Article | Received 9 May 2026; Revised 7 June 2026; Accepted 11 June 2026; Published 25 June 2026
https://doi.org/10.55092/mt20260002

A cross-modal alignment-based time-series large model for water supply networks flow forecasting

Tao Yang[1], Hanqi Gui[1], Juan Xu[2,3,*], Xiaochuan Li[4], Xu Ding[5] and Yongbin Liu[3]

[1] School of Computer and Information, Hefei University of Technology, Hefei, China
[2] Intelligent Manufacturing Institute of HFUT, Hefei, China
[3] School of Electrical Engineering and Automation, Anhui University, Hefei, China
[4] School of Electrical Engineering and Automation, Hefei University of Technology, Hefei, China
[5] School of Mechanical Engineering, Hefei University of Technology, Hefei, China

* Correspondence author; E-mail: xujuan@ahu.edu.cn.

Highlights:

- A cross-modal alignment framework is proposed to bridge numerical time-series features and large language model semantic representations.
- The proposed model enhances long-term flow forecasting accuracy for real-world water supply networks under non-stationary flow dynamics, stochastic demand variations, and multi-scale temporal fluctuations.
- Extensive experiments demonstrate superior forecasting performance and robustness compared with conventional deep learning and transformer-based methods.

Abstract: Water supply networks, as critical urban infrastructure, play an essential role in ensuring stable city operations. Accurate flow forecasting is therefore of great significance for optimizing operational scheduling, reducing energy consumption, and maintaining system stability. With the strong capability of large language models (LLM) in sequence modeling and representation learning, their application to time-series forecasting has become an emerging research direction. However, a key challenge lies in the modality gap between numerical time-series data and the semantic embedding space of language models. To address this issue, this paper proposes a cross-modal alignment-based time-series foundation model for forecasting. The proposed method constructs a mapping between time-series features and the semantic embedding space, enabling effective projection of numerical sequences into a semantic domain. Furthermore, a cross-modal alignment mechanism is designed to enhance feature fusion, thereby improving the model's ability to capture multi-scale periodic patterns, long-term temporal trends, and stochastic demand fluctuations commonly observed in water distribution systems. Experimental results demonstrate that the proposed method consistently outperforms baseline approaches across different prediction horizons in terms of mean absolute error (MAE) and mean squared erro (MSE), verifying the effectiveness and strong generalization capability of the cross-modal alignment strategy in water distribution network flow forecasting.

Keywords: water supply network; flow forecasting; time-series forecasting; large language models; cross-modal alignment; time-series foundation model

1. Introduction

Water distribution networks, as critical urban infrastructure, play a vital role in urban operations, and accurate flow forecasting is essential for optimizing scheduling, reducing energy consumption, and ensuring system stability [1]. Deep learning–based time-series methods have achieved notable progress in this task [2]; however, they remain limited by fixed modeling paradigms and often struggle to model multi-scale temporal patterns, stochastic demand fluctuations, and non-stationary flow dynamics in long-sequence forecasting scenarios.

Recent time-series foundation models typically convert continuous numerical sequences into discrete or embedding-based representations similar to text, thereby leveraging the strong contextual modeling capabilities of pretrained language models for forecasting [3,4]. Nevertheless, a fundamental modality gap exists: textual data consists of discrete symbols with explicit semantic boundaries and linguistic structures, whereas time-series data is continuous, emphasizing temporal dependency, periodicity, and dynamic variation [5,6]. This discrepancy prevents pretrained language models from directly and effectively capturing time-series characteristics [7], thereby limiting their performance in practical forecasting tasks.

In particular, water demand flow series in distribution networks exhibit multi-scale periodicity, long-term trends, stochastic fluctuations, and abrupt demand changes caused by weather conditions, holidays, unexpected events, and variations in consumer behavior [8]. These characteristics introduce strong temporal dependencies, non-stationarity, and uncertainty [9]. Relying solely on a single-modality time-series representation is therefore insufficient to fully capture such complex patterns [10], leading to degraded forecasting accuracy and limited generalization ability.

To address these challenges, this paper introduces a cross-modal alignment learning mechanism for water distribution network flow forecasting and proposes a dual-branch framework that integrates time-series and textual semantic modalities. Specifically, the model constructs a time-series branch and a text semantic branch, and learns a mapping from time-series representations to the semantic embedding space of language models via a cross-modal matching module, enabling alignment within a unified semantic space. Furthermore, a feature alignment constraint and an output consistency constraint are designed to jointly optimize the two modalities during training. The feature alignment constraint reduces the distribution gap between time-series and semantic representations, while the output consistency constraint enforces agreement between dual-branch predictions, thereby improving the model's stability in capturing future trends.

The main contributions of this work are as follows:

(1) We propose a cross-modal alignment-based time-series foundation forecasting framework that integrates a time-series branch and a text semantic branch. A cross-modal matching module is designed to map time-series features into the semantic space of pretrained language models, improving forecasting accuracy and stability in water distribution networks.

(2) We design a cross-modal feature matching mechanism and a joint dual-branch optimization

strategy. The word embedding matrix of a pretrained language model is leveraged as a semantic knowledge base, and cross-modal attention with PCA-based dimensionality reduction is employed to align time-series and semantic spaces. Additionally, Low-Rank Adaptation (LoRA)-based fine-tuning is applied to the time-series branch. Together with feature alignment and output consistency losses, the proposed strategy enhances the model's capability to capture complex temporal patterns while substantially reducing trainable parameters and GPU memory consumption.

2. Related work

2.1. Water supply networks flow forecasting

Water supply networks flow forecasting is essentially a time-series forecasting problem, and its development is closely related to the evolution of time-series modeling methods. Early studies mainly relied on statistical models represented by Autoregressive Integrated Moving Average(ARIMA) [11]. Based on the time-series analysis framework proposed by Box and Jenkins [12], such methods can effectively characterize trends and periodicity through differencing and seasonal modeling, and they show stable performance in short-term forecasting tasks. However, their linear assumptions limit performance in complex nonlinear and dynamically changing scenarios.

With the growth of data scale, research gradually shifted toward machine learning methods, such as Artificial Neural Networks (ANN) [13] and Support Vector Machines (SVM) [14]. These methods possess strong nonlinear fitting capability and compensate for some of the limitations of statistical models to a certain extent, but they rely heavily on feature engineering and have difficulty fully capturing complex temporal dependencies.

In recent years, deep learning methods have become mainstream [15], among which Recurrent Neural Networks (RNN) [16] and their variants, including Long Short-Term Memory (LSTM) [17] and Gated Recurrent Unit (GRU) [18], have been widely applied to time-series forecasting tasks. Through gating mechanisms, these models can effectively capture long-term dependencies and have significantly improved forecasting accuracy in tasks such as energy load forecasting and water demand forecasting [19]. However, their recurrent structure limits parallel computation and leads to insufficient efficiency and global modeling capability for ultra-long sequences.

2.2. Large language models for time-series forecasting

To address the above issues, Transformer-based methods have gradually emerged. Proposed by Vaswani *et al.* [20], the Transformer achieves global dependency modeling through the self-attention mechanism and offers strong parallel computation capability. On this basis, a series of improved models, such as Informer [21], Autoformer [22], and FEDformer [23], have been proposed in succession. By adopting strategies such as sparse attention, sequence decomposition, and frequency-domain modeling, these models further improve the efficiency and accuracy of long-sequence forecasting. In addition, models such as PatchTST [24] and TimesNet [25] have continued to advance forecasting performance through multi-scale modeling and patch-based strategies.

Building upon this line of work, time-series large models have further expanded the paradigm of time-series modeling in recent years. Inspired by Large Language Models (LLM), related studies can be broadly divided into two categories. One line transfers language models to time-series tasks and performs forecasting through prompt learning or sequence reprogramming, as exemplified by PromptCast [26], LLMTime [27], and Time-LLM [28]. The other line develops dedicated time-series foundation models, such as TimeGPT [29], Chronos[30], TimesFM [31], and Time-MoE [32]. Through large-scale pre-training, these methods learn general-purpose temporal representations, significantly enhancing model generalization and transferability, and they have shown strong performance in zero-shot or few-shot settings. Furthermore, recent studies have explored new directions for time-series foundation models. ChatTime [7] bridges numerical time-series data and textual information through a unified multimodal framework, enabling cross-modal understanding and reasoning. Meanwhile, Multi-scale Finetuning [33] improves the adaptability of encoder-based time-series foundation models by leveraging temporal patterns at different scales, further enhancing downstream forecasting performance.

Overall, time-series forecasting methods have evolved from statistical models to machine learning, then to deep learning, and further to time-series large models [34]. Recent advances have not only increased model scale but also expanded modeling paradigms toward multimodal learning, mixture-of-experts architectures, and parameter-efficient adaptation strategies [7,32,33]. Although model capability has continuously improved, data in practical engineering scenarios such as water supply networks usually exhibit strong periodicity, trend, and local fluctuation characteristics, and existing methods still suffer from limitations in structural adaptability and stability. Therefore, designing models in a targeted manner for specific application scenarios remains an important research direction.

3. Methodology

To fully exploit the semantic modeling capability of pre-trained language models while preserving the temporal structural information of time-series data, this paper constructs a time-series forecasting framework based on cross-modal alignment. The overall model architecture is shown in Figure 1 and mainly consists of four components: a time-series embedding module, a cross-modal feature alignment module, a dual-branch semantic encoding module, and a prediction module. These components are jointly optimized using a multi-objective loss function.

The model first embeds the input historical flow sequence to map the original numerical sequence into a high-dimensional feature representation. It then maps the time-series features to the semantic space of the language model through the cross-modal feature alignment module, thereby achieving semantic alignment between the time-series modality and the textual modality. On this basis, a textual semantic branch and a time-series branch are constructed for deep semantic encoding. Finally, a prediction head outputs the future flow forecasts.

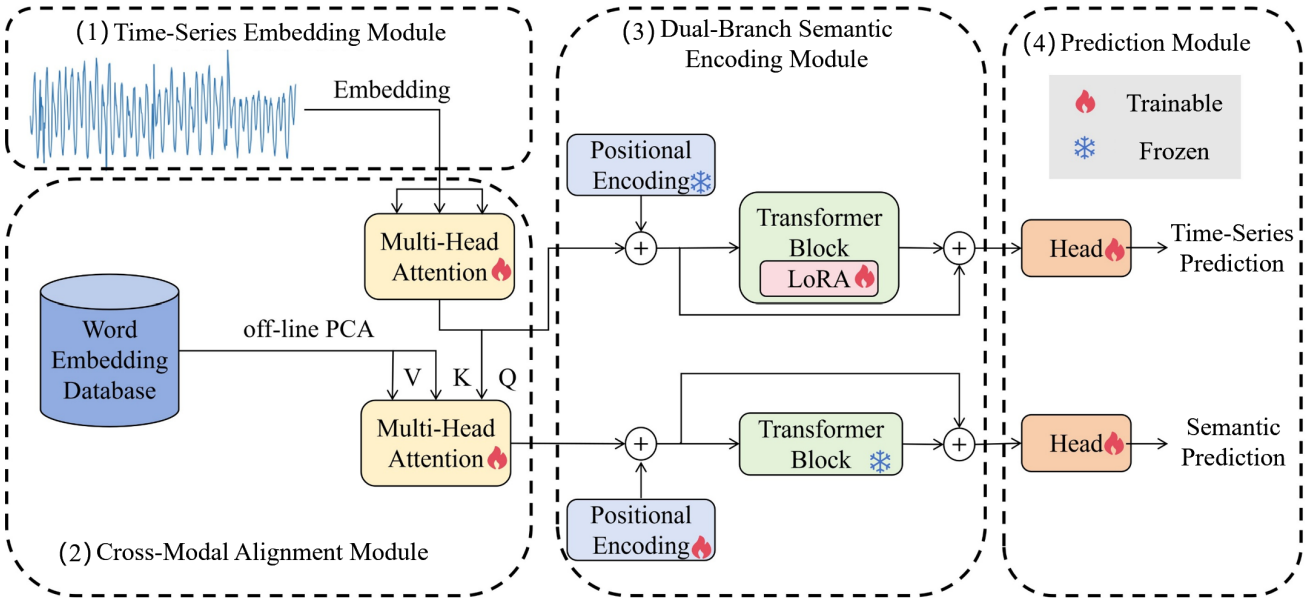


Figure 1. The framework of the proposed method.The framework includes four modules: **(1)** time- series embedding, where historical flow data are transformed into latent representations; **(2)** cross-modal alignment, which maps temporal features into the semantic space of the pre-trained GPT-2 model through a learnable alignment module and PCA-based dimensionality reduction; **(3)** dual-branch semantic encoding, where aligned temporal and semantic features are jointly processed using LoRA-enhanced LLM encoders; and **(4)** forecasting, where fused multi-modal representations are used to predict future flow. Feature alignment loss and output consistency loss are jointly optimized to improve forecasting performance.

3.1. Time-series embedding module

Water supply networks flow data exhibit evident periodic variation and trend variation. Therefore, feature representation learning must first be performed on the original time series at the model input stage. Let the historical flow sequence be:

$$X = \{x_1, x_2, ..., x_T\} \tag{1}$$

where T denotes the length of the historical time window, x_t denotes the observed flow value at time t, and C denotes the number of monitoring nodes. Then, the input sequence can be expressed as:

$$X \in \mathbb{R}^{T \times C} \tag{2}$$

To map the original time series into a high-dimensional feature space, a linear embedding layer is first used for feature transformation:

$$Z = XW_e + b_e \tag{3}$$

where $W_e \in \mathbb{R}^{C \times d}$, $b_e \in \mathbb{R}^d$, and d denotes the embedding feature dimension. Since the object of this study is univariate water flow data, $C = 1$ and $Z \in \mathbb{R}^{T \times d}$. After the embedding layer, the original time series is represented as a sequence of feature vectors of length T, where each time step corresponds to a d-dimensional feature vector. This representation maps the original flow sequence into a unified high-dimensional feature space, providing the basis for subsequent cross-modal feature alignment.

3.2. *Cross-modal feature alignment module*

Previous studies have shown that the word embedding matrix in a pre-trained LLM forms a structured semantic representation space in which semantic relationships among different words can be characterized through vector similarity. Therefore, the embedding layer essentially reflects the distributional characteristics of linguistic-modal input data in the pre-trained language model. Although this property provides potential advantages for cross-modal modeling, most existing LLM-based time-series forecasting methods do not fully exploit this distributional structure. Instead, they directly project time-series data into a feature space with the same input dimension as the language model. However, since pre-trained language models are primarily trained on large-scale text corpora, there remains a substantial discrepancy between the word-vector space and the time-series feature space. It is therefore necessary to construct a cross-modal feature alignment module to establish semantic connections between the time-series modality and the linguistic modality.

First, preliminary modeling is performed on the embedded time-series features, and the initial time-series features are obtained through a multi-head self-attention mechanism:

$$\tilde{F}_{time} = \mathrm{MultiHead}(Z) \tag{4}$$

After obtaining the high-level representation of the time series, it is necessary to introduce the semantic representation space of the language model as the target space for cross-modal matching in order to achieve semantic alignment between the time-series modality and the textual modality. The word embedding database used in the cross-modal feature alignment module is not additionally trained, but is directly derived from the word embedding matrix inside the pre-trained language model GPT-2. Specifically, during training on large-scale corpora, the pre-trained language model learns a word embedding matrix with a vocabulary size of $|V|$ and an embedding dimension of d, which is used to map discrete tokens into a continuous semantic vector space. This matrix essentially constitutes a word-vector dictionary containing rich prior semantic knowledge, where each row corresponds to the semantic representation of one token in the vocabulary. Therefore, in this paper, the pre-trained word embedding matrix is regarded as a semantic knowledge base for the textual modality and is denoted as:

$$D \in \mathbb{R}^{|V| \times d} \tag{5}$$

By directly using the word embedding parameters of the pre-trained model, the semantic structure and inter-word relationships learned by the language model from large-scale textual data can be inherited effectively, allowing time-series features to align with a representation space that follows a realistic semantic distribution during subsequent cross-modal mapping. At the same time, this strategy avoids the extra cost of constructing or training an additional textual semantic repository, thereby ensuring the efficiency and stability of the overall training process. Considering that the vocabulary size is usually large, directly using all word embeddings for cross-modal attention would incur high computational complexity. Therefore, principal component analysis (PCA) is further adopted to reduce and compress the dimensionality of the word embedding matrix:

$$\hat{D} = \mathrm{PCA}_k(D) \tag{6}$$

Thus, the dimensionality-reduced set of principal word vectors is obtained as:

$$\hat{D} \in \mathbb{R}^{k \times d} \tag{7}$$

where $k \ll |V|$.

Subsequently, a cross-modal attention mechanism is used to establish the mapping between time-series features and the word-vector space. For the time-series features $\tilde{F}_{time}$ and semantic word vectors $\hat{D}$, the query, key, and value are computed respectively as:

$$Q = \tilde{F}_{time} W_q \tag{8}$$

$$K = \hat{D} W_k \tag{9}$$

$$V = \hat{D} W_v \tag{10}$$

where $W_q, W_k, W_v \in \mathbb{R}^{d \times d}$.

The cross-modally aligned feature representation is given by:

$$F_{align} = \text{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right) V \tag{11}$$

Through this process, the time-series features can obtain corresponding representations in the semantic space of the language model, thereby realizing semantic alignment between the time-series modality and the textual modality.

3.3. Dual-branch semantic encoding module

After obtaining the cross-modally aligned feature representation, this paper constructs a dual-branch semantic encoding module to perform semantic modeling on the textual source branch and the time-series target branch, respectively, so as to fully exploit temporal features under different modalities.

For the textual source branch, positional encoding is first introduced to preserve the positional information of the sequence, and the encoded sequence is then fed into a pre-trained Transformer Block for feature extraction. The Transformer parameters in this branch remain frozen, thereby preserving the semantic representation capability already acquired by the pre-trained language model. The computation can be expressed as:

$$F^{text} = \text{Transformer}(F_{align} + PE) \tag{12}$$

For the time-series target branch, positional encoding is likewise introduced first, and the sequence is then fed into a Transformer Block for modeling. Unlike the textual branch, directly fine-tuning all parameters of the pre-trained language model would introduce substantial computational overhead and GPU memory consumption, while increasing the risk of overfitting due to the relatively limited scale of water supply flow datasets. Therefore, this paper adopts LoRA as a parameter-efficient fine-tuning strategy. Specifically, LoRA inserts trainable low-rank matrices into selected linear layers while keeping the original pre-trained weights frozen, thereby significantly reducing the number of trainable parameters and computational costs. At the same time, it enables effective adaptation of the language model to the

water supply flow forecasting task. Accordingly, the LoRA mechanism is introduced into the Transformer Block of this branch, as shown in Figure 2.

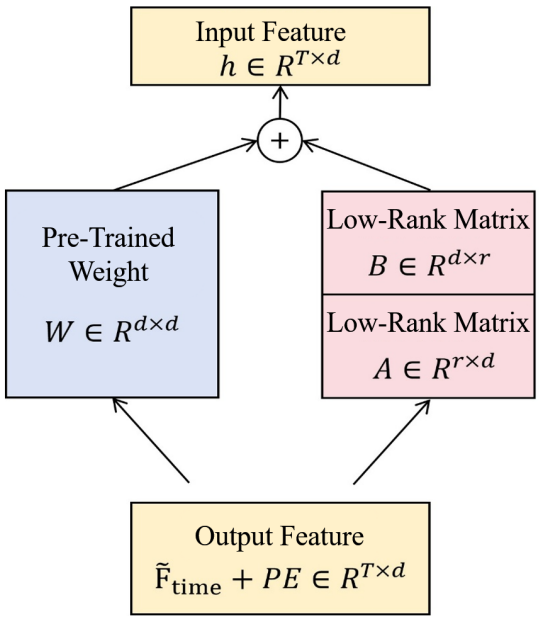


Figure 2. LoRA architecture.

Specifically, for the weight matrix W in the Transformer, after introducing LoRA, the forward computation of a single linear layer can be expressed as:

$$h = W(\tilde{F}_{time} + PE) + BA(\tilde{F}_{time} + PE) \tag{13}$$

where W denotes the frozen pre-trained weight, and $A \in \mathbb{R}^{r\times d}$, $B \in \mathbb{R}^{d\times r}$ denote learnable low-rank matrices. It should be noted that the sequence length T exists only as a batch dimension in the input and does not participate in the dimensional definitions of the low-rank matrices A and B. During forward computation, the input first passes through matrix A to reduce the feature dimension from d to r, and then through matrix B to restore the dimension from r back to d, thereby realizing low-rank adaptation. Here, A is initialized with a Gaussian distribution, $A \sim \mathcal{N}(0, \sigma^2)$, while B is initialized as a zero matrix.

Therefore, after introducing LoRA, the semantic encoding process of the time-series branch can be expressed as:

$$F^{time} = \text{Transformer}_{\text{LoRA}}(\tilde{F}_{time} + PE) \tag{14}$$

In this way, the model preserves the semantic knowledge learned by the pre-trained language model while enabling efficient task-specific adaptation with only a small number of trainable parameters, making it more suitable for practical forecasting applications.

3.4. Prediction module

The prediction module is used to map the encoded high-dimensional features to the forecasted flow values at future time steps. First, a linear mapping is applied to the output of the textual semantic branch to obtain the semantic prediction result:

$$Y_{text} = F_{text} W_p + b_p \tag{15}$$

where $W_p \in \mathbb{R}^{d \times H}$, $b_p \in \mathbb{R}^H$, and H denotes the forecasting horizon.

Meanwhile, prediction is also performed on the feature representation of the time-series branch:

$$Y_{time} = F_{time} W_t + b_t \tag{16}$$

where W_t and b_t are learnable parameters.

During model inference, the final prediction result is taken as the output of the time-series branch:

$$\hat{Y} = Y_{time} \tag{17}$$

3.5. *Loss function design*

To achieve effective model training and promote collaborative learning between time-series features and the semantic space of the language model, this paper adopts a multi-objective loss function for joint optimization.

First, the mean squared error is used as the predictive supervision loss:

$$L_{pred} = \frac{1}{H} \sum_{i=1}^{H} (y_i - \hat{y}_i)^2 \tag{18}$$

where y_i denotes the ground-truth flow value and $\hat{y}_i$ denotes the predicted value.

To make these pre-trained weights better adapt to time-series data, we align the output of each intermediate layer in the time-series branch with the output of the textual semantic branch. This alignment is promoted through a feature regularization loss, which matches the intermediate features between the two branches and allows the gradient at each intermediate layer to be guided more effectively for better weight updates. Let F_{text}^l and F_{time}^l denote the output features of the textual semantic branch and the time-series branch at the l-th Transformer Block, respectively. Then, the feature regularization loss is defined as follows:

$$L_{align} = \sum_{l=1}^{L} \gamma^{(L-l)} \mathrm{Sim}(F_{time}^l, F_{text}^l) \tag{19}$$

where $\mathrm{Sim}(\cdot)$ denotes the feature similarity function and γ denotes the decay coefficient.

On the basis of the feature regularization loss, we further enforce semantic contextual consistency between the textual modality and the time-series modality. The output consistency loss achieves this by ensuring effective correspondence between the output distributions and thereby addressing discrepancies in the representation space. Such alignment maintains a consistent and unified semantic representation for time-series data and textual data, which contributes to more accurate and reliable forecasting. Specifically, given Y_{text} and Y_{time} from the textual semantic branch and the time-series branch, respectively, the output consistency loss is defined as:

$$L_{cons} = \mathrm{Sim}(Y_{time}, Y_{text}) \tag{20}$$

To avoid damaging the existing knowledge structure of the pre-trained model during fine-tuning while improving training efficiency, this paper adopts a parameter-efficient fine-tuning strategy to adapt the pre-trained LLM to the target task. Specifically, for the time-series target branch, we introduce LoRA and fine-tune the positional encoding weights. The final training objective of the model is:

$$L = L_{pred} + \lambda_1 L_{align} + \lambda_2 L_{cons} \tag{21}$$

where λ_1 and λ_2 are weighting coefficients used to balance the contributions of different loss terms to model training. In the experiments, $\lambda_1 = 0.1$ and $\lambda_2 = 0.05$, which were selected according to validation-set performance.To investigate the influence of the loss weighting coefficients, we conducted additional validation experiments using different combinations of λ_1 and λ_2. The results indicate that excessively small values weaken the effectiveness of cross-modal alignment, while excessively large values may overemphasize auxiliary objectives and negatively affect forecasting accuracy. The selected setting ($\lambda_1 = 0.1$ and $\lambda_2 = 0.05$) achieves the best balance between prediction performance and cross-modal consistency, and is therefore adopted in all experiments.

4. Experiment

4.1. Experimental settings

The datasets used in this study include a pharmaceutical station dataset and a residential community dataset. Both datasets are derived from real water supply networks monitoring systems and record the historical flow variations at key nodes in the network. The main model parameter settings used in the experiments are listed in Table 1. Among them, training parameters are used to control the model optimization process, structural parameters define the network scale, and LLM parameters together with LoRA parameters are used to implement parameter-efficient fine-tuning of the pre-trained language model. In particular, the LoRA rank r controls the dimensionality of the low-rank update matrices, where a smaller value reduces trainable parameters but may limit adaptation capacity, while a larger value increases model flexibility at the cost of higher computation. The scaling factor α regulates the magnitude of the LoRA updates, balancing stability and adaptation strength during fine-tuning. These parameters are therefore crucial for both reproducibility and optimization behavior analysis in LLM-based forecasting tasks. The specific settings of the input sequence length and forecasting horizon will be further described in the subsequent comparative experiments to validate the performance of the model on short-term and long-term forecasting tasks, respectively. The evaluation metrics are the mean absolute error (MAE) and the mean squared error (MSE). In addition, to evaluate the practicality of the proposed model in industrial deployment scenarios, we report its computational efficiency. All experiments are conducted on an NVIDIA RTX 3090 GPU with 24 GB memory. Due to the use of parameter-efficient LoRA fine-tuning, the model significantly reduces GPU memory overhead compared with full fine-tuning. The lightweight adaptation strategy enables efficient inference, making the proposed framework suitable for near real-time monitoring applications in water supply networks.

Table 1. Model parameter settings.

Parameter Category	Parameter Name	Value
Training Parameters	Batch Size	256
	Initial Learning Rate	5×10^{-4}
	Number of Epochs	50
	Early Stopping Patience	5
Model Architecture Parameters	Hidden Dimension d_{model}	768
	Number of Attention Heads	4
	Feedforward Dimension d_{ff}	768
	Dropout Rate	0.3
	Input Dimension *enc_in*	1
	Output Dimension c_{out}	1
LLM Parameters	Number of GPT Layers	6
	Learning Rate Scheduler	type1 + Cosine Annealing
	Maximum Training Cycles T_{max}	20
LoRA Parameters	Rank r	8
	Scaling Factor α	32
	Dropout Rate	0.1

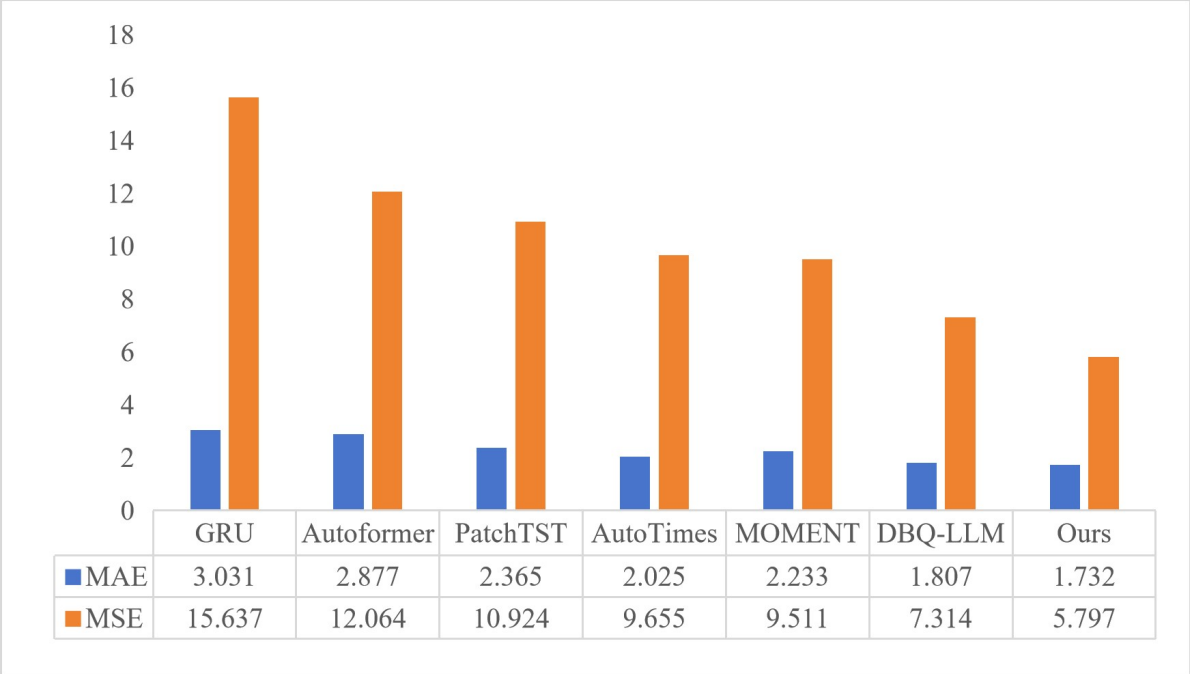
4.2. Comparative experiments

To verify the effectiveness of the proposed method in the water supply networks flow forecasting task, this paper designs comparative experiments covering both short-term and long-term time scales, and comprehensively compares the proposed model with a variety of classical and state-of-the-art time-series forecasting models, including GRU [18], Transformer. [20], Autoformer [22], PatchTST [24], Autotimes [35], MOMENT [36], and DBQ-LLM.

In the short-term forecasting experiment, the input sequence length is set to 96 and the forecasting horizon is set to 16, that is, the historical flow data from the previous 96 time steps are used to predict the variation trend over the next 16 time steps. This setting is mainly used to evaluate the model's ability to capture flow fluctuation patterns over a short time range. In the long-term forecasting experiment, the input sequence length is set to 512 and the forecasting horizon is set to 64, that is, historical data over a longer time range are used to predict flow changes over the next 64 time steps. Compared with the short-term forecasting task, long-term forecasting focuses more on evaluating the model's ability to capture long-range temporal dependencies and complex temporal patterns. By setting both short-term and long-term forecasting scenarios, the predictive performance of the model can be comprehensively evaluated across different time scales. Figure 3 and Figure 4 present the experimental results of the models on the long-term and short-term forecasting tasks, respectively.

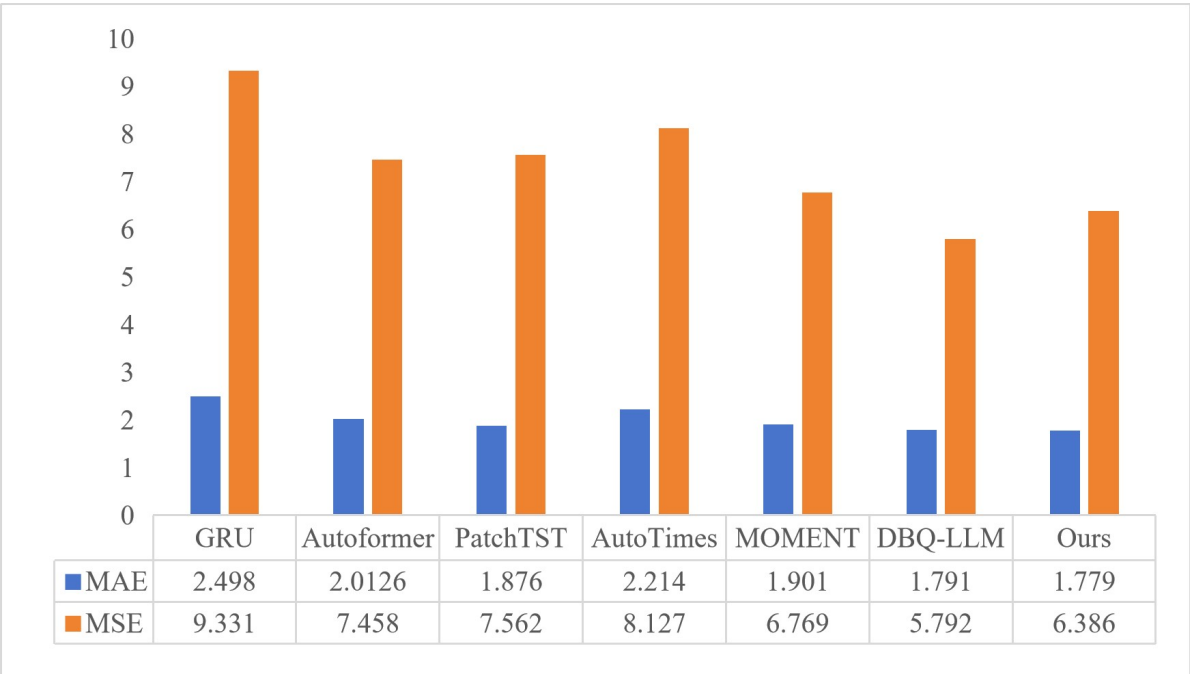
The experimental results demonstrate clear differences in forecasting performance across various models. The traditional GRU model exhibits higher prediction errors compared to other models in both short-term and long-term forecasting tasks. Transformer-based models, such as Autoformer and PatchTST, perform better due to their ability to capture global dependencies through the self-attention mechanism. More recent time-series foundation models, including AutoTimes and MOMENT, also

outperform traditional deep learning models by leveraging large-scale data pre-training to learn generalized temporal representations.



	GRU	Autoformer	PatchTST	AutoTimes	MOMENT	DBQ-LLM	Ours
MAE	3.031	2.877	2.365	2.025	2.233	1.807	1.732
MSE	15.637	12.064	10.924	9.655	9.511	7.314	5.797

Figure 3. Comparison of long-term forecast experiment results.



	GRU	Autoformer	PatchTST	AutoTimes	MOMENT	DBQ-LLM	Ours
MAE	2.498	2.0126	1.876	2.214	1.901	1.791	1.779
MSE	9.331	7.458	7.562	8.127	6.769	5.792	6.386

Figure 4. Comparison of short-term forecast experiment results.

In the long-term forecasting task, the proposed method achieves superior performance, with MAE and MSE values of 1.732 and 5.797, respectively, outperforming all comparison models. Compared to the GRU model, the proposed method reduces MAE by approximately 42.8% and MSE by 62.9%, highlighting a significant improvement. Even against Transformer-based models, such as PatchTST, the proposed method shows a clear advantage, reducing MAE from 2.365 to 1.732 and MSE from

10.924 to 5.797. These results suggest that, despite the Transformer architecture's ability to model global dependencies, it has limitations when handling long-duration time-series data. The proposed method also outperforms recent time-series foundation models, reducing MAE from 2.025 (AutoTimes) and 2.233 (MOMENT) to 1.732. Furthermore, when compared to DBQ-LLM, the proposed method also reduces MAE and MSE to 1.732 and 5.797, respectively. This demonstrates that the cross-modal feature alignment mechanism effectively enhances the model's representational power, enabling it to better leverage the structured semantic information from pre-trained language models and improve its ability to model complex temporal patterns, particularly in the context of water supply network flow data, which is influenced by periodic usage, sudden demand shifts, and other dynamic factors.

In the short-term forecasting task, performance differences among the models are less pronounced. The proposed method achieves the best MAE (1.779), slightly outperforming models like DBQ-LLM (1.791) and MOMENT (1.901). However, on the MSE metric, the proposed method (6.386) is slightly higher than DBQ-LLM (5.792), mainly due to the relatively simple temporal dependencies in short-term forecasting tasks. Since most deep learning models capture short-range fluctuations well, the performance gap is not as large. Additionally, the MSE metric is sensitive to outliers, and sudden fluctuations in water flow data can affect performance.

Overall, the results indicate that the proposed method demonstrates strong predictive performance in both short-term and long-term forecasting tasks, with a more pronounced advantage in long-term forecasting. By establishing a semantic alignment between time-series and linguistic modalities, the model's ability to capture complex temporal dependencies is enhanced, resulting in stronger predictive capability for long-term water flow data forecasting.

4.3. Ablation experiments

To further verify the effectiveness of each key module in the proposed model and analyze the impact of different design choices on model performance, this paper designs two categories of ablation experiments using the flow dataset of a residential community in Hefei as the test dataset. The first category focuses on the core modules in the model architecture. By progressively removing the cross-modal feature alignment module, the LLM encoding module, and the LoRA parameter-efficient fine-tuning mechanism, the contribution of each component to forecasting performance is analyzed. The second category focuses on key parameter settings in the cross-modal feature alignment module, particularly the number of principal components selected when applying PCA to the word embedding matrix.

4.3.1. Ablation study on model architecture

The contribution of each component is assessed by comparing the forecasting performance of the complete model (Ours) with three variant architectures. Specifically, "w/o Cross-Modal Alignment" removes the cross-modal feature alignment module and employs only linear mapping to project time-series features directly to the input dimension of the language model, aiming to evaluate the role of this mechanism in reducing the discrepancy between temporal features and the linguistic semantic space. "w/o LLM Encoder" excludes the LLM encoding module and relies solely on temporally encoded features with a simple prediction head for output, to assess the contribution of the LLM in capturing complex long-range

temporal dependencies. “w/o LoRA” freezes the pre-trained language model parameters during training and omits the parameter-efficient fine-tuning strategy, allowing an evaluation of the LoRA strategy’s role in improving model adaptability and training efficiency.

As shown in Table 2, the complete model (Ours) achieves the best performance across both evaluation metrics, with an MAE of 1.732 and an MSE of 5.797. In contrast, when individual modules are removed, the prediction errors increase to varying degrees, indicating the contribution of each module to overall model performance.

Table 2. Ablation study results.

Configuration	MAE	MSE
w/o Cross-Modal Alignment	2.964	8.842
w/o LLM Encoder	3.105	7.931
w/o LoRA	1.865	6.214
Ours	**1.732**	**5.797**

The removal of the cross-modal feature alignment module results in the most substantial performance degradation, with the MAE rising to 2.964 and the MSE increasing to 8.842. This highlights the critical role of the cross-modal feature alignment mechanism, which maps time-series features into the semantic space of the pre-trained language model through multi-head self-attention. By establishing a semantic connection between the two modalities, this mechanism significantly enhances the model’s ability to capture complex temporal patterns. Without this module, time-series features cannot be effectively aligned with the language model’s representation space, weakening the model’s temporal modeling capability.

When the LLM encoding module is removed, the model’s performance further deteriorates, with the MAE reaching 3.105. This indicates that the LLM contributes valuable contextual representations and a structured semantic space learned from large-scale corpora, which enhances the model’s ability to capture long-range temporal dependencies. Without this component, the model essentially reverts to a conventional deep network with significantly reduced representational capacity.

Removing the LoRA mechanism leads to a smaller, but still noticeable, performance drop, with the MAE and MSE increasing to 1.865 and 6.214, respectively. This result suggests that LoRA improves the model’s adaptability and task-specific performance by utilizing low-rank adaptation matrices while largely preserving the original model parameters. Although the performance drop is less severe than in the previous two variants, the removal of LoRA still results in inferior performance relative to the complete model.

In summary, the cross-modal feature alignment module has the most significant impact on performance, as it establishes the essential semantic relationship between time-series and linguistic modalities. The LLM encoding module and the LoRA fine-tuning mechanism further enhance the model’s ability to capture complex temporal dependencies and adapt efficiently to the forecasting task, respectively.

4.3.2. Analysis of the number of PCA principal components

In the cross-modal feature alignment module, PCA is applied to the GPT-2 word embedding matrix to extract the dominant semantic components while reducing the computational burden of cross-modal attention.

Theoretically, PCA preserves the directions with the largest variance in the semantic embedding space, which contain the most representative semantic information learned from large-scale text corpora. Therefore, the number of retained principal components (k) determines the trade-off between semantic information preservation and computational efficiency. If k is too small, important semantic structures may be discarded during dimensionality reduction, weakening the semantic guidance provided to time-series representations. Conversely, an excessively large k may introduce redundant semantic information and increase computational complexity without bringing substantial performance gains. To investigate this trade-off and determine an appropriate value of k, an ablation study is conducted under different PCA dimensions.

Under the condition that other experimental settings remain unchanged, experiments are conducted with different numbers of principal components: $d \in \{100, 300, 500, 700, 900\}$, and the results are compared on the water supply networks flow forecasting task. The experimental results are shown in Table 3.

Table 3. Model performance under different PCA dimensions.

Dimension d	**MAE**	**MSE**
100	1.832	7.349
300	1.813	6.986
500	**1.732**	**5.797**
700	1.759	6.204
900	1.795	6.411

The experimental results indicate that the number of PCA principal components has a clear impact on model forecasting performance. When the number of principal components is small (e.g., $d = 100$), the MAE and MSE are 1.832 and 7.349, respectively. This is because the PCA dimensionality reduction process discards important semantic information, preventing the semantic structure in the word embedding space from being fully preserved.

As the number of principal components increases, forecasting performance gradually improves. When $d = 300$, MAE and MSE decrease to 1.813 and 6.986, respectively, indicating that increasing the number of semantic principal components strengthens semantic representation capability.

When $d = 500$, the model achieves the best performance, with MAE and MSE reaching 1.732 and 5.797, respectively. This result indicates that under this setting, PCA can effectively reduce the feature dimension while preserving the semantic structural information in the word embeddings, thereby enabling the cross-modal feature alignment module to obtain more discriminative semantic representations. However, when the number of principal components is further increased to 700 and 900, model performance shows a certain degree of degradation, likely due to the introduction of redundant features.

In summary, the choice of the number of PCA principal components has an important influence on cross-modal feature representation. Too few principal components lead to semantic information loss, whereas too many introduce redundant features and increase computational complexity. Experimental results show that when $d = 500$, the model achieves a favorable balance between semantic expressive capability and computational efficiency. Therefore, this parameter setting is adopted in all subsequent experiments.

4.4. Visualization analysis of forecasting results

To provide a more intuitive presentation of the model's forecasting performance, this paper selects two representative time windows from the test set for visual analysis, as shown in Figure 5. The window in Figure 5a spans from 00:00 on July 15, 2023, to 23:00 on July 21, 2023. During this period, the flow exhibits sharp fluctuations with pronounced peak variations, making it suitable for evaluating the model's robustness and generalization ability under non-stationary flow dynamics, stochastic demand variations, and multi-scale temporal fluctuations. The window in Figure 5b covers the period from 00:00 on March 1, 2023, to 23:00 on March 7, 2023. In this window, the water flow varies smoothly and displays clear periodic characteristics, providing a basis for assessing the model's forecasting performance under typical operational conditions. In both figures, the blue curve represents the ground-truth flow values, while the orange curve represents the model's predictions.

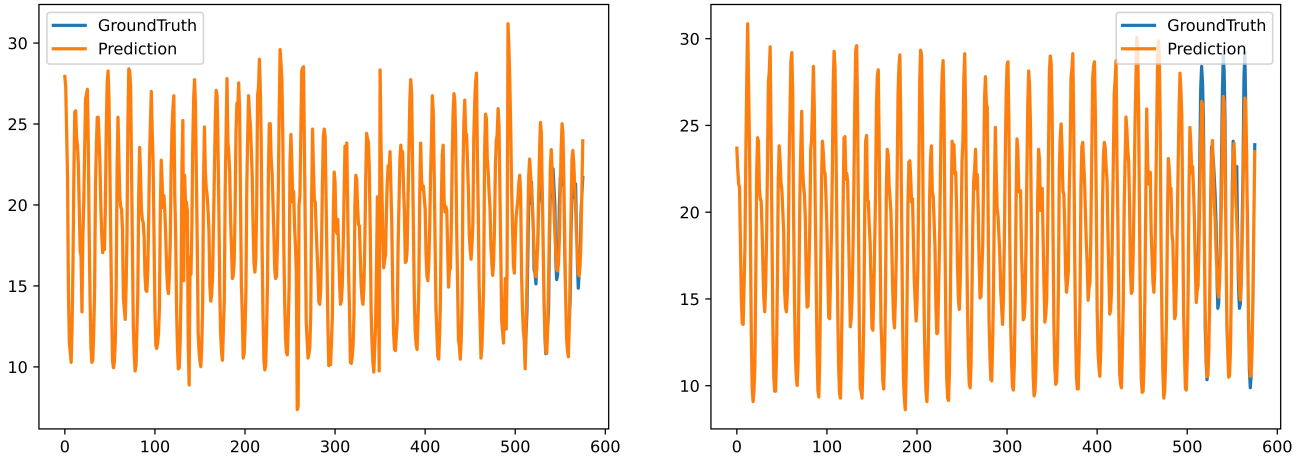


Figure 5. Visualization of prediction results under different testing windows.

As observed from the figure, the proposed model effectively captures the water flow variation trends across both time windows. In the period shown in Figure 5a, the flow data exhibit significant fluctuations, including multiple local peaks and valleys. The model's predictions closely track the trend of the true flow values, maintaining a high degree of consistency with the ground truth for most time steps. This indicates that the model can effectively capture the dynamic variations in the time series. In the period shown in Figure 5b, the flow variation is relatively smooth, with clear periodic fluctuations. The predicted curve almost overlaps with the ground-truth curve, with only slight deviations at a few fluctuation points. This demonstrates that the proposed method can also maintain high forecasting accuracy in stable-flow scenarios.

In summary, the proposed method exhibits strong fitting capabilities across different flow variation patterns. It not only accurately captures the overall trend of water flow changes but also maintains robust forecasting performance at peak and valley positions, further validating its effectiveness and stability in the task of water supply network flow forecasting.

5. Conclusion

For the water supply networks flow forecasting task, this paper proposes a time-series large-model forecasting method based on cross-modal alignment. By constructing a cross-modal feature alignment mechanism between time-series features and the semantic space of a pre-trained language model, the proposed method enables time-series data to make full use of the structured representational capability embedded in LLM, thereby enhancing the model's ability to capture complex temporal patterns. Comparative experiments on water supply networks flow datasets against a variety of classical time-series forecasting models demonstrate that the proposed method achieves the best performance in the long-term forecasting task and maintains strong forecasting accuracy in the short-term forecasting task. Ablation experiments further verify the effectiveness of key components, including the cross-modal feature alignment module, the LLM encoding module, and the LoRA fine-tuning mechanism. Visualization analysis of the forecasting results confirms that the proposed method exhibits favorable stability and accuracy in both trend fitting and fluctuation capture. Furthermore, the proposed framework is not limited to water supply networks and can be extended to other urban infrastructure forecasting tasks, such as smart grids, transportation networks, and energy systems. This highlights its potential for broader applications in urban cyber-physical systems. In future work, more external influencing factors can be incorporated for multimodal modeling, so as to further improve the model's ability to predict dynamic changes in complex water supply systems.

Data availability statement

The data that support the findings of this study are not publicly available due to confidentiality agreements and security restrictions associated with real-world water distribution network operational data, but are available from the corresponding author upon reasonable request.

Declaration of generative AI and AI-assisted technologies

During the preparation of this manuscript, the authors used generative ChatGPT only to improve language and readability. The authors take full responsibility for the content of the manuscript.

Acknowledgments

This work was supported in part by National Key R&D Program of China (NO.2024YFB3311600), National Natural Science Foundation of China (52375089), Hefei Natural Science Foundation Project (HZR2451), the Anhui Province Key Project (202304a05020059), Open Foundation of State Key Laboratory of High-end Compressor and System Technology (SKL-YSJ202412).

Authors' contribution

Formal analysis, investigation, data curation and writing—review and editing, Tao Yang; conceptualization, methodology, software, validation and writing—original draft preparation, Hanqi Gui; resources,

visualization and writing—review and editing, Juan Xu; supervision, Xiaochuan Li; project administration, Xu Ding; funding acquisition, Yongbin Liu. All authors have read and agreed to the published version of the manuscript.

Conflicts of 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.

References

[1] Taiwo R, Yussif AM, Zayed T. Making waves: generative artificial intelligence in water distribution networks: opportunities and challenges. *Water Res. X* 2025, 28:100316.

[2] Chen R, Wang Q, Javanmardi A. A review of the application of machine learning for pipeline integrity predictive analysis in water distribution networks. *Arch. Comput. Methods Eng.* 2025, 32(6):3821–3849.

[3] Kim J, Kim H, Kim H, Lee D, Yoon S. A comprehensive survey of deep learning for time series forecasting: architectural diversity and open challenges. *Artif. Intell. Rev.* 2025, 58:216.

[4] Liang Y, Wen H, Nie Y, Jiang Y, Jin M, *et al.* Foundation models for time series analysis: a tutorial and survey. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Barcelona, Spain, August 25–29, 2024, pp. 6555–6565.

[5] Song X, Deng L, Wang H, Zhang Y, He Y, *et al.* Deep learning-based time series forecasting. *Artif. Intell. Rev.* 2025, 58(1):23.

[6] Tang H, Zhang C, Jin M, Yu Q, Wang Z, *et al.* Time series forecasting with LLMs: understanding and enhancing model capabilities. *ACM SIGKDD Explor. Newsl.* 2025, 26(2):109–118.

[7] Wang C, Qi Q, Wang J, Sun H, Zhuang Z, *et al.* ChatTime: a unified multimodal time series foundation model bridging numerical and textual data. *Proc. AAAI Conf. Artif. Intell.* 2025, 39(12):12694–12702.

[8] Kavya M, Mathew A, Shekar PR, Sarwesh P. Short term water demand forecast modelling using artificial intelligence for smart water management. *Sustain. Cities Soc.* 2023, 95:104610.

[9] Fleming SW, Rittger K, Oaida Taglialatela CM, Graczyk I. Leveraging next-generation satellite remote sensing-based snow data to improve seasonal water supply predictions in a practical machine learning-driven river forecast system. *Water Resour. Res.* 2024, 60(4):e2023WR035785.

[10] Maussner C, Oberascher M, Autengruber A, Kahl A, Sitzenfrei R. Explainable artificial intelligence for reliable water demand forecasting to increase trust in predictions. *Water Res.* 2025, 268:122779.

[11] Hyndman RJ, Koehler AB, Ord JK, Snyder RD. Linear innovations state space models with random seed states. In *Forecasting with Exponential Smoothing: The State Space Approach*. Berlin: Springer, 2008. pp. 179–208.

[12] Box GEP, Jenkins GM, Reinsel GC, Ljung GM. *Time Series Analysis: Forecasting and Control*, 5th ed. Hoboken: John Wiley & Sons, 2015.

[13] Li Y, Anastasiu DC. MC-ANN: a mixture clustering-based attention neural network for time series forecasting. *IEEE Trans. Pattern Anal. Mach. Intell.* 2025, 47(8):6888–6899.

[14] Pourebrahim S, Seifi A, Ehteram M, Hadipour M, Chen JE. The CEEMDAN-EWT-CNN-GRU-SVM model: a robust framework for decomposing non-stationary time series, extracting data features, and predicting solar radiation. *Results Eng.* 2025, 25:104267.

[15] Yue H, Wang Y, Zhang L, Yang T. A machine learning-based water supply forecasting model to quantify the impact of snow water equivalent on seasonal streamflow variability over the western U.S. *J. Hydrol.* 2025, 660:133465.

[16] LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature* 2015, 521:436–444.

[17] Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput.* 1997, 9(8):1735–1780.

[18] Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, *et al.* Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, Doha, Qatar, October 25–29, 2014, pp. 1724–1734.

[19] Muñoz-Rodríguez D, González-Ortega MJ, Aguilera-Ureña MJ, Ortega-Ballesteros A, Perea-Moreno AJ. Innovation ARIMA models application to predict pressure variations in water supply networks with open-loop control: case study in Noja (Cantabria, Spain). *Energy Nexus* 2025, 18:100423.

[20] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, *et al.* Attention is all you need. In *Proceedings of the Advances in Neural Information Processing Systems 30*, Long Beach, USA, December 4–9, 2017, pp. 5998–6008.

[21] Zhou H, Zhang S, Peng J, Zhang S, Li J, *et al.* Informer: beyond efficient transformer for long sequence time-series forecasting. *Proc. AAAI Conf. Artif. Intell.* 2021, 35(12):11106–11115.

[22] Wu H, Xu J, Wang J, Long M. Autoformer: decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems 34*, Online, December 6–14 2021, pp. 22419–22430.

[23] Zhou T, Ma Z, Wen Q, Wang X, Sun L, *et al.* FEDformer: frequency enhanced decomposed transformer for long-term series forecasting. In *Proceedings of the 39th International Conference on Machine Learning*, Baltimore, USA, July 17–23, 2022, pp. 27268–27286.

[24] Nie Y, Nguyen NH, Sinthong P, Kalagnanam J. A time series is worth 64 words: long-term forecasting with transformers. *arXiv* 2022, arXiv:2211.14730.

[25] Wu H, Hu T, Liu Y, Zhou H, Wang J, *et al.* TimesNet: temporal 2D-variation modeling for general time series analysis. *arXiv* 2022, arXiv:2210.02186.

[26] Xue H, Salim FD. PromptCast: a new prompt-based learning paradigm for time series forecasting. *IEEE Trans. Knowl. Data Eng.* 2024, 36(11):6851–6864.

[27] Gruver N, Finzi M, Qiu S, Wilson AG. Large language models are zero-shot time series forecasters. *arXiv* 2023, arXiv:2310.07820.

[28] Jin M, Wang S, Ma L, Chu Z, Zhang J, *et al.* Time-LLM: time series forecasting by reprogramming large language models. *arXiv* 2023, arXiv:2310.01728.

[29] Garza A, Challu C, Mergenthaler-Canseco M. TimeGPT-1. *arXiv* 2023, arXiv:2310.03589.

[30] Ansari AF, Stella L, Turkmen C, Zhang X, Mercado P, *et al.* Chronos: learning the language of time series. *arXiv* 2024, arXiv:2403.07815.

[31] Das A, Kong W, Sen R, Zhou Y. A decoder-only foundation model for time-series forecasting. *arXiv* 2023, arXiv:2310.10688.

[32] Shi X, Wang S, Nie Y, Li D, Ye Z, *et al.* Time-MoE: billion-scale time series foundation models with mixture of experts. In *International Conference on Learning Representations*, Singapore, April 24–28, 2025, pp. 34635–34667.

[33] Qiao Z, Liu C, Zhang Y, Jin M, Pham Q, *et al.* Multi-scale finetuning for encoder-based time series foundation models. *Adv. Neural Inf. Process. Syst.* 2026, 38:22313–22345.

[34] Abdullahi S, Danyaro KU, Zakari A, Aziz IA, Zawawi NAWA, *et al.* Time-series large language models: a systematic review of state-of-the-art. *IEEE Access* 2025, 13:30235–30261.

[35] Liu Y, Qin G, Huang X, Wang J, Long M. AutoTimes: autoregressive time series forecasters via large language models. In *Advances in Neural Information Processing Systems 37*, Vancouver, Canada, December 10–15, 2024, pp. 122154–122184.

[36] Goswami M, Szafer K, Choudhry A, Cai Y, Li S, *et al.* MOMENT: a family of open time-series foundation models. *arXiv* 2024, arXiv:2402.03885.