techniques were introduced to identify the non-linear relationships in load data. SVM [6] and random forest model [7] were used when time series are nonstationary with multiple seasonal cycles. CNNs [34–36] were utilized for effective spatial data analysis. Meanwhile, RNNs-based models and their variants, especially LSTM networks, excelled in handling sequential time-series data and capturing temporal dependencies. LSTM was utilized in Ref. [37] for STLF for residential customers by retaining information from previous time steps. RNN-based models struggle to forecast holidays and weekends due to their recursive structure, leading to unstable training and difficulties in capturing long-term dependencies.

Recent research further improves the predictive analytics of STLF models. With the proposed attention mechanism [38] and its excellent feature extraction capability, the forecasting ability in STLF has been enhanced even more through Transformer-based models [12,39]. Transformer-based models [40–46] use an encoder–decoder structure with a self-attention mechanism to extract the global dependencies between inputs and outputs. Compared with RNN, the transformer has a more general structure, a wider range of correlation, and more universal capability. Researchers have proposed different Transformer variants to be applied in many practical scenarios [13,47–49]. Existing methods only take the historical load, calendar information, weather, and holiday factors [50] into consideration, but there is no method integrating textual information sources (e.g., news updates), improving models' adaptability to abrupt changing patterns, and enhancing the interpretability of AI-based models.

2.2. LLMs for time series

Pre-trained language models (PLMs), like bidirectional encoder representations from Transformers (BERT) [51] and

bidirectional and auto-regressive Transformers (BART) [52], based on Transformer architectures, have demonstrated strong performance in NLP tasks. LLMs [2], a subset of PLMs, contain tens or hundreds of billions of parameters. The LLMs training requires extensive practical experience in large-scale data processing and distributed parallel training. In recent years, the LLMs research has seen significant advancements, notably with the introduction of ChatGPT [53], a powerful AI chatbot developed using the LLMs technology, garnering considerable societal interest. The evolution of LLMs continues to influence various fields significantly. During the pre-training process, LLMs learn a lot of general and extensive knowledge and common sense. LLMs also have the capability of in-contextual learning. This implicitly encoded knowledge can be used by subsequent algorithms to construct machine learning systems that can reason and have common sense.

The application of LLMs for time series forecasting can be categorized into general-purpose models and domain-specific models [26]. General-purpose models, such as PromptCast [54] and TimeGPT [28], target various industries and tasks, showcasing superior performance, efficiency, and simplicity in zero-shot inference compared to traditional methods. Time-LLMs [27] focus on general time series forecasting without fine-tuning the backbone model, emphasizing efficiency and resource constraint reduction. Domain-specific models are employed in practical industry applications, like those applied in finance [29], transportation [30,31], and public health [32], demonstrate LLMs' capability in knowledge and reasoning abilities for explainable forecasting within specific industries.

Text-based time series forecasting (such as news-driven predictions) has been explored in domains such as finance [14,15], energy [17,18], entertainment [20], pandemics [21], and tourism [22], using inputs from news, social media, and public reports. However, prior methods often relied on keyword frequencies or dummy variables [18], lacking deep semantic understanding. Some studies applied traditional NLP techniques [17], extracting features like sentiment, word frequency, or topic distribution. However, these approaches require extensive manual engineering, struggle with long-text dependencies, and lack scalability. In contrast, LLMs offer unified, context-aware representations that reduce manual intervention and enable richer reasoning over unstructured text.

2.3. LLMs as generative agents

LLMs can be used to build agents to automate specific tasks with human-like reasoning ability through “chain of thought” prompting, enhancing complex reasoning [55]. Hence, the LLMs-based agent can solve a given task that reflects the task feedback signals in a series of text and then retains its reflected text in a memory buffer to induce better decisions [56]. Cai et al. [57] proposed an LLMs as tool makers (LATM) framework, where LLMs create their reusable tools for problem-solving. They interleave reasoning traces and task-specific actions, aiding in task completion and environmental interaction [58]. Multi-agents based on LLMs were also studied to improve the intelligence of AI tools further. Integrating LLMs with multiple AI agents can enable believable simulations of human behavior [59]. Multi-agent based on LLMs can also debate their responses and reasoning processes in multiple rounds to arrive at final actions [60]. Incorporating LLMs-based agents can advance the capabilities of LLMs and pave the way for further breakthroughs in language understanding and problem-solving.

3. Method

3.1. Overview

Language models are trained on a collection of sequences $\mathcal{U} = \{U_1, U_2, \dots, U_i\}_{i=1}^N$, where $U_i = (u_1, u_2, \dots, u_{n_i})$ and each u_i is a token representing word or fragment that constitute words. These tokens belong to a vocabulary $\mathcal{V}$ and are obtained by a pre-defined corresponding tokenizer. LLMs typically encode an auto-regressive distribution, in which the probability of each token is only dependent on the previous tokens in the sequence as

$$p_{\theta}(U_i) = \prod_{j=1}^{n_i} p_{\theta}(u_j | u_{0:j-1}), \quad (1)$$

where θ is the model parameters learned by maximizing the probability of the entire dataset $p_{\theta}(\mathcal{U}) = \prod_{i=1}^N p_{\theta}(U_i)$, and $u_{0:j-1}$ is the sequence from 0 to $(j-1)$ th token. We utilize a trained language model through a sampling method. Sampling from a language model typically starts with a prompt $u_{0:k}$, and proceeds sequentially using $p_{\theta}(u_j | u_{0:j-1})$. Language models predict sequences token-by-token, where data is a collection of sequences $\{V_i = (v_1, v_2, \dots, v_{n_i})\}$ with numerical values v_j . Forecasting involves predicting a future sequence $v_{j:k}$ from a prompt $v_{0:j-1}$. While language models predict word tokens, for time series they predict numbers. Given their proficiency in representing complex probability distributions over sequences, language models are well-suited for time series prediction when predicting numerical tokens.

LLMs, with their large amount of parameters and extensive training datasets, are adept in natural language's conditional probabilities, auto-regressively generating text as described in Eq. (1). This study utilizes LLMs through a pre-training and fine-tuning approach. Pre-training involves unsupervised learning on a vast internet-sourced corpus, absorbing a wide range of foundational knowledge and inferential skills relevant to human activities, including aspects of electricity generation and demand. To tailor LLMs for STLF, we must fine-tune them with specific load data, enabling the application of their broad capabilities to the precise forecasting task.

The fine-tuning process is supervised training on a small amount of load data based on the PLMs. This training uses the same loss function as in pre-training but with data specifically designed for our forecasting needs. The data for supervised fine-tuning

consists of prompts and response, both textual. The prompts contain the historical time series and other relevant information for predictions. The responses hold the forecasted time series in text form. The fine-tuned model predicts the response from the prompts, effectively forecasting the load time series.

Data construction and language model fine-tuning are our approach's two most important parts. The collection and screening of supplementary information, along with the construction of our STLTF dataset, are detailed in Section 3.2. The network architecture is described in Section 3.4, while the fine-tuning method is detailed in Section 3.5

3.2. Data construction

The data used for fine-tuning is composed of three main parts: load, supporting information, and news articles.

Regional load demand. Historical load data in different geographical locations is critical for understanding typical consumption patterns and training forecast models. We integrate the data into the instruction input of the prompt, which is used by the model to understand and predict the regional load demand before the prediction date. We convert the numbers into numerical strings for the language model to read and predict.

Supporting information. Our approach incorporates historical load data and weather forecasts, humidity, pressure, holidays, and special dates into a natural language narrative for flexible integration with language models. For example, as shown in Fig. 1, we use the text “weather of the prediction date: the minimum temperature is 292.01 K; the maximum temperature is 298.07 K; the humidity is 94%; the pressure is 1009 hPa” to express weather conditions. This enriches the input, covering various factors influencing load demand. The units are consistent throughout our data processing.

News articles. We utilize real-time news articles to enhance load forecasting, leveraging the textual data to capture information on emergent events like weather shifts and health crises that numerical data might miss, thus improving prediction accuracy. News offers context on lifestyle, societal, and technological developments, aiding our STLTF model in anticipating demand fluctuations due to irregular events such as public gatherings or unexpected shutdowns.

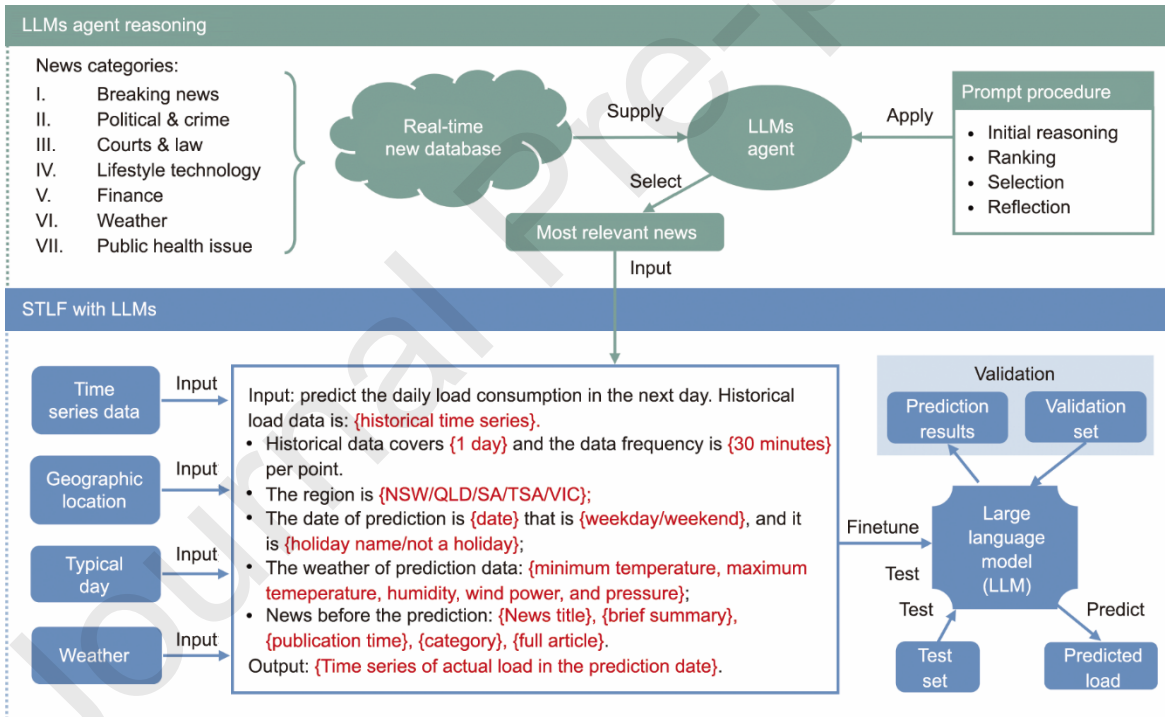


Fig. 1. Framework for ITA-LF model. The process begins with inputting various data while the LLMs-based agent processes and selects relevant real-time news and constantly updates the real-time news database. NSW: New South Wales; QLD: Queensland; SA: South Australia; TSA: Tasmania; VIC: Victoria.

3.3. News selection using LLMs-based agents

Directly utilizing raw news for STLTF presents challenges: much of the news is irrelevant to residential electricity demand, adding noise and potentially decreasing accuracy. Additionally, overly lengthy news articles can overwhelm language models, which have difficulty processing long inputs containing redundant details. Relevant news also typically doesn't specify its direct effect on energy use, requiring models to deduce its impact on demand, a sophisticated task for language models applied directly to load forecasting.

In response, we propose employing an LLMs-based agent, a sophisticated AI system powered by LLMs, capable of performing a wide array of tasks including conversations, reasoning, and exhibiting semi-autonomous behavior. By employing carefully

crafted prompts that define roles, instructions, permissions, and context, an LLMs-based agent can interpret human commands and generate text to execute complex tasks. Specifically, an LLMs-based agent endowed with inferencing prowess is utilized within our framework to efficiently sift through, categorize, and interpret news texts according to their projected influence on future load demands. This approach enables the condensation of vast news datasets into a refined collection of pertinent and well-formatted articles aiding in the prediction of load demand.

An LLMs-based agent is guided using prompts and the chain-of-thought approach, simplifying complex problems into manageable parts for reasoned solutions. Our three-phase method filters news data and conducts inference, as depicted in Fig. 2. This work operates through a sequence of three steps, each initiated by a prompt.

Prompt 1 develops an understanding of regional load influencers, sorting them by impact (positive/negative) and duration (short/long-term), considering economic, policy, seasonal, and technological factors.

Prompt 2 directs the LLMs-based agent to filter and categorize news based on impact analysis from Prompt 1, focusing on relevance to energy demand and classifying the impact (e.g., long-term, short-term, and immediate).

Prompt 3 defines the agent's output format, instructing the agent to format the curated news in JavaScript object notation (JSON) with details like the brief summary, affected region, reporting time, and selection rationale.

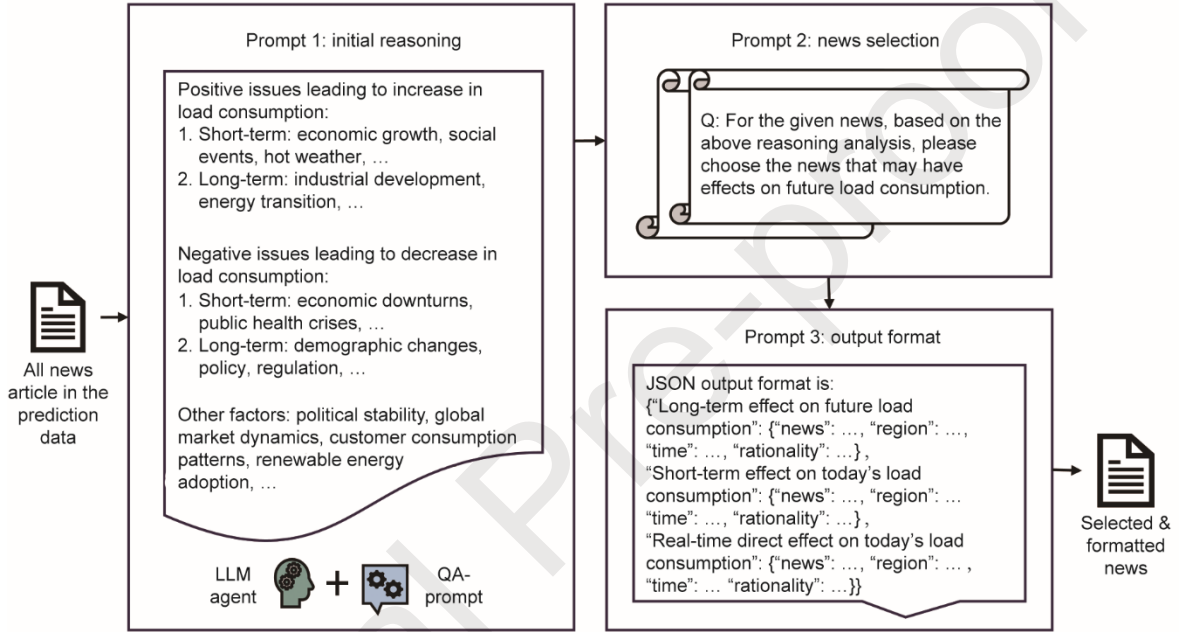


Fig. 2. Workflow of an LLMs-based agent for news analysis and selection in ITA-LF: The workflow starts from initial reasoning to determine positive and negative issues affecting load demand, followed by news selection based on these insights and structuring the selected news in a standard output format. QA: question and answer.

3.4. LLMs Transformer architecture

This work primarily utilizes the Llama2 model [61], selected for its ease of use, generality, and relatively good performance, among other potent LLMs capable of similar functionalities. While proprietary models like GPT-3.5 or GPT-4 may offer higher raw performance, their closed nature makes integration and adaptation challenging. We selected LLaMA 2 for its favorable balance between performance and efficiency, as well as its open-weight availability, which enables reproducible fine-tuning in controlled environments. Llama2 has demonstrated good performance on reasoning-intensive tasks when combined with lightweight fine-tuning techniques such as low-rank adaptation (LoRA), as supported by recent benchmarks. In this part, we briefly review the architecture of the transformer network used in the Llama2 language model. This Transformer network follows the standard transformer decoder design. Fig. 3 illustrates the main computational process of the network, which we will proceed to introduce in sequence.

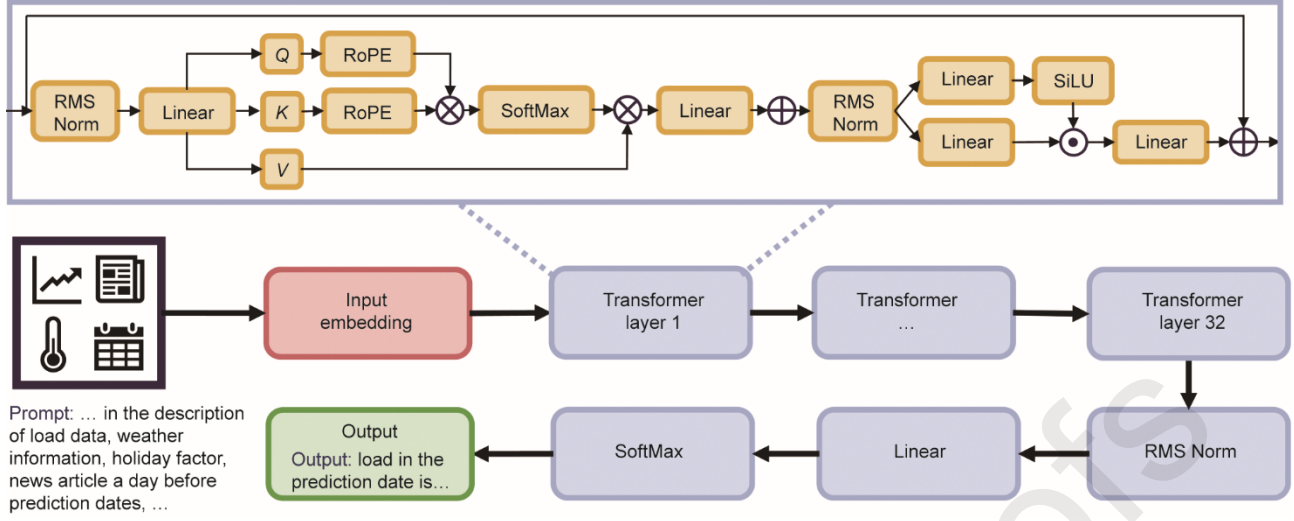


Fig. 3. Framework of an LLMs Transformer architecture. The architecture comprises an input embedding layer, followed by 32 Transformer blocks, each containing a self-attention mechanism with rotary position encoding (RoPE) and a feedforward neural network. Root mean square normalization (RMS Norm) is applied before each sub-layer. Q : query; K : key; V : value; SiLU: sigmoid linear unit.

3.4.1. Input embedding

The input of the Transformer is a sentence or a paragraph, which is usually represented as a sequence of words or characters ($w_1, w_2, \dots, w_{n_w}$). w_{n_w} denotes the n_w th word. The text needs to be tokenized and divided into words or characters to form a token sequence ($u_1, u_2, \dots, u_{n_u}$). u_{n_u} denotes the n_u th token. The dictionary $\mathcal{V}$ provides each token u_i to a word vector $x_i \in \mathbb{R}^d$, where d is the dimension of the word embedding.

3.4.2. Transformer layer

The Llama2 model relies on stacked transformer layers to process these tokens. Each transformer layer module consists of normalization layers, a self-attention layer, and a feed-forward neural network.

Self-attention layer. The self-attention layer enables it to prioritize different segments of the input sequence during processing. This is achieved by allowing each element of the sequence to “interact” with others, thereby dynamically refining its representation. Given an input sequence $X = \{x_1, x_2, \dots, x_n\}$ with n elements, where each element is a d -dimensional vector (potentially the output from an embedding layer with shape (n, d)), the model generates three distinct representations for each element through linear transformations: query (Q), key (K), and value (V), all of the same dimension e . This process involves three weight matrices W^Q , W^K , and W^V to compute Q , K , and V :

$$Q = XW^Q, K = XW^K, V = XW^V \quad (2)$$

where W^Q , W^K , and W^V are parameters learned by the model. The self-attention layer determines how much each element in the sequence affects other elements by calculating the dot product between the query and all keys and then applying the softmax function to obtain the attention weights. These attention weights represent how much attention each element should give to other elements when generating output:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \quad (3)$$

The attention mechanism calculates the attention weights matrix $A = \frac{QK^T}{\sqrt{d}}$, where $A \in \mathbb{R}^{n \times n}$. The softmax function is applied to each row of A , scaling with $\sqrt{d}$ to moderate the dot product’s magnitude and prevent gradient vanishing during softmax. The softmax ensures that the sum of weights for each output element equals 1. The model employs multi-head attention by dividing Q , K , and V into multiple “heads,” processing them in parallel, and then concatenating the results through another linear transformation for diverse subspace representation.

Feed-forward network (FFN). The FFN module is a key component used to process the output of the self-attention layer further. This layer is mainly responsible for nonlinear transformation and dimension transformation based on the information captured by the self-attention layer. In our network, three different linear transformation layers are included as f_1 , f_2 , and f_3 . Each linear transformation layer can be expressed as $f_k = XW^{f_k} + b^{f_k}$, W^{f_k} is the weight matrix, b^{f_k} is the bias term, and X is the input, usually the output of the previous layer. Then, the FFN is performed as:

$$\text{FFN}(X) = f_3(\text{SiLU}(f_1(X)) \odot f_2(X)) \quad (4)$$

where $\text{SiLU}(\cdot)$ is the activation function, defined as sigmoid-weighted linear unit [62]:

$$\text{SiLU}(x) = x \cdot \text{Sigmoid}(x) = x / (1 + e^{-x}) \quad (5)$$

which applied to the input element-wise.

Root mean square normalization (RMS Norm). RMS Norm [63] is a normalization technique used in the Transformer model. As a variant of layer normalization, it is designed to solve problems when training deep networks, such as gradient disappearance or explosion. The traditional normalization layer has a re-centering and re-scaling invariance property, which is sensitive to noise on both inputs and weights. It keeps the output representations intact when both inputs and weights are randomly scaled. RMS Norm focuses on re-scaling invariance and regularizes the summed inputs simply according to the root mean square (RMS) statistic. The layer normalization of the standard Transformer decoder structure is given by

$$y = \frac{x - E[x]}{\sqrt{\text{Var}[x] + \epsilon}} \cdot \gamma + \beta \quad (6)$$

where ϵ is a constant for numerical stability, and γ and β are learnable affine parameters for scaling and shifting the normalized output. The expected value $E[x]$ and the variance $\text{Var}[x]$ are

$$E[x] = \frac{1}{N} \sum_{i=1}^N x_i, \text{Var}[x] = \frac{1}{N} \sum_{i=1}^N (x_i - E[x])^2 \quad (7)$$

RMSNorm simplifies the layer normalization by removing the mean statistic at the cost of sacrificing the invariance that mean normalization affords. RMSNorm removes the parameter β and the calculation of mean, thus:

$$y = \frac{x}{\sqrt{\text{Mean}(x^2) + \epsilon}} \cdot \gamma, \text{Mean}(x^2) = \frac{1}{N} \sum_{i=1}^N x_i^2 \quad (8)$$

Rotary position encoding (RoPE). RoPE encodes sequence position information. RoPE preserves the order of words in the sequence in a relative position by directly integrating position information into the representation of each token. The key advantage is that it maintains rotational equivariance of self-attention operations, allowing the model to better understand the relative positional relationships between words within a sequence.

Suppose an input sequence of length N : $X = \{x_i\}_{i=1}^N$, $x_i \in \mathbb{R}^d$ is a word vector in the sequence. We want to encode the position of this vector in the sequence X . When performing self-attention calculation, we calculate key and query according to Eq. (2). When considering relative position, we hope to add position information to q and k through $\bar{q}_m = f_q(x_m, m)$ and $\bar{k}_n = f_k(x_n, n)$, where x_m represents the m th vector in X , $\bar{q}_m$ denotes the query vector for x_m , and $\bar{k}_n$ is the key vector for x_n . Following this function, $\bar{q}_m$ and $\bar{k}_n$ acquire the absolute positional information of positions m and n . Subsequently, the self-attention operation computes the inner product between $\bar{q}_m$ and $\bar{k}_n$, aiming to incorporate the relative positional information of m and n , as follows:

$$\langle f_q(x_m, m), f_k(x_n, n) \rangle = g(x_m, x_n, m - n) \quad (9)$$

where $\langle \cdot, \cdot \rangle$ is the inner product. For g , we only need its expression to contain $x_m, x_n, m - n$. Su et al. [64] gave an expression of f that satisfies this condition, as:

$$f_q(x_m, m) = (W^Q x_m) e^{im\theta}, f_k(x_n, n) = (W^K x_n) e^{in\theta} \quad (10)$$

$$g(x_m, x_n, m - n) = \text{Re}[(W^Q x_m)(W^K x_n)^* e^{i(m-n)\theta}]$$

where $\text{Re}[\cdot]$ represents the real part of a complex number, and $(W^K x_n)^*$ represents the complex conjugate of $(W^K x_n)$. θ is the rotation frequency parameter to encode positional information. In practical calculations, for the embedding of multi-dimensional feature vectors $q_m = W^Q x_m$. The $f_q(x_m, m)$ can be calculated as:

$$f_q(x_m, m) = \mathbf{R}_{\theta_m, m}^d W^Q x_m \quad (11)$$

where $\mathbf{R}_{\theta_m, m}^d$ is the rotary matrix with predefined parameters ($\theta = 10000^{-2(i-1)/d}, i \in [1, 2, \dots, d/2]$). Eq. (12) represents detailed expression of the block-diagonal matrix $\mathbf{R}_{\theta_m, m}^d$ allowing to compute the rotated RoPE.

$$\mathbf{R}_{\theta_m, m}^d = \begin{pmatrix} \cos(m\theta_1) & -\sin(m\theta_1) & 0 & 0 & \cdots & 0 & 0 \\ \sin(m\theta_1) & \cos(m\theta_1) & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cos(m\theta_2) & -\sin(m\theta_2) & \cdots & 0 & 0 \\ 0 & 0 & \sin(m\theta_2) & \cos(m\theta_2) & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \cos(m\theta_{d/2}) & -\sin(m\theta_{d/2}) \\ 0 & 0 & 0 & 0 & \cdots & \sin(m\theta_{d/2}) & \cos(m\theta_{d/2}) \end{pmatrix} \quad (12)$$

3.5. Fine-tuning LLMs

We perform fine-tuning on pre-trained LLMs using a parameter-efficient method called LoRA. It adapts the Llama2 backbone to our domain-specific forecasting task. Full fine-tuning of LLMs is computationally expensive and memory-intensive, often requiring extensive graphics processing unit (GPU) resources and risking instability or catastrophic forgetting. LoRA mitigates this by freezing the original model weights and introducing a small number of trainable low-rank matrices within attention and feedforward layers. This greatly reduces the number of trainable parameters and allows efficient domain adaptation with limited resources. In our case, LoRA enables the model to internalize energy-related reasoning patterns, such as interpreting load-affecting events or understanding temporal contexts, without altering the core language model.

The core concept of LoRA is to conduct LoRA on the self-attention module and the weight matrix of feed-forward neural network layers in the Transformer. This involves adding a low-rank matrix to the weight matrix obtained by multiplying smaller, trainable parameter matrices. Thus, the original weight matrix remains unchanged during training, while only the newly added low-rank matrix is updated, significantly reducing the number of trainable parameters. LoRA typically applies to the query W^Q and key W^K matrices of the self-attention layer, as well as the weight matrix of the feed-forward neural network layers W^{f_i} .

Assume that the original weight matrix is $W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, where d_{out} is the output dimension and d_{in} is the input dimension. The two low-rank matrices introduced by LoRA are $A \in \mathbb{R}^{d_{\text{out}} \times r}$ and $B \in \mathbb{R}^{r \times d_{\text{in}}}$, where r is the rank of a low-rank matrix, which is usually much smaller than d and k . The original weight matrix W is updated to $W + AB$, where the calculation of AB represents a low-rank approximation. For a given input $x \in \mathbb{R}^{d_{\text{in}}}$, the updated weight matrix ($W + AB$) is used to calculate the output $y = (W + AB)x$. In our calculations, r can usually be set to 8. By adding two smaller matrices, A and B , as trainable parameters, LoRA greatly reduces the number of parameters required for fine-tuning, making it more efficient when dealing with large models. The original LLMs involve 7 billion parameters, but through LoRA technology, we only need to fine-tune no more than 20 million parameters to achieve good results.

4. Experimental details

4.1. Dataset

For comprehensive STLF analysis, our dataset integrates reliable open-source repositories, comprising load demand, real-time news, and weather data from five Australian states (Queensland, New South Wales, Tasmania, Victoria, and South Australia) covering January 1, 2019, to December 31, 2021. Primary data sources include:

Half-hourly regional load data. We utilize half-hourly demand data sourced from the Australian Energy Market Operator (AEMO) website. This dataset provides comprehensive historical electricity usage by region, aiding in understanding consumption patterns and trends.

Real-time news articles. We compile a real-time news database to capture current events potentially influencing electricity demand. This database covers various topics and regions, offering timely insights into factors impacting energy usage.

Weather information. Data from OpenWeatherMap provides detailed meteorological information, including daily temperature (maximum and minimum), atmospheric pressure, wind speed, and humidity.

To enhance the prediction of electricity demand in Australia, we collected 380 560 articles encompassing a wide variety of topics relevant to this subject. Through web crawling and intelligent parsing, we extracted crucial information from each article, such as titles, summaries, categories, uniform resource locators (URLs), publication dates, and the complete article content. Regular expressions were employed to extract these essential data points accurately. The structured data was subsequently formatted into JSON files to ensure compatibility and ease of integration into model training processes.

4.2. Implementation

When fine-tuning, we employ the LoRA [65] method to fine-tune the Llama2 LLM [66]. The rank in LoRA is set to either 8 or 16, depending on the required token length to be processed. The alpha value in LoRA is set to 16, and the learning rate is set at 0.0001. When formatting numeric tokens, we uniformly maintain three significant figures. Retaining more significant figures offers minimal benefits and introduces an excessive number of tokens. The fine-tuning is conducted on a single NVIDIA A100 40G GPU, with the LLM undergoing several hundred to 1000 training iterations on our curated data, requiring several hours. For the designed agent, we use GPT4 [53] for best results. It is also feasible to replace other language models.

In our experiments, the training dataset comprises half-hourly load data, weather information, calendar dates, and news from the five states of Australia for the years 2019 and 2020. The prediction dataset includes half-hourly load data from the same states for the year 2021, measured in kilowatts. For our method and all other baseline prediction models, we use the historical data from the previous day to predict the half-hourly load demand for the next day.

4.3. Baselines

A diverse set of baselines, which represent the latest advancements in load forecasting, are selected. Our selection includes Transformer-based models, such as Transformer-based model: Informer [40], Autoformer [41], frequency enhanced decomposed Transformer (FEDformer) [42], Pyraformer [43], patch time series Transformer (PatchTST) [44], iTransformer [45], feature-wise linear modulation (FiLM) [46]. Additionally, we included the CNN-based Timesnet [36] and MLP-based dlinear [67]. We incorporated the LLM-based GPT for time series (GPT4TS) [68], leveraging GPT2’s generative capabilities. For Informer, Autoformer, FEDformer, Pyraformer, PatchTST, iTransformer, FiLM, and Timesnet, we use the code from the Time-Series-Library.

While our LLM fine-tuning approach maintains the original scale of the numbers to preserve the physical meaning of the time series data, we normalize the variables to the range $[0, 1]$ for training the baseline methods. We represent non-numeric variables using dummy variables for these baseline methods. To ensure optimal performance, we adhere to the official architecture settings for these methods with learning rates of 0.001, 0.0005, 0.0001, and 0.00005. Each configuration is tested three times with different initializations, and the best-performing setup is selected from all trials.

5. Results analysis

In the experiment, we conducted experiments using the previous day's load data and relevant information for next-day half-hourly load forecasting. We designed four distinct cases to evaluate our methodology's impact on STLF:

Case 1 (only numeric prompt) combines all-time series, including historical load and weather conditions, along with categorical data like region and holiday indicators, as input vectors for our model: $[l_{t_1}, \dots, W_1, \dots, d_{\text{region}}, d_{t, \text{weekday}}, d_{t, \text{holiday}}, \dots]$, where l_{t_1} represents the day-ahead actual load at time t_1 , and W_n denotes weather data. $d_{t, \text{weekday}}$, $d_{t, \text{holiday}}$ are dummy variables of the prediction date, and d_{region} denotes the prediction region.

Case 2 (textual prompt without news) transforms all relevant data into a descriptive text input, simulating a natural language understanding task for the LLMs to forecast load demand:

The historical load is...; Based on the historical load data, please predict the load demand in the next day. The region for prediction is ...; The start date of historical data was on ... that is, a Weekday/weekend, and it is/ is not a public holiday. The data frequency is 30 minutes per point. Historical data covers 1 day. The date of prediction is on ... that is a Weekday/weekend, and it is/ is not a public holiday. Weather of the start date: the minimum temperature is ...; the maximum temperature is ...; the humidity is ...; the pressure is ...;

Case 3 (textual prompt with non-filtered news) extends Case 2 by incorporating text paragraphs of all prior information plus news articles available before the prediction date into the model's input.

Case 4 (textual prompt with filtered news) enhances the input further by employing LLMs-based agents to filter and integrate the most relevant news information into the model, aiming for more accurate predictions.

5.1. Overall performance

We choose four metrics to compare the prediction performances of different models. Mean squared error (MSE) is used to verify outliers, focusing on larger prediction errors. RMS error (RMSE) measures error magnitude by averaging squared errors and taking the square root, aligning the error scale with the original data. Mean absolute error (MAE) is more balanced for different magnitude errors. Mean absolute percentage error (MAPE) expresses the error as a percentage of the actual values, that is, on average, how far off the predictions are from the actual values in percentage terms. These metrics are calculated by:

$$\begin{cases} \text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \\ \text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \\ \text{MAE} = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \\ \text{MAPE} = \frac{100}{N} \sum_{i=1}^N \left| \frac{y_i - \hat{y}_i}{y_i} \right| \end{cases} \quad (13)$$

where y_i represents the unnormalized true value of load, and $\hat{y}_i$ denotes the unnormalized result of the prediction model. N denotes the number of prediction timestamps.

Table 1 presents a comparative analysis of prediction results between the baseline models and our proposed method across four distinct metrics. Our approach, ITA-LF, consistently outperforms all baseline methodologies across various metrics. ITA-LF demonstrates superior performance with the lowest MSE (0.07862), RMSE (280.39), MAE (180.96), and MAPE (5.15%). This indicates that ITA-LF is a highly accurate and reliable forecasting model with minimal errors and high precision. CNN-based, MLP-based, and Transformer-based models show varied performance levels, with GPT4TS and Pyraformer having comparatively better prediction results among the baselines, yet still significantly worse than ITA-LF. While these models are effective in hourly predictions, they are surpassed by the advanced ITA-LF configurations in half-hour frequency, especially when complex data integration and sophisticated forecasting techniques are applied. Unlike traditional methods, our model achieves good results without normalizing cross-domain data, underscoring its advantages: simplified data processing, enhanced adaptability, and flexibility. This independence from data normalization broadens the model's application potential across various fields.

Table 1
Comparison of baselines for load forecasting on different metrics.

Model	MAE	MSE ($\times 10^{-3}$)	RMSE	MAPE (%)
Ours	180.96	78.62	280.39	5.15
Autoformer	480.51	396.94	630.03	18.82
Informer	498.35	370.76	608.89	20.33
Dlinear	255.61	160.61	400.76	7.29
iTransformer	244.26	142.82	377.91	6.95

	257.85	153.90	398.98	7.36
FiLM	238.15	134.07	366.15	6.86
TimesNet	364.49	246.95	496.94	10.90
Pyraformer	245.68	151.54	389.28	6.87
PatchTST	261.86	159.23	399.03	7.92
FEDformer	254.69	160.05	400.07	7.06
GPT4TS				

Baseline without normalization. A lower value indicates better performance. Bold: the best and the second best.

5.2. Ablation study

The contribution of different features to a model's performance is assessed by systematically removing or altering them. Table 2 shows the comparison of prediction results with different prompt designs from Case 1 to Case 4. The results highlight the importance of textual prompts and news selection on prediction accuracy:

Importance of textual prompt. The use of an LLM-based agent in Case 4 significantly enhances performance, suggesting that integrating news data with an LLM-based agent can improve prediction accuracy. The progressive improvement observed from Case 1 to Case 2 and then to Case 4 in the ITA-LF model indicates that incorporating diverse data inputs (such as text and news, with or without agents) consistently boosts forecasting accuracy. The transition from Case 1 to Case 2 demonstrates a slight improvement in RMSE and MSE metrics when all information is organized in a textual format, indicating that text provides additional valuable information to capture patterns not evident in purely numeric data. Conversely, the shift from Case 2 to Case 3 results in poorer prediction accuracy due to the inclusion of too much irrelevant news content, which introduces noise and affects the outcome. This highlights the importance of careful news selection. The results of Case 4 reveal that the most relevant news articles, after selection, contain valuable information that impacts load demand, such as events or situations not captured by historical numeric data alone.

Effectiveness of news filtering and selection. The most significant leap in prediction accuracy is observed between Case 3 and Case 4. The only difference between these two cases is applying an LLMs-based agent as a filter to the news content in Case 4, which suggests that filtering allows the model to focus on the most pertinent information, reducing noise and potentially misleading data that could degrade performance. It is noteworthy that the accuracy improvements are especially pronounced at the peaks and troughs of the load demand curve. These points are often the most challenging to predict due to their association with atypical events. The enhanced performance of Case 4 in these periods underscores the value of well-curated data in capturing extreme scenarios.

Table 2
Performance comparison of various prompts designs.

Prompt design	RMSE	MSE ($\times 10^{-3}$)	MAE	MAPE (%)
Case 1: only numeric prompt	337.10	113.64	204.89	5.27
Case 2: textual prompt without news	336.41	113.17	206.08	5.29
Case 3: textual prompt with non-filtered news	407.86	166.35	250.75	6.84
Case 4: textual prompt with filtered news	280.39	78.62	180.96	5.15

Bold value indicates the best.

5.3. Information from the news driving the performance

Our ablation results highlight two key factors that drive the effectiveness of news integration. First, by comparing LLM-selected news with raw, unfiltered news, we observe that incorporating causally meaningful events, accompanied by a brief explanation of their potential impact (e.g., weather disruptions, public gatherings, or policy changes), provides valuable signals for forecasting non-repetitive load fluctuations. Such information enables the model to reason about likely deviations from historical patterns based on contextual understanding and increase prediction accuracy. Second, a comparison between region-specific news and generic news shows that geographic and temporal alignment plays a critical role. News that is directly relevant to the target region and forecast period yields significantly better predictive performance. Thus, our LLM-based agent is designed to capture both dimensions by jointly reasoning over event content and its associated metadata (e.g., publication time and location).

Fig. 4 traces how news content, such as coronavirus disease 2019 (COVID-19) lockdowns, anticipated power outages, and behavioral shifts in residential usage, contribute to the model's ability to adjust forecast peaks and valleys. Results demonstrate the practical effect of incorporating effective news content into forecasting models. The "With News" predictions are closer to the actual values than the "No News" predictions, particularly at critical timestamps where abrupt events significantly influence load demand:

The effect of incorporating news about the COVID-19 lockdown on predictions. Predictions with news better match the actual value curve during the sharp decline, suggesting that news about lockdowns is crucial for accurate forecasting.

The effect of incorporating news about the residential behavior. Predictions with news closely match the actual value at the peak load, implying that it can anticipate the increased demand during peak hours.

The effect of incorporating news about the anticipated power outages. Predictions with news closely mirror the actual consumption pattern, including the drop in load, suggesting it can effectively incorporate news of such events into its predictions.

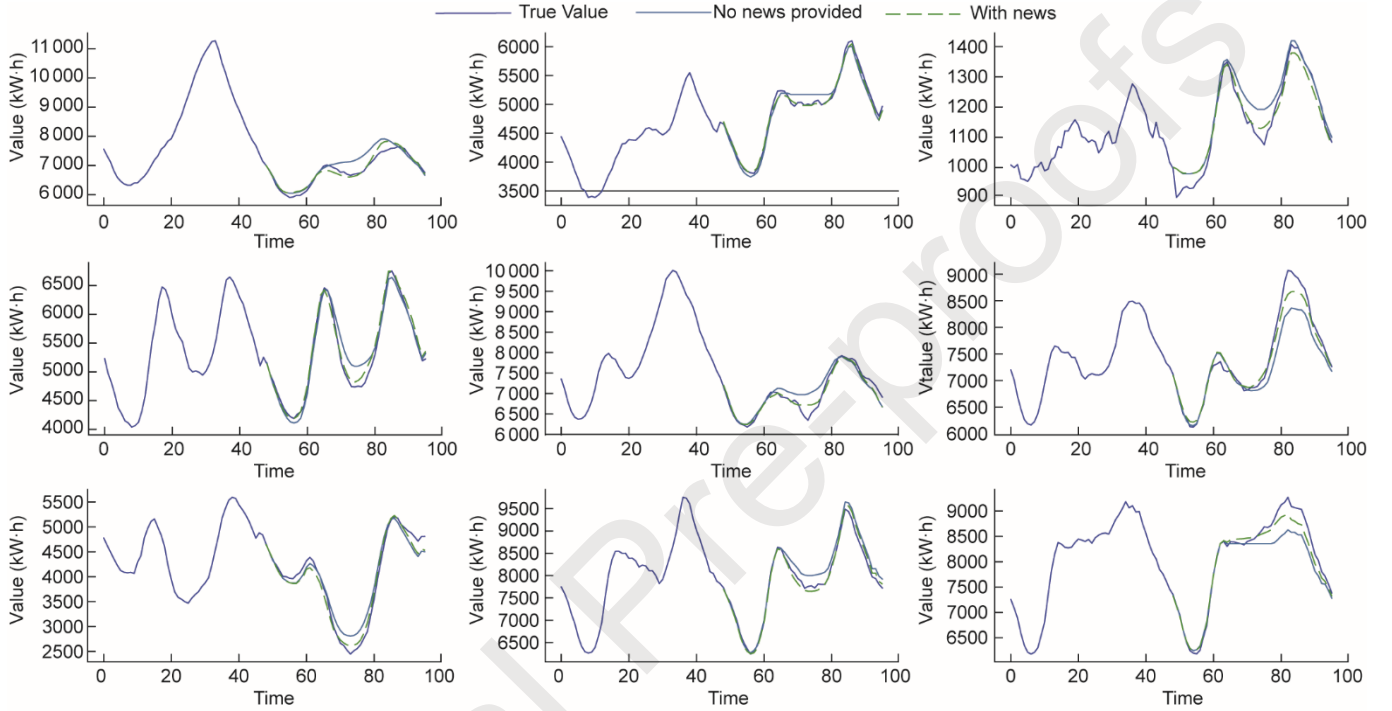


Fig. 4. Day-ahead load forecasting cases with and without news contents. The horizontal axis represents the time index (half hour). The predictions are plotted against the actual load demands in solid blue. Predicted values with news are in dashed green, and predicted values without news are in solid light blue.

5.4. Predictions in different regions and dates

Fig. 5 shows that our model consistently outperforms all baseline models in aligning with actual values across all Australian regions and different calendar dates. Among the various regions, ITA-LF particularly excels in capturing peak and valley demands, while baseline models exhibit varied accuracies and tend to underestimate such loads. The predictions in Tasmania deserve special attention due to its smaller and more volatile load demand, influenced by specific local conditions such as its reliance on hydroelectric power. These factors contribute to the increased complexity and difficulty of making accurate predictions in this region. Nonetheless, our model maintains closer adherence to actual values, highlighting its robustness in dealing with complex environments.

Fig. 5 also compares day-ahead load forecasting accuracy for various models across different day types, including weekdays and weekends. The ITA-LF model effectively captures the typical consumption patterns of weekends and weekdays. In contrast, baseline models, like LSTM, often struggle with volatile weekend load patterns due to changes in human activity. These results suggest that ITA-LF is more reliable for forecasting load in different regions and dates due to its advanced feature extraction, ability to capture complex temporal dependencies, and integration of more sophisticated data sources.

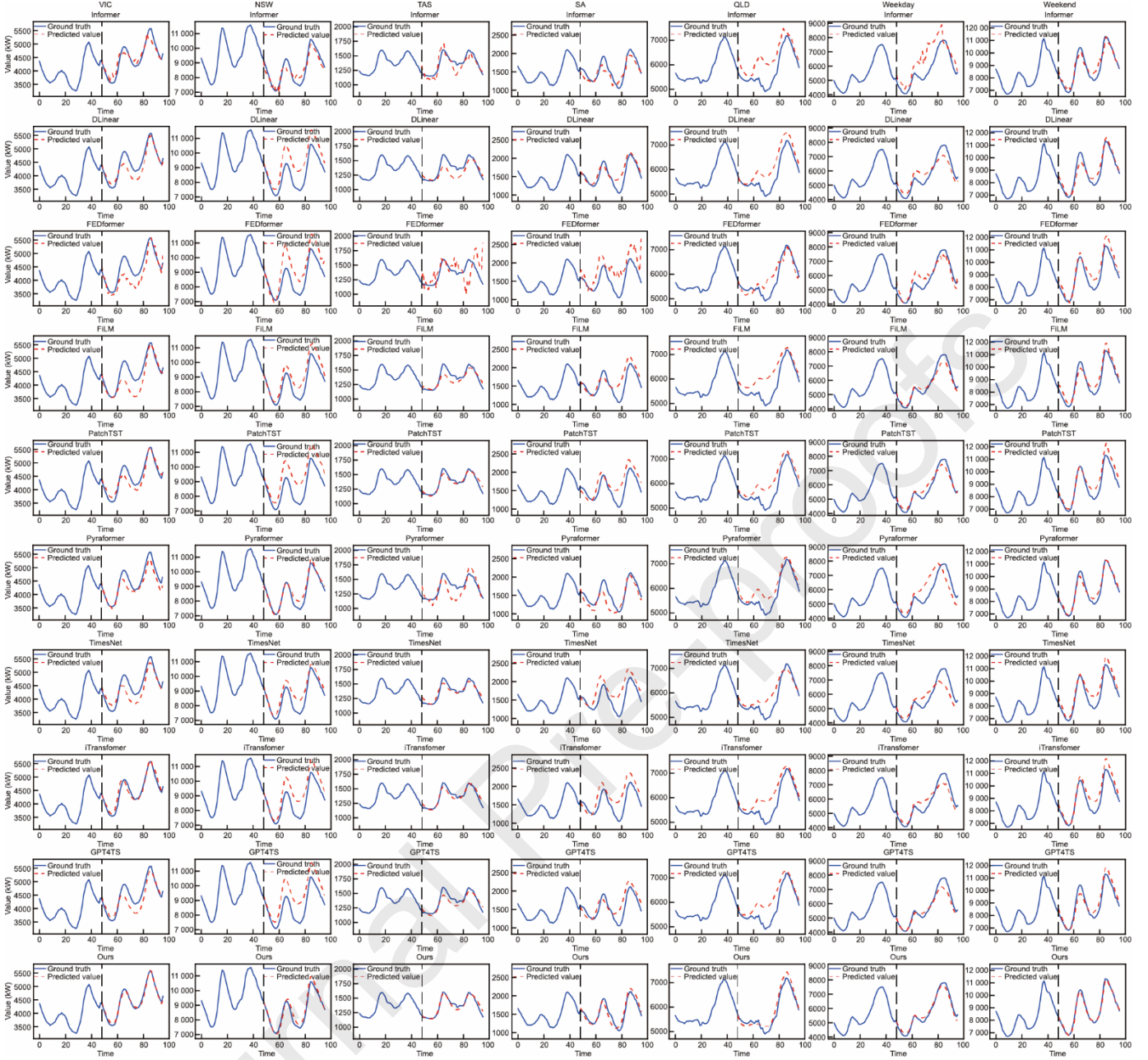


Fig. 5. Day-ahead load forecasting cases of different regions and calendar dates in Australia. The horizontal axis represents the time index (half hour), while the vertical axis represents the load demand (kW).

6. Conclusions

This paper introduces a novel STLF method—ITA-LF—utilizing advanced LLMs to enhance the prediction of volatile power system demand by integrating diverse, unstructured data sources, including news on social and economic events for improved accuracy. Based on LLMs, unstructured data, which come from diverse sources across different modalities, are fully explored to achieve higher accuracy in load forecasting. For instance, our method adeptly accounts for the volatility spurred by unforeseen socio-economic developments. Our LLM-enabled agent is systematically designed to curate pertinent load-related news, bolstering forecasting efficacy. For example, news about unpredictable social and economic events is considered in our approach. The LLMs-based agent is also designed to select load-related news information and assist in load forecasting.

Our results indicate that using diverse data sources and LLMs significantly improves prediction accuracy and reliability. Specifically, predictions that incorporate news articles align more closely with real outcomes, especially in response to urgent events such as traffic accidents, wildfires, pandemics, or lockdowns. This highlights the value of including all relevant real-time events in models to better reason and anticipate sudden changes in load demand, underscoring the role of context in enhancing predictive model performance. Moreover, integrating filtered news content into forecasting models with the LLM-based agent significantly boosts accuracy by filtering for relevance. Overall, our model demonstrates superior adaptability and alignment with real-world demand across different temporal and geographical dimensions, proving crucial for managing energy supply and demand amid unforeseen events. This approach marks a paradigm shift in STLF strategies, emphasizing the integral role of

language processing in unifying diverse unstructured information.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (72331009), the Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS), the Shenzhen Key Lab of Crowd Intelligence Empowered Low-Carbon Energy Network (ZDSYS20220606100601002), and Shenzhen Stability Science Program 2023.

References

- [1] Gross G, Galiana FD. Short-term load forecasting. *Proc IEEE* 1987;75(12):1558–73.
- [2] Zhao WX, Zhou K, Li J, Tang T, Wang X, Hou Y, et al. A survey of large language models. 2023. arXiv:2303.18223.
- [3] Papalexopoulos AD, Hesterberg TC. A regression-based approach to short-term system load forecasting. *IEEE Trans Power Syst* 1990;5(4):1535–47.
- [4] Huang SJ, Shih KR. Short-term load forecasting via ARMA model identification including non-gaussian process considerations. *IEEE Trans Power Syst* 2003;18(2):673–9.
- [5] Christiaanse W. Short-term load forecasting using general exponential smoothing. *IEEE Trans Power Apparatus Syst* 1971;90(2):900–11.
- [6] Chen BJ, Chang MW, Lin CJ. Load forecasting using support vector machines: a study on EUNITE competition 2001. *IEEE Trans Power Syst* 2004;19(4):1821–30.
- [7] Dudek G. Short-term load forecasting using random forests. In: *Proceedings of the 7th IEEE International Conference Intelligent Systems*; 2014 Sep 24–26; Warsaw, Poland. Springer; 2015. p. 821–8.
- [8] Hippert HS, Pedreira CE, Souza RC. Neural networks for short-term load forecasting: a review and evaluation. *IEEE Trans Power Syst* 2001;16(1):44–55.
- [9] Torres JF, Hadjout D, Sebaa A, Martínez-Álvarez F, Troncoso A. Deep learning for time series forecasting: a survey. *Big Data* 2021;9(1):3–21.
- [10] Xia M, Shao H, Ma X, de Silva CW. A stacked GRU-RNN-based approach for predicting renewable energy and electricity load for smart grid operation. *IEEE Trans Industr Inform* 2021;17(10):7050–9.
- [11] Liu C, Jin Z, Gu J, Qiu C. Short-term load forecasting using a long short-term memory network. In: *Proceedings of 2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*; 2017 Sep 26–29; IEEE; 2017. p. 1–6.
- [12] Bedi J. Attention based mechanism for load time series forecasting: AN-LSTM. In: *Proceedings of Artificial Neural Networks and Machine Learning*; 2020 Sep 15–18; Bratislava, Slovakia. Springer; 2020. p. 838–49.
- [13] Qu K, Si G, Shan Z, Wang Q, Liu X, Yang C. Forwardformer: efficient transformer with multi-scale forward self-attention for day-ahead load forecasting. *IEEE Trans Power Syst* 2023.
- [14] Cecchini M, Aytug H, Koehler GJ, Pathak P. Making words work: using financial text as a predictor of financial events. *Decis Support Syst* 2010;50(1):164–75.
- [15] Schumaker RP, Chen H. Textual analysis of stock market prediction using breaking financial news: the AZFin text system. *ACM Trans Inf Syst* 2009;27(2):1–19.
- [16] Schumaker RP, Chen H. A discrete stock price prediction engine based on financial news. *Computer* 2010;43(1):51–6.
- [17] Bai Y, Camal S, Michiorri A. News and load: a quantitative exploration of natural language processing applications for forecasting day-ahead electricity system demand. *IEEE Trans Power Syst* 2024;39(5):6222–34.
- [18] Obst D, de Vilmarést J, Goude Y. Adaptive methods for short-term electricity load forecasting during COVID-19 lockdown in France. *IEEE Trans Power Syst* 2021;36(5):4754–63.
- [19] Zhou Z, Ren C, Xu Y. Forecasting masked-load with invisible distributed energy resources based on transfer learning and Bayesian tuning. *Energy Convers Econ* 2024;5(5):316–26.
- [20] Joshi M, Das D, Gimpel K, Smith NA. Movie reviews and revenues: an experiment in text regression. In: *Proceedings of Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*; 2010 Jun; Los Angeles, CA, USA. ACL; 2010. p. 293–6.
- [21] Zhang X, Saleh H, Younis EM, Sahal R, Ali AA. Predicting coronavirus pandemic in real-time using machine learning and big data streaming system. *Complexity* 2020;2020:1–10.
- [22] Rodrigues F, Markou I, Pereira FC. Combining time-series and textual data for taxi demand prediction in event areas: a deep learning approach. *Inf Fusion* 2019;49:120–9.
- [23] Chowdhary K. Natural language processing. In: *Fundamentals of artificial intelligence*. New Delhi: Springer; 2020. p. 603–49.
- [24] Berrios W, Mittal G, Thrush T, Kiela D, Singh A. Towards language models that can see: computer vision through the lens of natural language. 2023. arXiv:2306.16410.
- [25] Ghosal D, Majumder N, Mehrish A, Poria S. Text-to-audio generation using instruction-tuned LLM and latent diffusion model. 2023. arXiv:2304.13731.
- [26] Jin M, Wen Q, Liang Y, Zhang C, Xue S, Wang X, et al. Large models for time series and spatio-temporal data: a survey and outlook. 2023. arXiv:2310.10196.
- [27] Jin M, Wang S, Ma L, Chu Z, Zhang JY, Shi X, et al. Time-LLM: time series forecasting by reprogramming large language models. 2023. arXiv:2310.01728.
- [28] Garza A, Mergenthaler-Canseco M. TimeGPT-1. 2023. arXiv:2310.03589.
- [29] Yu X, Chen Z, Ling Y, Dong S, Liu Z, Lu Y. Temporal data meets LLM—explainable financial time series forecasting. 2023. arXiv:2306.11025.
- [30] Xue H, Voutharoja BP, Salim FD. Leveraging language foundation models for human mobility forecasting. In: *Proceedings of the 30th international conference on advances in geographic information systems*; 2022 Nov 1–4; Seattle, WA, USA. ACM; 2022. p. 1–9.
- [31] Wang X, Fang M, Zeng Z, Cheng T. Where would I go next? Large language models as human mobility predictors. 2023. arXiv:2308.15197.
- [32] Liu X, McDuff D, Kovacs G, Galatzer-Levy I, Sunshine J, Zhan J, et al. Large language models are few-shot health learners. 2023. arXiv:2305.15525.
- [33] Karim SA, Alwi SA. Electricity load forecasting in UTP using moving averages and exponential smoothing techniques. *Appl Math Sci* 2013;7(77–80):4003–14.
- [34] Ryu S, Noh J, Kim H. Deep neural network based demand side short term load forecasting. *Energies* 2016;10(1):3.
- [35] Sadaei HJ, de Lima e Silva PC, Guimarães FG, Lee MH. Short-term load forecasting by using a combined method of convolutional neural networks and fuzzy time series. *Energy* 2019;175:365–77.
- [36] Wu H, Hu T, Liu Y, Zhou H, Wang J, Long M. TimesNet: Temporal 2D-variation modeling for general time series analysis. In: *Proceedings of Eleventh International Conference on Learning Representations*; 2022 May 1–5; Kigali, Rwanda. ICLR; 2022.
- [37] Kong W, Dong ZY, Jia Y, Hill DJ, Xu Y, Zhang Y. Short-term residential load forecasting based on LSTM recurrent neural network. *IEEE Trans Smart Grid* 2017;10(1):841–51.
- [38] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. 2017. arXiv:1706.03762.

- [39] Du J, Cheng Y, Zhou Q, Zhang J, Zhang X, Li G. Power load forecasting using BiLSTM-attention. *IOP Conf Ser Earth Environ Sci* 2020;440:032115.
- [40] Zhou H, Zhang S, Peng J, Zhang S, Li J, Xiong H, et al. Informer: beyond efficient transformer for long sequence time-series forecasting. *AAAI Conf Artif Intell* 2021;35(12):11106–15.
- [41] Wu H, Xu J, Wang J, Long M. Autoformer: decomposition transformers with auto-correlation for long-term series forecasting. *Adv Neural Inf Process Syst* 2021;34:22419–30.
- [42] Zhou T, Ma Z, Wen Q, Wang X, Sun L, Jin R. FEDformer: frequency enhanced decomposed transformer for long-term series forecasting. In: *Proceedings of International conference on machine learning*; 2022 Jul 17–23; Baltimore, MD, USA. PMLR; 2022. p. 27268–86.
- [43] Liu S, Yu H, Liao C, Li J, Lin W, Liu AX, et al. Pyraformer: low-complexity pyramidal attention for long-range time series modeling and forecasting. In: *Proceedings of International conference on learning representations*; 2022 Apr 25; online. OpenReview; 2022.
- [44] Nie Y, Nguyen NH, Sinthong P, Kalagnanam J. A time series is worth 64 words: long-term forecasting with transformers. 2022. arXiv:2211.14730.
- [45] Liu Y, He T, Zhang H, Wu H, Wang S, Ma L, et al. iTransformer: inverted transformers are effective for time series forecasting. 2023. arXiv:2310.06625.
- [46] Zhou T, Ma Z, Wang X, Wen Q, Sun L, Yao T, et al. FiLM: frequency improved Legendre memory model for long-term time series forecasting. *Adv Neural Inf Process Syst* 2022;35:12677–12690.
- [47] Zhao Z, Xia C, Chi L, Chang X, Li W, Yang T, et al. Short-term load forecasting based on the transformer model. *Information* 2021;12(12):516.
- [48] Wang C, Wang Y, Ding Z, Zheng T, Hu J, Zhang K. A transformer-based method of multienergy load forecasting in integrated energy system. *IEEE Trans Smart Grid* 2022;13(4):2703–14.
- [49] Ryu S, Yu Y. Quantile-mixer: a novel deep learning approach for probabilistic short-term load forecasting. *IEEE Trans Smart Grid* 2024;15(2):2237–50.
- [50] Taylor SJ, Letham B. Forecasting at scale. *Am Stat* 2018;72(1):37–45.
- [51] Devlin J, Chang MW, Lee K, Toutanova K. Bert: pre-training of deep bidirectional transformers for language understanding. 2018. arXiv:1810.04805.
- [52] Lewis M, Liu Y, Goyal N, Ghazvininejad M, Mohamed A, Levy O, et al. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. 2019. arXiv:1910.13461.
- [53] GPT-4V(ision) system card. Report. San Francisco: OpenAI; 2023.
- [54] Xue H, Salim FD. Prompt-based time series forecasting: a new task and dataset. 2022. arXiv:2210.08964.
- [55] Wei J, Wang X, Schuurmans D, Bosma M, Ichter B, Xia F, et al. Chain-of-thought prompting elicits reasoning in large language models. *Adv Neural Inf Proc Syst* 2022;35:24824–37.
- [56] Shinn N, Labash B, Gopinath A. “Reflexion: an autonomous agent with dynamic memory and self-reflection. 2023. arXiv:2303.11366.
- [57] Cai T, Wang X, Ma T, Chen X, Zhou D. Large language models as tool makers. 2023. arXiv:2305.17126.
- [58] Yao S, Zhao J, Yu D, Du N, Shafran I, Narasimhan K, et al. React: synergizing reasoning and acting in language models. 2022. arXiv:2210.03629.
- [59] Park JS, O’Brien J, Cai CJ, Morris MR, Liang P, Bernstein MS. Generative agents: Interactive simulacra of human behavior. In: *Proceedings of the 36th annual ACM symposium on user interface software and technology*; 2023 Oct 29–Nov 1; San Francisco, CA, USA. ACM; 2023. p. 1–22.
- [60] Du Y, Li S, Torralba A, Tenenbaum JB, Mordatch I. Improving factuality and reasoning in language models through multiagent debate. 2023. arXiv:2305.14325.
- [61] Touvron H, Lavril T, Izacard G, Martinet X, Lachaux MA, Lacroix T, et al. LLaMA: open and efficient foundation language models. 2023. arXiv:2302.13971.
- [62] Elfwing S, Uchibe E, Doya K. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Netw* 2018;107:3–11.
- [63] Zhang B, Sennrich R. Root mean square layer normalization. In: *Proceedings of Advances in Neural Information Processing Systems*; 2019 Dec 8–14; Vancouver, BC, Canada. University Library of Zurich; 2019.
- [64] Su J, Ahmed M, Lu Y, Pan S, Bo W, Liu Y. Roformer: enhanced transformer with rotary position embedding. *Neurocomputing* 2024;568:127063.
- [65] Hu EJ, Shen Y, Wallis P, Allen-Zhu Z, Li Y, Wang S, et al. LoRA: low-rank adaptation of large language models. 2021. arXiv:2106.09685.
- [66] Touvron H, Martin L, Stone K, Albert P, Almahairi A, Babaei Y, et al. Llama 2: open foundation and fine-tuned chat models. 2023. arXiv:2307.09288.
- [67] Zeng A, Chen M, Zhang L, Xu Q. Are transformers effective for time series forecasting? In: *Proceedings of the AAAI Conference on Artificial Intelligence*; 2023 Feb 7–14; Washington, DC, USA. Association for the Advancement of Artificial Intelligence; 2023. p. 11121–8.
- [68] Zhou T, Niu P, Wang X, Sun L, Jin R. One fits all: power general time series analysis by pretrained LM. *Adv Neural Inf Process Syst* 2023;36:43322–55.

Declaration of interests

☐ 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.

☒ The author is an Editorial Board Member/Editor-in-Chief/Associate Editor/Guest Editor for *[Journal name]* and was not involved in the editorial review or the decision to publish this article.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: