Electronic Research Archive

ERA, 33(2): 697–724.
DOI: 10.3934/era.2025032
Received: 22 September 2024
Revised: 18 November 2024
Accepted: 10 December 2024
Published: 11 February 2025

https://www.aimspress.com/journal/ERA

Research article

An Informer-based multi-scale model that fuses memory factors and wavelet denoising for tidal prediction

Peng Lu[1,*], Yuchen He[1], Wenhui Li[2], Yuze Chen[1], Ru Kong[3] and Teng Wang[3]

[1] College of Information Technology, Shanghai Ocean University, Shanghai 201306, China
[2] Modern Educational Technology Center, Shanghai Maritime University, Shanghai 201306, China
[3] Shandong Provincial Institute of Land Space Data and Remote Sensing Technology, Shandong Ocean Bureau, Jinan 250002, China

* **Correspondence:** Email: plu@shou.edu.cn.

Abstract: Tidal time series are affected by a combination of astronomical, geological, meteorological, and anthropogenic factors, revealing non-stationary and multi-period features. The statistical features of non-stationary data vary over time, making it challenging for typical time series forecasting models to capture their dynamism. To solve this challenge, we designed memory factors, leveraging the fusion of statistical data at the channel dimension to enhance the model's prediction capacity for non-stationary data. On the other hand, traditional approaches have limitations in trend and cycle decomposition, making it difficult to detect complicated multi-period patterns and accurately separate the components. We combined integrated frequency domain optimization and multi-level, multi-scale convolutional kernel technologies. By employing Fourier-based methods and iterative recursive decomposition strategies, we effectively separated periodic and trend components. Then, the periodic multi-level wavelet block was applied to extract the periodic interaction features, aiming to deeply mine the latent information of periodic components and enhance the model's long-term prediction capabilities. In this paper, we used the Informer model as the foundational framework for further research and development. In comparative experiments, our proposed model outperformed LSTM, Informer, and MICN by 61.4%, 51.7%, and 23.8%, respectively. In multi-time-span prediction, the model's error remained stable as the prediction span increased from 48 to 96 steps (from 0.059 to 0.067). Under multi-site conditions, the model achieved varying degrees of improvement over the baseline in three key evaluation metrics, with average increases of 35.2%, 35.6%, and 61.2%, respectively. In this study, we focused on the extraction of short-period features from tidal data, providing an innovative and reliable solution for tidal height prediction. The results are significant for tidal assessments and protective engineering construction.

Keywords: tidal prediction; time series; Informer; memory factor; multi-scale convolution; multi-level wavelet

1. Introduction

Tides are periodic seawater fluctuations formed under the gravitational influences of the Moon and the Sun. Related tidal risks emerge from changes in tidal and marine meteorological conditions, commonly including storm surges, tsunamis, and sea level rise [1]. These hazards can lead to coastal erosion, land subsidence, and saltwater incursion, posing serious threats to life and property in coastal areas [2]. Monitoring and anticipating tide level fluctuations at tidal stations, as well as timely transmitting the data to relevant authorities and citizens, can enable early preparation for prevention and evacuation. In addition, real-time and accurate tide level forecasts have practical implications for coastal engineering operations, ecological preservation, and renewable energy development.

Physical-based methods, such as the tidal dynamics theory proposed by P. S. Laplace, and the systematic study of tidal phenomena through harmonic analysis by G. H. Darwin, provide a high degree of interpretability for tidal prediction by establishing mathematical models directly related to tidal changes. However, these methods often require extensive physical data inputs and complex model constructions, resulting in poor adaptability to environmental changes. In addition, physical approaches demonstrate considerable limitations when dealing with nonlinear and high-dimensional data. Statistical methods, such as the least squares method [3] and the Kalman filtering approach [4], optimize physical models by analyzing historical tidal data to enhance the accuracy of model parameters. Sequential data assimilation techniques based on the Monte Carlo method [5] can flexibly adapt to nonlinear ocean data. However, statistical methods frequently rely on past patterns and struggle to capture deep nonlinear relationships within the data.

Signal processing techniques effectively remove noise from tidal data through the application of filtering, transformation, and denoising methods. The multi-scale decomposition method is widely recognized for analyzing the frequency characteristics of signals, effectively extracting the irregular oscillatory components of signals, including time-varying amplitude and phase. For instance, Yang et al. [6] proposed a short-term load forecasting method based on multi-scale deep neural networks, which validated the effectiveness of multi-scale decomposition in processing complex signals by decomposing load sequences into high-frequency and low-frequency components. Zhang et al. [7] combined variational mode decomposition (VMD) with long short-term memory networks (LSTM) to address the issue of prediction delay in the LSTM model for wave height forecasting. Furthermore, Yin et al. [8] coupled the discrete wavelet transform (DWT) with variable structure neural network techniques to accomplish real-time predictions at selected US tidal stations. Such methods not only effectively handle the decomposition of time series data from high to low frequencies, but also greatly reduce the data complexity.

Traditional artificial neural networks [9] often struggle to model complex problems due to their shallow structure, and their usefulness, especially in long-term prediction, has yet to be thoroughly explored [10]. With the development of artificial neural networks, deep learning models, such as RNN [11], LSTM [12], and Bi-LSTM [13], utilize multi-layered structures to extract more abstract and high-level features. These models can capture complex patterns and relationships in massive data. However, the training and optimization processes of deep learning models are more complex and prone

to overfitting difficulties.

Fusion models, which utilize the complementarity of different models to enhance predictive performance, are favored by researchers. Luo et al. [14] found that using a single neural network model to forecast wave heights has limitations. They combined bi-directional LSTM with attentional mechanisms, achieving stable forecasting performance and accurately forecasting wave heights in the Atlantic storm areas within 12 hours. Oh and Suh [15] combined EOF analysis, DWT, and neural networks, which exhibited stronger nonlinear capabilities than the ANN model. Aly [16] experimented with various combinations and sequencing of wavelet networks (WNN), artificial neural networks (ANN), least squares-based Fourier series (FS), and recursive Kalman filters (RKF), determining the optimal model for forecasting tidal components. The challenge of fusion models lies in achieving proper integration and optimization among different algorithms. Currently, in the field of artificial intelligence, large-scale deep learning models based on the Transformer [17] have become the focus of research and application. These models excel in capturing long-term dependencies, handling high-dimensional data, and combating noise. When integrated with other techniques, such models [18–20] can be applied to forecasting in various scenarios. Traditional Transformer-based models have limitations in capturing complex periodic patterns, especially when multiple periodic components overlap in the data. Moreover, the non-stationary characteristics of tidal data may make it difficult for the model to learn stable patterns, which affects the prediction accuracy. Furthermore, models may experience a significant decline in performance when making long-term predictions, which fails to meet the requirements of practical applications. Therefore, we propose a Transformer architecture model that can effectively model the multi-periodicity and non-stationarity characteristics of tidal data, addressing the research gap.

We aim to utilize an Informer-based architecture, integrating intrinsic statistical features, and combining multi-scale and multi-period feature extraction and fusion methods to improve tidal data prediction and analysis. The main contributions are as follows:

- By utilizing Informer—which retains the excellent adaptability of the Transformer in dealing with large-scale, multi-scale, and nonlinear data—and leveraging its unique advantage of reducing high memory usage, we efficiently forecast tidal height sequences in one step.
- Given the non-stationary properties of tidal data, a statistical feature fusion mechanism based on 2D convolution and channel self-attention is proposed. This design integrates the stability of data normalization and the flexibility of integrated modelling with multiple statistical features, enabling the model to sensitively capture key events in time series through the attention mechanism.
- Employing statistical methods in time series analysis, we have integrated the multi-periodic characteristics triggered by astronomical tides and multi-scale analysis. Successfully combining the integrated frequency domain optimization technique with the multi-scale multi-level convolution kernel technique through cascade processes, we enhance the decomposition precision of each component under multi-factorial influences.
- A periodic-based dynamic adaptive architecture is proposed, employing segmentation, combination, multi-level wavelet decomposition, interaction, and reconstruction to mine fine-grained features within periodic components, addressing the issue of significant declines in long-term prediction performance.

The paper follows this structure: In Section 2, we discuss the related work. In Section 3, we describe the model architecture and components of the modules. In Section 4, we cover the data sources and experimental setup. In Section 5, we present the model evaluation and experimental analysis. In

Section 6, we summarize the paper and outline the scope and potential avenues for further exploration.

2. Related work

In this section, we list other content related to our work on tidal prediction. This includes recent developments in Transformers, Informer's core prediction algorithm, strategies for recovering non-stationary characteristics of sequences, and wavelet decomposition.

2.1. Recent progress of Transformers

Transformer-based networks have become a dominant force in natural language processing. Vaswani et al. [17] introduced Transformer, which is based on a self-attention mechanism structure, replacing the traditional recurrent neural networks (RNNs) and convolutional neural networks (CNNs), displaying a high potential in expressing long-distance dependencies. Zhou et al. [21] invented Informer, which tackles the high memory usage and inherent limitations of the encoder-decoder architecture in Transformer. Autoformer [22] introduced an auto-correlation mechanism based on the progressive decomposition architecture to replace the self-attention mechanism. This mechanism identifies the similarity of subsequences based on the sequence's periodicity, reducing time complexity while breaking the information usage bottleneck. FEDformer [23], based on Autoformer, proposed a frequency-enhanced Transformer architecture, which employs Fourier enhanced blocks and wavelet enhanced blocks. This allows for the capture of important information through frequency-domain mapping, achieving linear computational complexity and memory overhead by randomly selecting a fixed number of Fourier components. The same year, Nie et al. [24] proposed PatchTST, which segments the time series into various time periods and uses the self-attention mechanism to model them, maintaining the locality of the time series. A general framework named Non-stationary Transformers [25] was introduced to solve the issue of feature loss while maintaining data stationarity. iTransformer [26] alters the roles of the attention mechanism and feedforward network while preserving the architecture of Transformer for better temporal representation. In summary, the development of Transformer and its variants in the field of time series forecasting has demonstrated powerful predictive performance and broad application prospects.

2.2. Informer

Informer [21] is a high-performance prediction algorithm based on Transformer [17] and follows an encoder-decoder structure. This algorithm feeds sequences into the encoder for processing, then reduces the temporal complexity through a ProbSparse self-attention mechanism. Subsequently, a self-attention distilling operation effectively reduces the temporal dimension of the input sequence. Finally, the output is generated by the decoder. Informer's design provides high scalability in handling large-scale and complex time series data, making it a reasonable choice as the base model.

2.2.1. ProbSparse self-attention

Traditional self-attention requires $0(L_Q L_K)$ memory and the cost of quadratic dot product computation, which are the main drawbacks limiting its prediction ability. Research has revealed the

long tail distribution in the self-attention feature map, where only a few dot product pairs contribute to primary attention, whereas others can be ignored. To measure the importance of each key under the given query, an attention probability distribution $\mathrm{p}(k_j|q_i)$ [27] and a uniform distribution $\mathrm{q}(k_j|q_i)$ have been introduced:

$$\mathrm{p}(k_j|q_i) = \frac{k(q_i,k_j)}{\sum_l k(q_i,k_l)} \tag{1}$$

$$\mathrm{q}(k_j|q_i) = \frac{1}{L_K} \tag{2}$$

where q_i represents the query vector, k_j represents the key vector, and L_K is the total number of keys. $k(q_i,k_j)$ is an asymmetric exponential kernel function $\exp(\frac{q_i k_j^{\mathrm{T}}}{\sqrt{d}})$.

Furthermore, the direct correlation between two distributions is assessed using the discrete Kullback-Leibler divergence formula:

$$\mathrm{D_{KL}}(\mathrm{P}||\mathrm{Q}) = \sum_{i\in X} \mathrm{P}(i) * \left[\log\left(\frac{\mathrm{P}(i)}{\mathrm{Q}(i)}\right)\right] \tag{3}$$

where $\mathrm{Q}(i)$ and $\mathrm{P}(i)$ represent the probabilities of the probability distributions Q and P at the *i*-th event, respectively. This formula is used to measure the expected information loss when one probability distribution approximates another. Each query needs to assess its sparsity; thus, it is evaluated by calculating the Kullback-Leibler divergence between the two distributions as mentioned above, substituting Eqs (1) and (2) into Eq (3):

$$\begin{aligned}\mathrm{D_{KL}}(\mathrm{q} \parallel \mathrm{p}) &= \sum_{j=1}^{L_K}\left(\frac{1}{L_K}\cdot\ln\frac{1}{L_K} - \frac{1}{L_K}\cdot\ln\frac{k(q_i,k_j)}{\sum_l k(q_i,k_l)}\right)\\ &= -\ln L_K - \sum_{j=1}^{L_K}\frac{1}{L_K}\cdot\left(\ln\ e^{\frac{q_i k_j^{\mathrm{T}}}{\sqrt{d}}} - \ln\Sigma_l e^{\frac{q_i k_l^{\mathrm{T}}}{\sqrt{d}}}\right)\\ &= -\ln L_K - \frac{1}{L_K}\sum_{j=1}^{L_K}\frac{q_i k_j^{\mathrm{T}}}{\sqrt{d}} + \ln\sum_{l=1}^{L_K} e^{\frac{q_i k_l^{\mathrm{T}}}{\sqrt{d}}}\end{aligned} \tag{4}$$

After discarding the constant terms, the sparsity measure of the *i*-th query vector is defined as:

$$\begin{aligned}\mathrm{M}(q_i,K) &= \ln\sum_{j=1}^{L_K} e^{\frac{q_i k_j^T}{\sqrt{d}}} - \frac{1}{L_K}\sum_{j=1}^{L_K}\frac{q_i k_j^T}{\sqrt{d}}\\ &\le \ln\left(L_K\cdot\max_j\{\frac{q_i k_j^{\mathrm{T}}}{\sqrt{d}}\}\right) - \frac{1}{L_K}\sum_{j=1}^{L_K}\left(\frac{q_i k_j^{\mathrm{T}}}{\sqrt{d}}\right)\\ &= \ln L_K + \max_j\{\frac{q_i k_j^{\mathrm{T}}}{\sqrt{d}}\} - \frac{1}{L_K}\sum_{j=1}^{L_K}\left(\frac{q_i k_j^{\mathrm{T}}}{\sqrt{d}}\right)\end{aligned} \tag{5}$$

Based on the sparsity measure results, a random selection of $L_Q \ln L_K$ queries is made to participate in the dot product calculation of the attention mechanism. This reduces the complexity from $O(L^2)$ to $O(L\ln L)$. The selected queries are considered to be relatively far from the uniform distribution.

2.2.2. Encoders

Due to the presence of numerous redundant vectors in sequences processed by ProbSparse self-attention mechanisms, it is necessary to employ the self-attention distilling operation to selectively

refine the most informative parts of the input data. This effectively overcomes the drawbacks of the Transformer, such as the bottlenecks of memory usage and stacked layers, while reducing network parameters without losing crucial information. A series of operations based on one-dimensional convolution and max pooling is called the distilling operation. The process from j-th layer to j+1-th layer can be summarized as follows:

$$X_{j+1}^{t} = \mathrm{MaxPool}\left(\mathrm{ELU}\left(\mathrm{Conv1d}\left(\left[X_{j}^{t}\right]\right)\right)\right) \tag{6}$$

where $[\cdot]$ represents the ProbSparse self-attention block, Conv1d(·) uses ELU(·) as the activation function to execute regular convolution operations in the time dimension. After each convolution layer, a max pooling layer is added to downsample to half of its length, reducing memory utilization to $O\left((2-\lambda)L\log L\right)$.

2.2.3. Decoders

Unlike the traditional Transformer, which predicts outputs step-by-step, the generative decoder of the Informer can predict long sequence outputs in a single forward propagation rather than a step-by-step way, drastically boosting the prediction speed and lowering the cumulative error. The input to the decoder at time t is a concatenation of the following two parts:

$$X_{\mathrm{feed_{de}}}^{t} = \mathrm{Concat}\left(X_{\mathrm{token}}^{t}, X_{0}^{t}\right) \in \mathbb{R}^{\left(L_{\mathrm{token}}+L_{y}\right)\times d_{\mathrm{model}}} \tag{7}$$

where $X_{\mathrm{feed_{de}}}^{t}$ represents the input to the decoder, X_{token}^{t} is the start token of the sequence. X_{0}^{t} denotes the placeholder for the target sequence, and zero padding is used to maintain the consistency of the input dimensions. Then, the masked multi-head self-attention is employed to focus each position solely on the information preceding it, thus preserving autoregressive features, preventing future information leakage, and enhancing generalization capabilities.

2.3. Recovering the non-stationary properties of the sequence

To address the critical information related to the original data dimensions that may be removed by over-standardization, it is necessary to reintroduce important statistical information from the original sequence at key parts of the model. In this way, the model can effectively utilize and restore the inherent temporal dependencies of the original sequence [25]. It is noted that the normalization process handles Q, K as follows:

$$Q' = \frac{\left(Q-1\mu_{Q}^{\top}\right)}{\sigma_{x}}, K' = \frac{\left(K-1\mu_{K}^{\top}\right)}{\sigma_{x}} \tag{8}$$

where $\mu_{Q}, \mu_{K} \in \mathbb{R}^{d_k\times 1}$ respectively correspond to the means of $Q, K \in \mathbb{R}^{S\times d_k}$, and $\sigma_{x} \in \mathbb{R}^{1\times d_k}$ is the standard deviation of the original sequence. The normalized Q', K' are then incorporated into the attention mechanism:

$$\mathrm{Softmax}\left(\frac{QK^{\top}}{\sqrt{d_k}}\right) = \mathrm{Softmax}\left(\frac{\sigma_{x}^{2}Q'K'^{\top}+1\left(\mu_{Q}^{\top}K^{\top}\right)+\left(Q\mu_{K}\right)1^{\top}-1\left(\mu_{Q}^{\top}\mu_{K}\right)1^{\top}}{\sqrt{d_k}}\right) \tag{9}$$

where $1 \in \mathbb{R}^{S\times 1}$ is an all-ones vector, based on the translational invariance of the Softmax operator,

Eq (9) simplifies to:

$$\text{Softmax}(\frac{QK^{\top}}{\sqrt{d_k}}) = \text{Softmax}\left(\frac{\sigma_x^2 Q' K'^{\top} + 1\mu_Q^{\top} K^{\top}}{\sqrt{d_k}}\right) \tag{10}$$

Computational results confirm that after the attention-based Softmax normalization, there are discrepancies between the original sequence and the normalized sequence. Therefore, subsequent efforts will need to focus on reconstructive modeling of these discrepancies at corresponding positions.

2.4. Wavelet decomposition

Wavelet decomposition [28] is a mathematical method for decomposing signals into multiple frequency components, enabling analysis in both time and frequency domains. In selecting wavelet bases, Legendre polynomials [29] and Chebyshev polynomials [30] are widely used in various signal and image processing scenarios. The common advantages of these two polynomials include their orthogonality, stable numerical properties, and high analytical efficiency. Generalized filter matrices *H* and *G* can be constructed and are defined as follows:

$$\begin{cases} H(k_i, k_j, n) = \frac{1}{\sqrt{2}} \sum_m w_m \, \phi_{k_i}\left(\frac{x_m + n}{2}\right) \phi_{k_j}(x_m) \\ G(k_i, k_j, \text{psi1}, \text{psi2}, n) = \frac{1}{\sqrt{2}} \sum_m w_m \, \psi\left(\text{psi1}, \text{psi2}, k_i, \frac{x_m + n}{2}\right) \phi_{k_j}(x_m) \end{cases} \tag{11}$$

Here, n represents the time offset, taking values of 0 or 1, used to extract information from odd and even positions in the signal. Two sets of filters are defined based on the values: Low-pass filters (H_0, H_1) and high-pass filters (G_0, G_1). The indices k_i and k_j in the filter matrix represent different frequency components involved in the wavelet transform. The scale function ϕ captures the low-frequency part of the signal. The wavelet function ψ derives two auxiliary base functions, psi1 and psi2, which capture the high-frequency part of the signal in odd and even segments, respectively. By transforming and translating basis functions across different scales and combining weight factors w_m and polynomial roots x_m to construct the filter matrix, the multi-scale decomposition of the signal [23] is achieved by matrix multiplication with signals at different frequency levels, accurately capturing the signal's low-frequency trends and high-frequency fluctuations.

3. Modeling

We propose a multi-scale Informer framework that fuses memory factors and wavelet denoising, as shown in Figure 1. First, memory factors α and β, which fully consider the statistical characteristics of the data, are introduced. They are strategically placed in the attention mechanisms of the encoder and decoder to restore the model's ability to capture dynamic changes in time series. Details will be discussed in Section 3.1. Second, the encoder employs a cascade structure, alternating between multi-head ProbSparse self-attention mechanisms and distilling operations. The multi-level processed encoder output and the embedded decoder input are each passed through a cascading process using Fourier-based frequency domain optimization and the multi-scale, multi-level convolutional technique (as shown in the dashed box in Figure 1). This approach successfully decomposes periodic and trend components, overcoming the limitations of traditional methods for decomposing various tidal dataset components. Detailed discussions on this part will be provided in Section 3.2.

Building on the precise decomposition of the original time series, we extend our methodology to the extraction of detailed periodic interaction features, applying the periodic multi-level wavelet block to process the periodic components. Section 3.3 will elaborate on this content. Moreover, multi-head ProbSparse cross-attention is applied to the decomposed trend components to extract trend interaction features, enhancing the prediction accuracy of tidal level change trends. Finally, the interacted periodic and trend components are summed and normalized, and a multi-step forecast result is mapped through a fully connected layer.

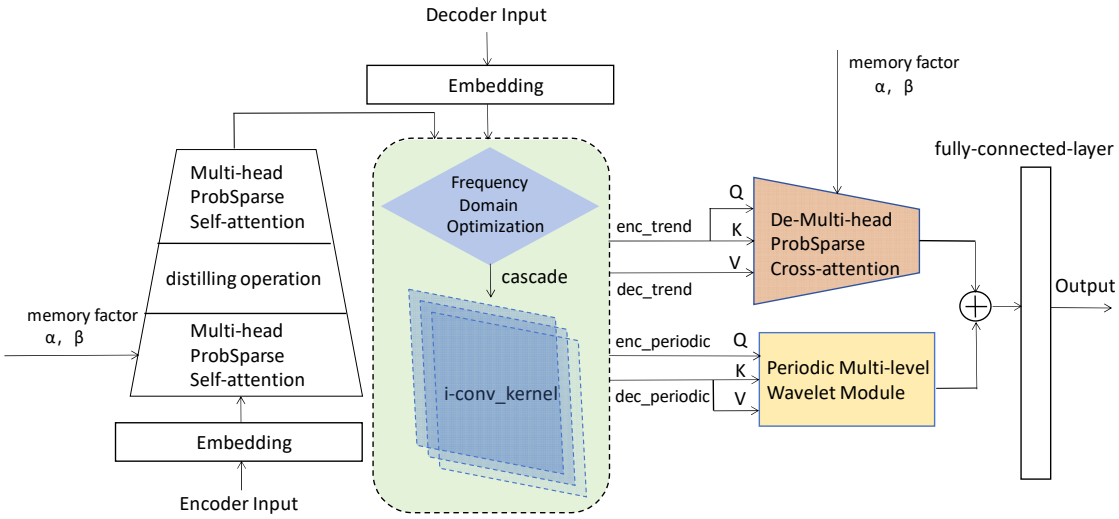

Figure 1. Overall framework diagram of the model.

3.1. Memory factors

In addition to astronomical tides, tidal heights may be influenced by multiple factors, resulting in non-stationary characteristics. During data training, differences in statistical characteristics (such as mean and variance) among different batches of data may make it difficult to capture dynamic changes. Although models based on differencing [31] and standardization [32] operations can partially address these issues and enhance the predictability of sequences, these methods often fail to consider breakpoints, leading to the over-stationarization problem [25].

In this section, we provide a detailed description of the framework of the memory factors (see Figure 2). The framework is centered on data normalization and the feature cross-fusion mechanism (see dashed box in Figure 2) based on dynamic channel self-attention [33] and 2D convolution. We will focus on two main aspects: (1) feature fusion; (2) construction of memory factors and restoration of non-stationary characteristics.

3.1.1. Feature fusion

When deep learning models process time series data, one of the primary challenges is how to effectively integrate statistical information from different dimensions to enhance the accuracy of predictions. For instance, models need not only to understand the current values of data points, but also to grasp the statistical properties of surrounding points. These properties often contain key

information about data volatility and stability. Traditional processing methods tend to handle the statistical properties separately [34] or ignore data volatility [35], thus failing to fully utilize the potential correlations among them.

To address this issue, we have designed an improved feature fusion mechanism (see dashed box in Figure 2). First, Z-score normalization is applied to each input sequence $X = [x_1, x_2, \ldots, x_S]^\top \in \mathbb{R}^{S\times C}$, performing translation and scaling transformations to obtain $\overline{X} = [\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_S]^\top \in \mathbb{R}^{S\times C}$, where S and C respectively represent the sequence length and the number of variables. The Z-score normalization process involves calculating the mean and variance of the data:

$$\mu_x = \frac{1}{S}\sum_{i=1}^{S} x_i\,,\ \sigma_x^2 = \frac{1}{S}\sum_{i=1}^{S}(x_i-\mu_x)^2\,,\ \bar{x}_i = \frac{1}{\sigma_x}\odot(x_i-\mu_x) \tag{12}$$

Here, $\mu_x, \sigma_x^2 \in \mathbb{R}^{C\times 1}$ represent the mean and standard deviation, respectively, and $\odot$ denotes element-wise multiplication. By subtracting μ_x and then dividing by σ_x for each data point in the original sequence, the processed data conforms to a standard normal distribution with mean 0 and variance 1. This standardization eliminates differences in data magnitudes, ensuring numerical stability and comparability between datasets.

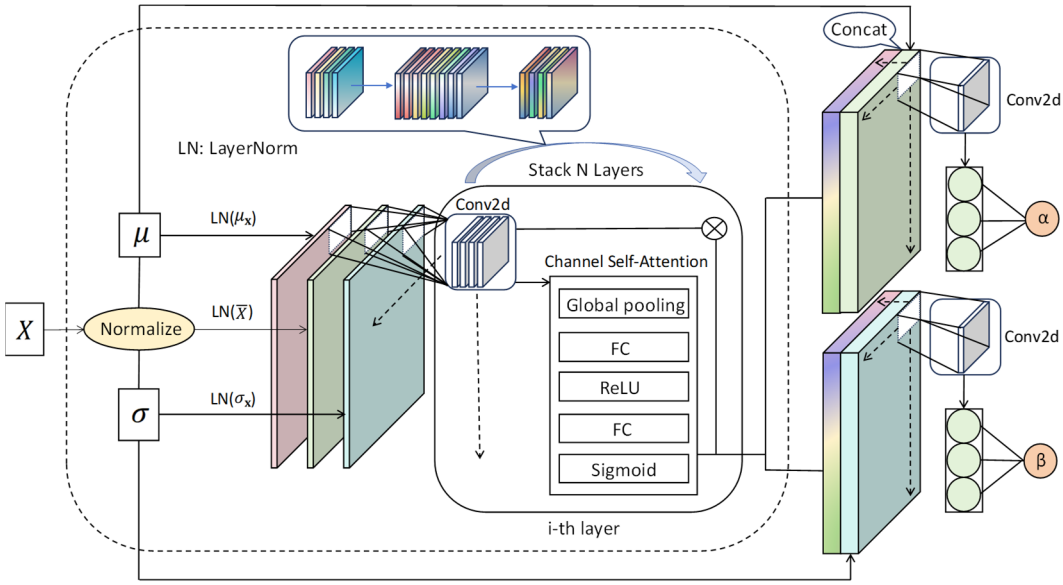

Figure 2. Construction of memory factors.

$\overline{X}$ represents the standardized raw data, μ_x describes the central tendency of the data, and σ_x measures the dispersion of data around the mean. After individually applying layer normalization to these three variables, they are concatenated along the channel dimension to obtain the variable X''. This procedure can be expressed with the following mathematical formula:

$$X''=\text{Stack}(\text{LayerNorm}(\overline{X}),\text{LayerNorm}(\mu_x),\text{LayerNorm}(\sigma_x),\text{dim}=1) \tag{13}$$

Next, we use a multi-layer 2D convolutional network to process the concatenated three-channel feature data. By progressively increasing or decreasing dimensions, we effectively integrate relationships between channel features at different levels. Applying 2D convolution to increase the number of channels enables the network to capture more detailed and local features, whereas reducing

the number of channels encourages the network to learn more abstract and global feature representations.

To further enhance the ability to distinguish feature representations at different channels, we introduce a channel-wise self-attention layer following each 2D convolutional layer. Specifically, we employ global average pooling (GAP) to capture the global contextual information of each channel. Subsequently, a gating network composed of fully connected layers, ReLU and Sigmoid activations dynamically adjusts the weights of each channel, thereby enhancing the model's sensitivity to features at crucial channels. The implementation of the dynamic channel attention layer can be described by Eqs (14) and (15):

$$\mathrm{GAP}(X_c'') = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X_{cij}'' \tag{14}$$

Here, H and W represent the height and width of the feature data, respectively, X_{cij} is the element at position (i, j) on channel c, and $\mathrm{GAP}(X_c'')$ denotes the average value of channel c.

$$\text{Channel Self-Attention}(X'') = \sigma(W_2 \mathrm{ReLU}(W_1 \mathrm{GAP}(X'') + b_1) + b_2) \odot X'' \tag{15}$$

where W_1, W_2, b_1, b_2 denote learnable parameter matrices, respectively, and □ represents element-wise multiplication, i.e., attention weighting.

3.1.2. Constructing memory factors and recovering non-stationary characteristics

Sequence normalization improves the statistical stability of data by adjusting its mean and variance. However, this may limit the model's ability to capture the inherent temporal dependencies of the raw sequence.

Therefore, we denote the fused features obtained after alternating processing through multiple 2D convolutional layers and channel self-attention layers in Section 3.1.1 as *fused_feature*, which are then concatenated (cat) with the main mutated statistical measures, namely mean and standard deviation, respectively, along the channel dimension. Subsequently, the concatenated features undergo further processing through a dimension reduction convolution layer (conv), flattening operation (flatten), and fully connected layer (fc). The aim is to specifically address the intrinsic variability and overall trends of the data, with the processed results marked as memory factors α and β:

$$\begin{cases} \alpha = \mathrm{fc}\big(\mathrm{conv}\big(\mathrm{cat}(fused\_feature, mean\_expended)\big).\mathrm{flatten}()\big) \\ \beta = \exp\Big(\mathrm{fc}\big(\mathrm{conv}\big(\mathrm{cat}(fused\_feature, std\_expended)\big).\mathrm{flatten}()\big)\Big) \end{cases} \tag{16}$$

Here, *mean_expended* and *std_expended* refer to the extended mean tensor and standard deviation tensor, respectively. α is closely associated with the concept of mean, emphasizing the capture of data's stable characteristics and long-term trends, while β integrates the concept of standard deviation, focusing on capturing the dynamic changes and instantaneous fluctuations in data. Below is the method of applying memory factors in the attention mechanism based on Section 2.3 theory:

$$\mathrm{Attn}(Q', K', V', \alpha, \beta) = \mathrm{Softmax}\left(\frac{\beta Q' K'^{\top} + \alpha^{\top}}{\sqrt{d_k}}\right) V' \tag{17}$$

While retaining the advantages of data normalization, the model utilizes the attention mechanism

to reintroduce the original magnitude information carried by the memory factors, thus enhancing the prediction accuracy of complex, non-stationary sequences.

3.2. Multi-scale and multi-level convolutional decomposition blocks

In practical situations, influenced by astronomical tides, the multi-periodic properties of tides frequently emerge. However, tidal models struggle to effectively handle multiple periods and comprehensively analyze the various periodic components within tidal datasets. Furthermore, influenced by other natural environmental factors and human activities, there are various trend changes in tidal heights, exhibiting non-stationary characteristics of trends beyond periodic fluctuations.

In the context of periodic decomposition and multi-scale trend extraction, traditional methods such as classical time series decomposition techniques (e.g., additive models [36] and multiplicative models [37]) are widely used for periodic decomposition, while moving averages and exponential smoothing are commonly employed to extract multi-scale trends. However, these methods have numerous limitations when dealing with complex patterns in time series. Specifically, when the data contains multiple periodic components, the aforementioned traditional models for periodic decomposition typically employ a singular approach. The accuracy and efficiency of this approach often depend on the nature of the data being decomposed and the compatibility of the chosen method with the dataset. Similarly, the separation of trend components faces similar challenges, and the residuals from decomposition are often directly discarded [38].

To address this issue, we propose a novel time series analysis framework, as shown in Figure 3, which contains a cascading process using Fourier-based frequency domain optimization to identify periodic components (see Section 3.2.1). Subsequently, an iterative recursive decomposition strategy is employed to extract trend components. During the recursive process, multi-scale convolutional kernels are sequentially arranged, which will be elaborated in Section 3.2.2. It should be noted that the size of the single-cycle/multi-cycle pattern convolutional kernels selected during the recursive process corresponds to the periods identified in Section 3.2.1.

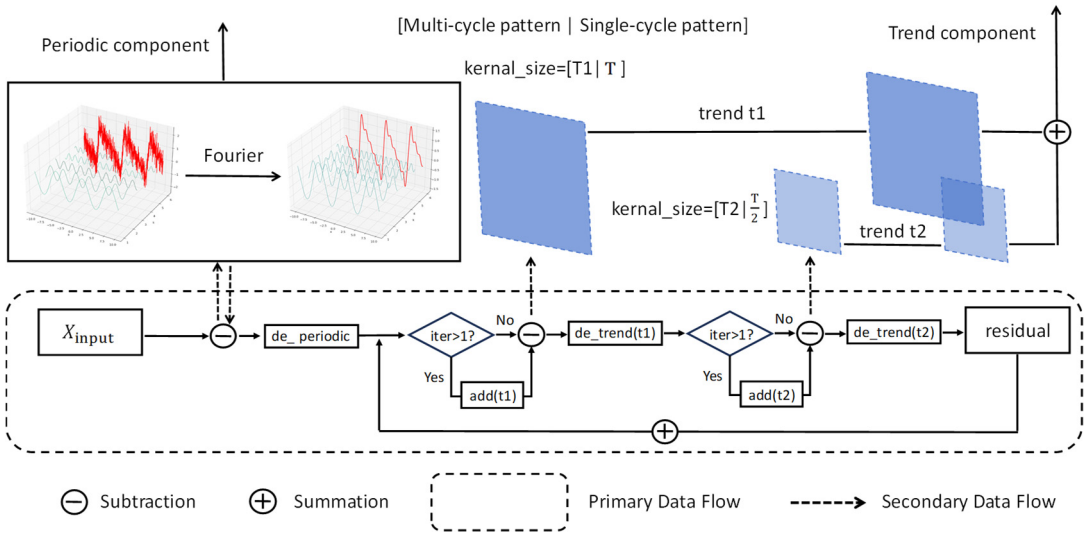

Figure 3. Framework of multi-scale multi-level convolutional decomposition block.

3.2.1. Frequency domain optimization to enhance periodic signals

In implementing frequency domain optimization, the framework initially performs a fast Fourier Transform (FFT) on the input data to extract frequency domain information. Subsequently, non-core high frequencies and extremely low frequencies are filtered out in the frequency domain, and a set of the top k frequencies with significant amplitudes are selected, denoted as $F_{\text{top-amp}}$:

$$A = \text{Amp}\left(\text{FFT}\left(X_{\text{input}}\right)\right) \tag{18}$$

$$F_{\text{top-amp}} = \arg \ \text{top-}k(A) \tag{19}$$

Here, Amp denote the calculation of amplitudes in the frequency domain. Unlike Wu et al. [39], the frequencies we select are not only concentrated at high amplitudes but also related to specific periods. The specific periods are sequentially defined using the periodogram P(f) [40], with the results cascading into the autocorrelation coefficient ρ_t for further analysis. P(f) defines a method for estimating the power spectral density at frequency f:

$$\text{P}(f) = \left|\frac{1}{N}\sum_{n=0}^{N-1} x\,(n)e^{-i2\pi fn}\right|^2 \tag{20}$$

where $x(n)$ is the sample value of a sequence signal of length N at time point n, and $e^{-i2\pi fn}$ is the complex exponential function used in the Fourier transform. By setting a threshold, significant peaks in the power spectral density are selected from P(f), preliminarily defining the set of specific frequencies. Further, the reciprocal of the frequencies with significant peak power is taken to determine their autocorrelation coefficients:

$$\rho_t = \frac{\sum_{i=1}^{N}[X_i - \text{mean}(X)][X_{i-t} - \text{mean}(X)]}{\sum_{i=1}^{N}[X_i - \text{mean}(X)]^2} \tag{21}$$

where ρ_t denotes the autocorrelation coefficient at lag t, which measures the linear correlation between a point in the sequence and another at time t delayed. Calculating the autocorrelation coefficient aims to more deeply focus on the periodic values with high self-similarity. The combination of Eqs (20) and (21) significantly enhances the credibility of the detection method for specific periods, ensuring that only the frequencies significant in both time and frequency domains are recognized and added to the frequency list, thus guaranteeing the integrity and accuracy of the periodic analysis. We summarize the frequency selection and determination of the periodic component as follows:

$$F_{\text{specific}} = \left\{\frac{1}{P_1}, \ldots, \frac{1}{P_n}\right\},\ \{f_1, \cdots, f_m\} = F_{\text{top-amp}} \cup F_{\text{specific}}\ , f_* \in \left\{1, \cdots, \left[\frac{t}{2}\right]\right\} \tag{22}$$

$$X_{\text{periodic}} = \text{IFFT}(A|_{F_{\text{top-amp}} \cup F_{\text{specific}}}) \tag{23}$$

Here, $\{P_1, \ldots, P_n\}$ represents specific periods selected using the periodogram and autocorrelation coefficients, with F_{specific} being the corresponding set of specific frequencies. Due to the conjugate symmetry in the frequency domain, f_* only focuses on the former $\left[\frac{t}{2}\right]$ frequencies. The final set of m frequencies, composed of F_{specific} and $F_{\text{top-amp}}$, not only enhances the principal periodic structure of the signal, but also facilitates the comprehensive understanding of the dynamic characteristics of

periodic signals by the model. Ultimately, these frequencies are transformed through the inverse Fourier transform to constitute the periodic component X_{periodic}.

3.2.2. Iterative multi-scale convolution to extract trend signals

During multi-scale trend extraction, the framework employs an iterative multi-scale multi-level convolution algorithm to extract trend signals. To reveal the trend changes in the data from macroscopic to microscopic levels, each iteration begins by smoothing and removing the more prominent trend parts of the data, followed by a focus on more detailed local fluctuations.

Specifically, depending on the periodic pattern of the sequence data (multi-period or single-period modes), the size of the convolution kernels is adaptively determined. In the multi-period mode, the array of convolution kernels is arranged in descending order of period values, i.e., [T1, T2, T3], where T1 > T2 > T3; in the single-period mode, the array of convolution kernels is formatted as [*T*, *T*/2, *T*/4, ...]. This strategy begins with larger convolution kernels and gradually transitions to smaller ones, sequentially extracting trends from the data to achieve trend capture from coarse to fine. Algorithm 1 details the common algorithm for both multi-period and single-period modes.

Algorithm 1. Iterative Multi-scale Multi-level Convolution Algorithm

Input: original signal $X \in \mathbb{R}^{L\times d_{\text{model}}}$, periodic signal $X_{\text{periodic}} \in \mathbb{R}^{L\times d_{\text{model}}}$

Parameter: convolution kernel at level i K_i

Output: $X_{\text{trend\_sum}} \in \mathbb{R}^{L\times d_{\text{model}}}$

1: for each iteration $iter \in \{1, \ldots, num\_iterations\}$ do

2: if *iter* = 1, initialize $X_{\text{de\_trend}}^{iter}$ with the periodic signal removed:

3: $X_{\text{de\_trend}}^{iter} = X - X_{\text{periodic}}$

4: if *iter* > 1, initialize $X_{\text{de\_trend}}^{iter}$ with the residual after the last iteration:

5: $X_{\text{de\_trend}}^{iter} = residual^{iter-1}$

6: for each convolution kernel $i \in \{1, \ldots, \text{sum}(K)\}$ do

7: if *iter* > 1, re-add trend term based on the last convolution kernel's position:

8: $X_{\text{de\_trend}}^{iter} = X_{\text{de\_trend}}^{iter} + Trend_i^{iter-1}$

9: Extract trend terms using average pooling at kernel level i:

10: $Trend_i^{iter} = \text{AvgPool}\left(\text{Padding}\left(X_{\text{de\_trend}}^{iter}\right),\ \text{kernel\_size} = K_i\right)$

11: Update detrended data:

12: $X_{\text{de\_trend}}^{iter} = X_{\text{de\_trend}}^{iter} - Trend_i^{iter}$

13: End for

14: Update the residuals after removing all level trend terms:

15: $residual^{iter} = X_{\text{de\_trend}}^{iter}$

16:End for

Return the final trend component $X_{\text{trend\_sum}} = \sum_{i=1}^{\text{sum}(K)} Trend_i^{num\_iterations}$

In each iteration, the trend extracted by the convolution kernels from the corresponding positions in the previous iteration is combined with the residual data, continuously adjusting and optimizing the precision of the extracted trend to approximate the true dynamics of the data. Finally, the trend component is obtained by summing up the $Trend_i^{num\_iterations}$ extracted by all convolution kernels.

Additionally, to simplify data processing, subsequent analyses will no longer consider the residual terms after multiple iterations.

To enhance the interpretability of the chosen convolution kernel sizes and maintain the coherence of the F_{specific} determined in Section 3.2.1 for subsequent use, we adapt the periodic values corresponding to F_{specific} into the selection of convolution kernel sizes. This approach ensures consistency of the entire data analysis process, from cycle identification to trend analysis, all based on the same theoretical foundation.

3.3. Periodic multi-level wavelet block (PMW-Block)

Inspired by several models, such as Autoformer [22] and FedFormer [23], we believe that further exploring the implicit internal information of periodic components through interactive means can significantly enhance the experimental results of long-term forecasting.

Based on an in-depth study of time-frequency analysis methods, we introduce a new perspective, namely the PMW-Block, specifically designed for analyzing time series data with distinct periodic characteristics. The complete architecture of the PMW-Block is shown in Figure 4. This architecture incorporates a periodic-based dynamic adaptive mechanism, emphasizing fine-tuning within the same positional set, with the fine-tuning process displayed in the right half of Figure 4.

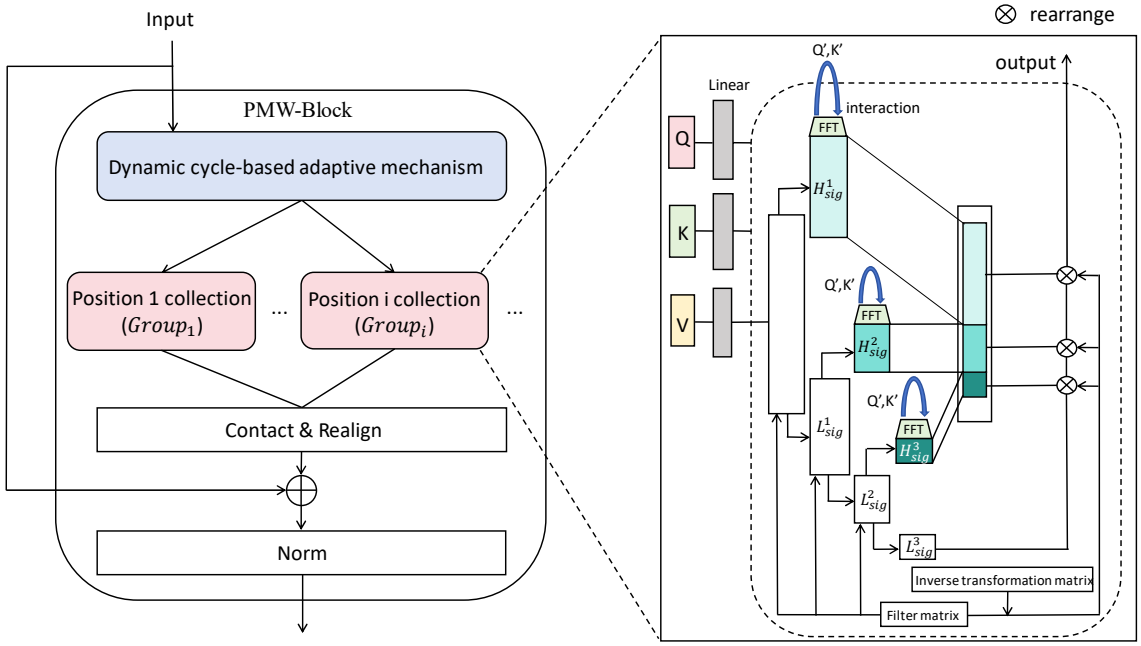

Figure 4. Periodic multi-level wavelet block (PMW-Block).

In the study of the periodic-based dynamic adaptive mechanism, initially, the least common multiple P_{LCM} of the potentially significant multiple periods (corresponding to the specific periodic components $\{P_1, \ldots, P_n\}$ obtained in Section 3.2.1) is calculated. Based on this least common multiple, consecutive data blocks are divided:

$$K = \left\lfloor \frac{N}{P_{\text{LCM}}} \right\rfloor, B_k = \left\{ x_{(k-1)*P_{\text{LCM}}+1}, x_{(k-1)*P_{\text{LCM}}+2}, \ldots, x_{k*P_{\text{LCM}}} \right\} \tag{24}$$

Here, N denotes the length of the sequence, $\lfloor \cdot \rfloor$ denotes the floor function. B_k represents the k-th

data block, where the range of k is [1, K], including data points from (k-1)*P_{LCM}+1 to k*P_{LCM}. Next, data points in the same position across all blocks are aggregated into a group:

$$Group_i = \{ B_k(i) \mid k = 1,2,\dots,K \} \tag{25}$$

where $Group_i$ is the group of data points at position $i \in \{1,2,\dots,P_{\mathrm{LCM}}\}$ after aggregation. Each dataset will focus on common features within periodic structures, enhancing the accuracy of analyses when following the same or similar periodic patterns. However, if the dataset cannot guarantee sufficient length for subsequent refinement, it is necessary to appropriately adjust the lookback length or discard the divisional structure to ensure the effectiveness of the model's processing. After respective refinements (right half of Figure 4) of all groups, the complete signal is reconstructed based on the inverse of decomposition, followed by residual connections, and ultimately normalization.

During the refinement, we apply a multi-level wavelet fusion with a cross-attention mechanism. In the forward propagation, the multilayer perceptrons (MLPs) first preprocess the input queries (Q), keys (K), and values (V), respectively, to modify the data to the proper processing dimensions, allowing for multi-level decomposition at different frequency levels. As shown in Algorithm 2, the multi-level decomposition strategy [41] employs an even-odd interleaving sampling method, concatenating even and odd segments of data in the feature dimension. It then multiplies with the pre-calculated filter matrices from Section 2.4, progressively extracting the high-frequency (H_{sig}^{i}) and low-frequency (L_{sig}^{i}) components of the signal. The process is carried out recursively, with the low-frequency part further decomposed into H_{sig}^{i+1} and L_{sig}^{i+1} in the next layer using the same method. Alternating sampling and matrix multiplication operations reduce the sequence length by half while maintaining the feature dimension, thereby effectively overcoming the limitations of high computational demands and memory usage.

Algorithm 2. Multi-level Decomposition Strategy

Input: fragment $x \in \mathbb{R}^{N \times d_{\mathrm{model}}}$ to be decomposed

Parameter: pre-computed filter matrices H_0、H_1、G_0 and G_1 (see Section 2.4)

Output: low- and high-frequency component lists L_{list} and H_{list} after multi-level decomposition

initialize $\hat{x}^0, i$: $\hat{x}^0 = x, i = 1$

while decomposition level i not reached and signal meets continue condition:

  split input sequence length by even and odd indices into two sub-sequences:

  $x_{\mathrm{even}} = \{\hat{x}^{i-1}[2j]\}, x_{\mathrm{odd}} = \{\hat{x}^{i-1}[2j+1]\}$ for $j = 0,1,\dots,\lfloor \mathrm{len}(\hat{x}^{i-1})/2 \rfloor - 1$

  concatenate even and odd segments along the feature dimension:

  $\hat{x}^{i-1} = \mathrm{cat}(x_{\mathrm{even}}, x_{\mathrm{odd}}, \mathrm{dim} = -1)$

  calculate high-frequency signal H_{sig}^{i} and low-frequency signal L_{sig}^{i}:

  $H_{\mathrm{sig}}^{i} = \hat{x}^{i-1} \times \mathrm{cat}(G_0^{\mathrm{T}}, G_1^{\mathrm{T}})$

  $L_{\mathrm{sig}}^{i} = \hat{x}^{i-1} \times \mathrm{cat}(H_0^{\mathrm{T}}, H_1^{\mathrm{T}})$

  append H_{sig}^{i} and L_{sig}^{i} to their respective lists:

  L_{list}.append(L_{sig}^{i})

  H_{list}.append(H_{sig}^{i})

  update for the next decomposition level:

  $\hat{x}^{i} = L_{\mathrm{sig}}^{i}$, $i = i + 1$

End while

Return $L_{\mathrm{list}}, H_{\mathrm{list}}$

After converting the time-domain signals of Q', K', V' at the same frequency level into the frequency domain via Fourier transform respectively, the cross-attention mechanism is applied to further adjust the weight distribution in the frequency domain.

After processing the attention mechanism in the frequency domain, we perform an inverse Fourier transform on the frequency domain data to convert it back to the time domain. Subsequently, the multi-level wavelet module reconstructs the data across various frequency levels through inverse wavelet transformation. Specifically, we use an inverse transformation filter matrix (obtained by multiplying the wavelet decomposition filters and the corresponding inverse transformation matrices of the wavelet bases) to progressively reconstruct the details and approximations of each decomposition level from bottom to top. In each layer of reconstruction, we use an even-odd rearrangement method to recombine the high and low frequency components, restoring them to the representation of the previous layer:

$$V_i = \text{even0dd}\big(\text{cat}(V_{i+1}, U_i)\big) \tag{26}$$

Here, V_{i+1} represents the reconstruction result of the previous layer, U_i denotes the high-frequency component of the current layer, and the even0dd function indicates the even-odd rearrangement operation. By recursively performing inverse wavelet transformations, we ultimately reconstruct the representation of the original signal.

4. Data sources and experimental set-up

4.1. Data sources

The dataset used in this paper is sourced from https://mds.nmdis.org.cn/, with the primary tidal dataset selected from the Dandong area (coordinates: 40°7′N, 124°24′E). Dandong, located east of Dandong City in Liaoning Province, China, has significant tidal height variations, making it an ideal location for marine activities and ecological research. The experimental details described in Section 4.2 and the parameter analysis in Section 5.2 are introduced using this area as an example.

We use tidal data spanning one year (2023), with the sampling interval measured in hours. The dataset includes tidal height data (unit: cm) referenced to the tidal datum, which we primarily use for experiments.

To enhance the breadth and diversity of data sources for this study, multiple tidal monitoring stations were selected, including Wusong (31°24′N, 121°30′E), Xiamen (24°27′N, 118°4′E), and Fangchenggang (21°36′N, 108°20′E) in China, along with Busan (35°6′N, 129°2′E) in South Korea, Kamaishi (39°16′N, 141°53′E) in Japan, San Francisco (37°48′N, 122°28′W) in the United States, and Sydney (33°51′S, 151°13′E) in Australia. These monitoring stations are distributed across different marine areas, spanning both hemispheres, and covering locations with unique geographical conditions and tidal characteristics. Due to variations in natural conditions, there are significant differences in the tidal characteristics among these monitoring stations, including daily tidal times and tidal ranges. By conducting a multi-regional comparative study of these representative monitoring stations, the analysis results are given practical significance.

4.2. Experimental details

Based on the description of astronomical tides in physical oceanography, we calculate the power

spectral density maps based on Eq (20) during the experiments, limiting the frequency in the periodograms to the range of 0.001Hz to 0.1 Hz. Furthermore, 0.1% of the maximum value of the power spectral density is set as the threshold for peak detection to exclude peaks caused by noise or unrepresentative fluctuations.

According to the periodogram analysis results, we found that most results have slight deviations from the defined values of astronomical tides. This discrepancy may be caused by various actual tidal influencing factors, including meteorological conditions, seabed topography, coastline shape, and human interventions. To simplify and more clearly display the main energy concentration points in the periodogram, we use an approximation merging approach. Concretely, the obtained periodic values are grouped based on their proximity; for each group of close periodic values, the one with the highest energy is chosen as representative. It should be noted that, considering our dataset is sampled hourly, the model's embedding layer does not account for minutes; the convolution kernels used for extracting multi-level trends are integers; when using Eq (21), the lag periods also need to be rounded, thus the representative periodic values containing decimals should be rounded to the nearest whole number. Subsequently, this paper defines periodic values with an autocorrelation coefficient greater than 0.7 as specific periods. For the Dandong area in China, based on the serial calculations of Eqs (20) and (21), the experimental results selected convolution kernel array [24, 12] as the periods in Section 3.2.1 and as the sizes of the convolution kernels in Section 3.2.2.

During the PMW-Block processing, to ensure that the length of each segmented data fragment meets the conditions for effective multi-level wavelet decomposition, the following condition must be satisfied: $\frac{S}{P_{\mathrm{LCM}}} \geq L_{\min}$, where $L_{\min}$ represents the minimum fragment length (set to 8 in this study). If the length of the data fragments does not meet this condition (i.e., in cases where the predicted step sizes in our experiment are 12 or 24), a multi-level wavelet decomposition method is applied directly to analyze the periodic components, and the initial dynamic adaptive segmentation and grouping strategy is discarded.

After experimenting with a large number of stations, we have found that short-period variations are more important and apparent in tidal predictions and daily observations, whereas the long-period effects of astronomical tides on the tidal dataset have not been found to be significant according to the above-mentioned procedures. Therefore, the design of our model focuses more on extracting short-period features.

4.3. Training settings

To ensure the effectiveness of the model training, we have divided the dataset into training, validation, and test sets with ratios of 7 : 2 : 1, respectively. The division is as follows:

Training set: Comprises 70% of the total dataset, used for the training process of the model.

Validation set: Comprises 20% of the total dataset, used for hyperparameter adjustment and performance evaluation of the model.

Test set: Comprises 10% of the total dataset, used for performance testing of the model.

The model was implemented and trained using Python 3.9 and PyTorch 2.1.2. The experimental platform is Ubuntu 20.04.6 LTS, the CPU is AMD Ryzen 9 5950X @ 3.4 GHz, and the graphics card is GPU: NVIDIA GeForce RTX 3090 24 GB.

5. Model evaluation and experimental analysis

5.1. Indicators for model evaluation

Prediction performance can be assessed by the following performance metrics: MAE (mean absolute error), RMSE (root mean square error) and MSE (mean square error):

$$\mathrm{MAE}(y_{\mathrm{true}}, y_{\mathrm{predict}}) = \frac{1}{N}\sum_{i=1}^{N}\left|y_{\mathrm{true}} - y_{\mathrm{predict}}\right| \tag{27}$$

$$\mathrm{RMSE}(y_{\mathrm{true}}, y_{\mathrm{predict}}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_{\mathrm{true}} - y_{\mathrm{predict}}\right)^2} \tag{28}$$

$$\mathrm{MSE}(y_{\mathrm{true}}, y_{\mathrm{predict}}) = \frac{1}{N}\sum_{i=1}^{N}\left(y_{\mathrm{true}} - y_{\mathrm{predict}}\right)^2 \tag{29}$$

5.2. Parameter resolution

The network structural parameters and training hyperparameters of the model in this paper are shown in Table 1:

Table 1. Network structure parameters and training hyperparameters.

Encoder	Number of layer	2
	Number of distilling layer	1
	Memory factors' channel dimension	[8,16,8]
	Number of multi-scale multi-level convolutional iterations	2
Decoder	Number of layer	1
	Number of multi-level wavelet block decompositions	3
	Minimum sequence length required for multi-level wavelet decomposition	8
	Top-k frequency components selected by amplitude	1
	Number of multi-scale multi-level convolutional iterations	1
Training hyperparameters	Seq_len	48
	Label_len	12
	Pred_len	12
	Batch_size	32
	Learning_rate	1×10^{-3}
	Hidden dimension	512
	Optimizer	Adam
	Dropout rate	0.05
	Sampling factor for Informer's ProbSparse self-attention	5
	Loss function	MSE
	Wavelet basis	Legendre

5.3. Ablation experiments

To study the impact of different configurations on the performance of Informer, this experiment is designed with various configuration schemes. These configurations include the memory factors (A), periodic multi-level wavelet block (B), multi-scale multi-level convolution (C), as well as combined configurations A + B, A + C, B + C, and finally, the integrated model (A + B + C) is tested. The results of the ablation experiments are shown in Table 2.

Table 2. Comparison of ablation experiments.

	A	B	C	A + B	A + C	B + C	A + B + C
MAE	0.1265	0.1207	0.1200	0.1101	0.1084	0.1036	**0.0877**
RMSE	0.1637	0.1563	0.1532	0.1424	0.1389	0.1320	**0.1134**
MSE	0.0268	0.0244	0.0234	0.0202	0.0193	0.0174	**0.0128**

From the indicators MAE, RMSE, and MSE, it is observed that as the model structure becomes increasingly complex, there is an improvement in predictive performance. In terms of the single configuration, the increase of each configuration on the model performance is generally similar. The performance of paired configurations is superior to that of individual configurations, with the greatest enhancement seen when the multi-scale multi-level convolution is used in combination with the periodic multi-level wavelet block. We believe this enhancement is due to a higher degree of correlation between these two configurations. Under the integrated model configuration (A + B + C), MAE decreases to 0.0877, RMSE to 0.1134, and MSE to 0.0128, demonstrating the effectiveness of the ensemble configurations.

5.4. Comparative experiments

To demonstrate the exceptional performance of our prediction model, we designed two rounds of testing comparisons to evaluate the model's predictive capability over the same time span (set to 12). Initially, to test the adaptability of the base model on a tidal dataset, we selected a series of Transformer-based model variants for the first round of comparison. This round of comparison includes the Transformer and three classic Transformer variants: Informer, Autoformer, and Prayformer [42], with the results shown in Figure 5.

The analysis indicates that, in predicting 12 steps ahead, Informer performed best on the metrics MSE (0.0394), MAE (0.1586), and RMSE (0.1987). Transformer and Prayformer showed similar performance, while Autoformer had the highest loss values on these three metrics, indicating the poorest performance. This demonstrates that the Informer surpasses the Transformer in terms of prediction accuracy, computational efficiency, and memory usage, as well as excels in balancing these three aspects better than other Transformer-based models. Additionally, this validates the effectiveness of selecting this base model for the tidal dataset.

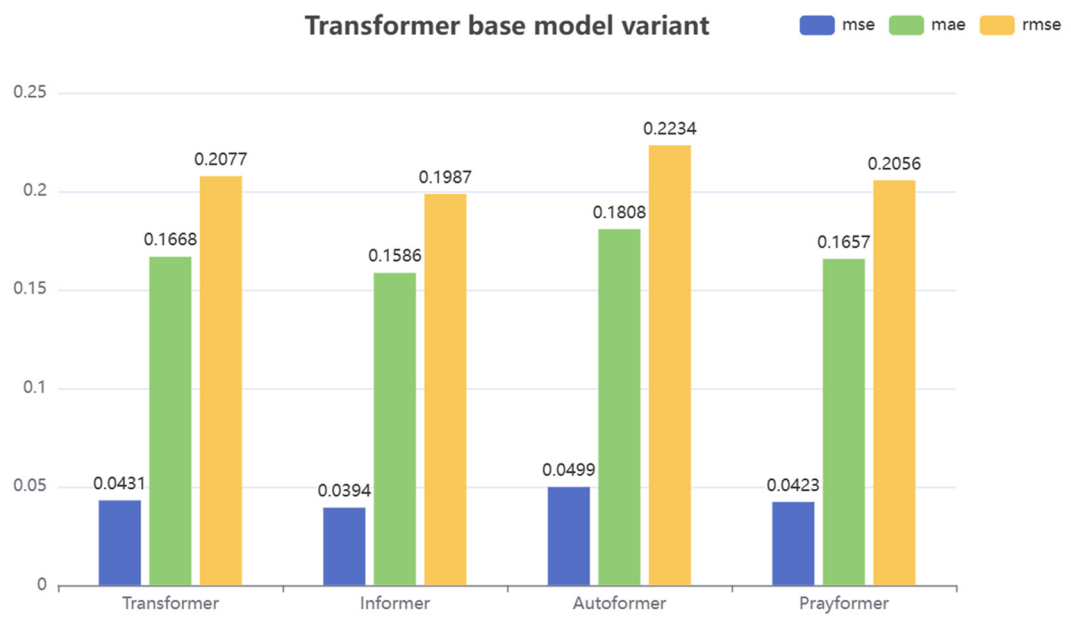

Figure 5. Comparison of Transformer-based model performance.

Subsequently, other comparative experiments included the model proposed in this paper, LSTM, Attention + TCN (shortened as Attn + TCN), and several advanced prediction models introduced in the past two years. Table 3 displays the final accuracy assessment scores for these models.

Table 3. Other comparative experiments.

Models	MAE	RMSE	MSE
Ours	**0.0877**	**0.1134**	**0.0128**
LSTM	0.1890	0.2412	0.0582
Attn+TCN	0.1381	0.1727	0.0298
MICN	0.1060	0.1404	0.0197
NS_Transformer	0.1340	0.1720	0.0296
NS_Informer	0.1134	0.1472	0.0216
SCINet	0.1340	0.1712	0.0293
FedFormer	0.1210	0.1597	0.0255
Dlinear	0.1506	0.1938	0.0375
Nlinear	0.1739	0.2263	0.0512

The results indicate that models with smaller numerical values predict more accurately. Three representative models among these were selected for numerical analysis: Compared with LSTM, the baseline model Informer, and the advanced model MICN [43], using MAE as the evaluation metric, our model improved by 53.5%, 44.7%, and 17.2%. Using RMSE as the evaluation metric, our model improved by 52.9%, 42.9%, and 19.2%. Using MSE as the evaluation metric, our models improved by 78.0%, 67.5%, and 35.0%. In terms of the average of the overall evaluation metrics, our model improved by 61.4%, 51.7%, and 23.8% over the LSTM, Informer, and MICN, respectively. Across all evaluation metrics, our model demonstrated superior predictive accuracy compared to other models, exhibiting optimal performance. According to Table 3, the performance of MICN and

NS_Informer [25] is quite exceptional, closely following our model. The performance of NS_Transformer, SCINet [44], Attn+TCN, and FedFormer is moderate, while the performance of LSTM, Dlinear [45], and Nlinear [45] is relatively low, indicating that they may not be suitable for predicting tidal datasets. We selected several representative experimental results and displayed them in Figure 6.

From the graph, we can see how different models perform on a real tidal dataset. Through the fitting of time series graphs, it is observed that the NS_Transformer, which also focuses on predicting abrupt changes in non-stationary data, may not be precise during the fitting process, leading to greater errors in some cases. The LSTM underperforms in capturing mutation points and handling long-term dependencies. Similarly, the Attn+TCN fails to accurately capture areas with significant local fluctuations, resulting in substantial deviations in peak predictions.

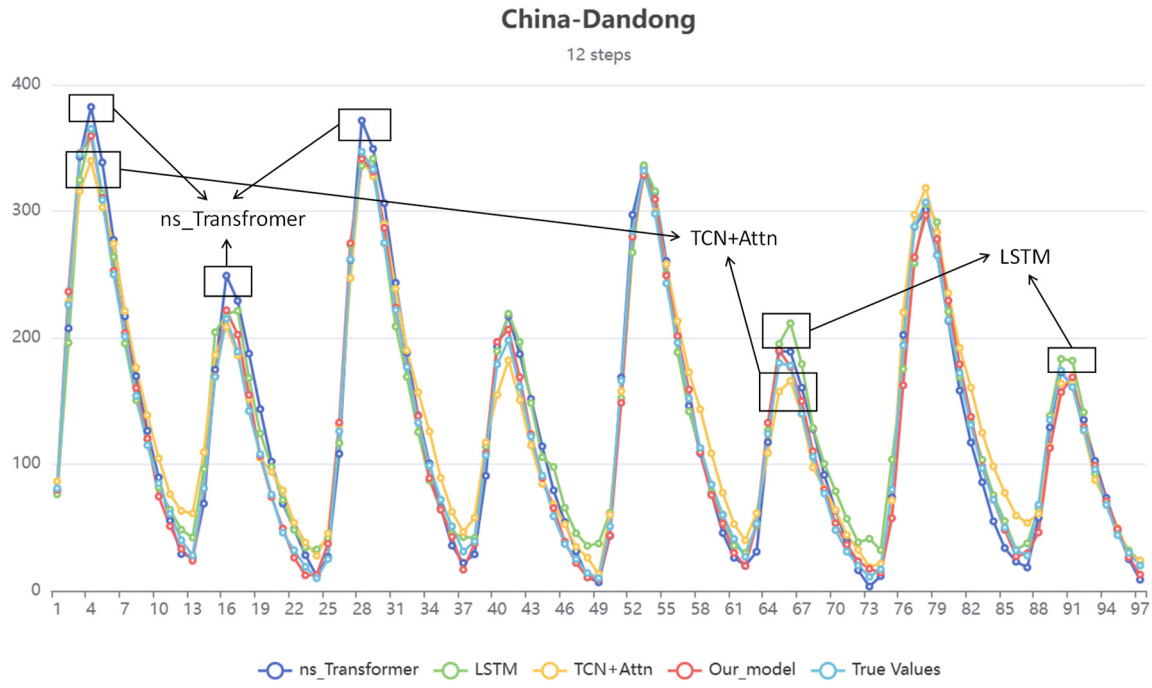

Figure 6. Tidal prediction results of several comparison models for Dandong area.

5.5. Multi-time span experiments

To assess the model's predictive performance over different time spans, we also selected tasks with time spans of 24 hours, 48 hours, and 96 hours for a systematic analysis of the prediction results, as shown in Table 4.

Our model outperforms the Transformer and Informer across all evaluation metrics (MAE, RMSE, MSE). Compared to our model and the Transformer, the Informer shows a significant increase in error as the prediction span progressively doubles to 24, 48, and 96 steps, relative to the 12-step prediction task. This indicates that despite the Informer's ProbSparse self-attention mechanism and distilling operation being designed to focus more on the significant information within the sequence to reduce spatio-temporal complexity, inevitably, much crucial information is lost during the downsampling process. The results indicate that the Informer may not be suitable for long-term tidal prediction tasks.

It is also noteworthy that the Transformer's MSE doubled from 0.068 to 0.1366 as the forecast span increased from 48 to 96 steps. In contrast, our model maintained a stable error rate during the same period (0.059 > 0.067). Figure 7 displays the data fitting under different spans.

Table 4. Comparison of prediction results across different time spans.

Prediction Span	Models	MAE	RMSE	MSE
24 hours	Ours	**0.1049**	**0.1298**	**0.0168**
	Transformer	0.1588	0.1944	0.0378
	Informer	0.4050	0.4763	0.2268
48 hours	Ours	**0.1938**	**0.2429**	**0.0590**
	Transformer	0.2164	0.2607	0.0680
	Informer	0.6456	0.8309	0.6904
96 hours	Ours	**0.1928**	**0.2588**	**0.0670**
	Transformer	0.2811	0.3696	0.1366
	Informer	0.7411	0.9281	0.8613

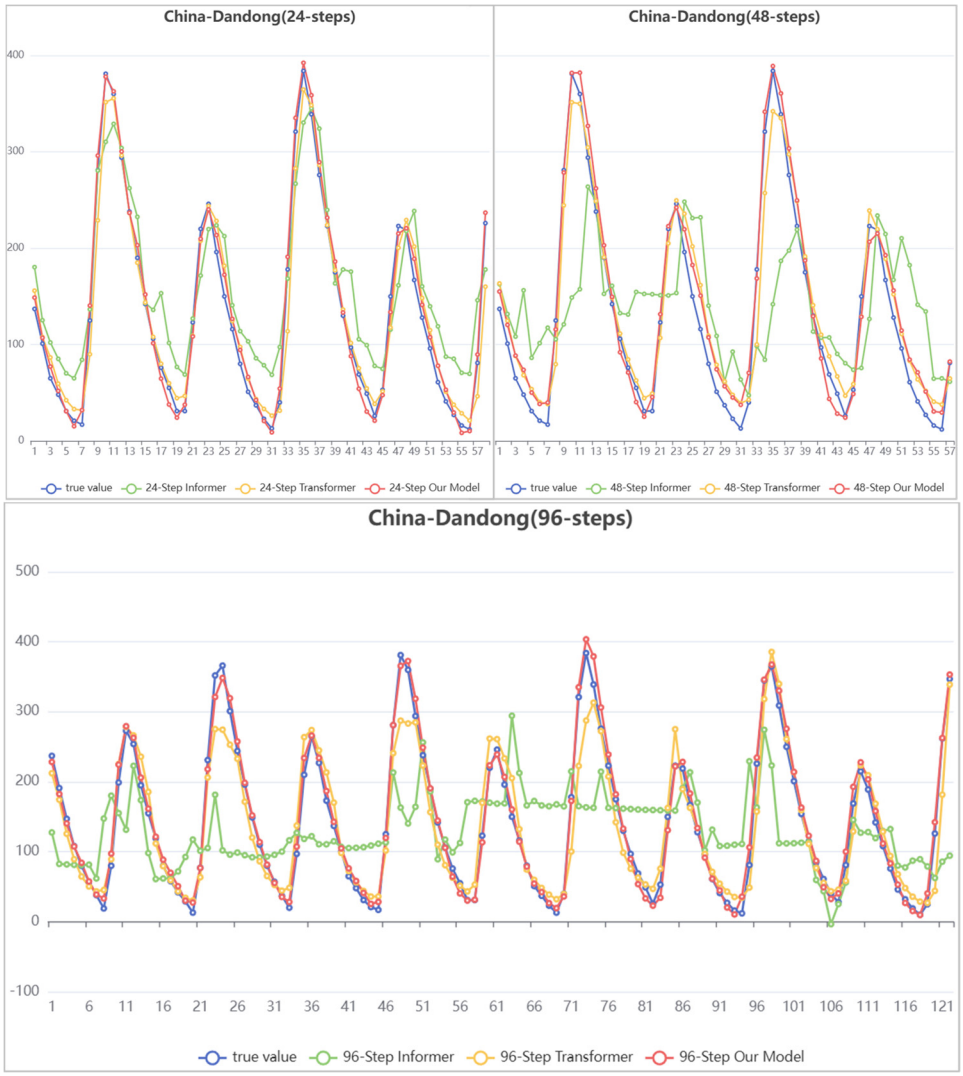

Figure 7. Data fitting across time spans.

As can be seen from the figure, the Informer struggles to accurately capture basic trends in long-term forecasting tasks, and the Transformer also lacks in extreme value prediction. In contrast, our model maintains high accuracy in predicting peaks and troughs across different time spans. From the perspective of model structure, it can also be confirmed that the PMW-Block (Section 3.3) effectively captures the long-distance dependencies of the same or similar periodic patterns by aggregating the same positions across periods. This strategy meets the demand for high precision in long-term prediction.

5.6. Multi-site experiments

We also selected seven tidal sites with varying geographical locations and environmental conditions. The selection of these sites allows for a comprehensive examination of our model's applicability and stability. After calculations, the tidal characteristics of the sites at China-Fangchenggang, Japan-Kamaishi, and USA-San Francisco are more consistent with the single-period mode. Section 5.4 has verified that the Informer outperforms other Transformer-based model variants in predicting 12-step scenarios. To further demonstrate that our model effectively extends the foundational theoretical framework of the Informer, and due to space limitations, Table 5 presents only the multi-site comparison results between our model and the Informer.

Table 5. Comparison of multi-site experiment results (12 steps).

Station	MAE (Ours)	MAE (Informer)	RMSE (Ours)	RMSE (Informer)	MSE (Ours)	MSE (Informer)
Wusong	**0.0830**	0.1681	**0.1068**	0.2263	**0.0114**	0.0512
Xiamen	**0.0805**	0.1123	**0.1001**	0.1400	**0.0101**	0.0196
Fangchenggang	**0.0811**	0.1869	**0.1028**	0.2271	**0.0105**	0.0516
Busan	**0.0758**	0.0924	**0.0945**	0.1168	**0.0089**	0.0136
Kamaishi	**0.0844**	0.1323	**0.1074**	0.1664	**0.0115**	0.0277
San Francisco	**0.0813**	0.1043	**0.1056**	0.1334	**0.0111**	0.0178
Sydney	**0.0914**	0.0966	**0.1134**	0.1254	**0.0128**	0.0157
Average	**0.0825**	0.1275	**0.1043**	0.1622	**0.0109**	0.0281

Evaluating the normalized values, it can be concluded that our model exhibits similar performance across different sites, with more stable predictions compared to the Informer, and is applicable to tidal datasets of various period types. Specifically, compared to the Informer, the proposed model in this paper achieved average improvements of 35.2%, 35.6%, and 61.2% in the three evaluation metrics, respectively. To further substantiate our model's fit, we selected sites with relatively large tidal ranges (Fangchenggang, China) and complex fluctuation patterns (Kamaishi, Japan) for deeper analysis. To show more intuitively, we used the following method in the plotting: Real data points (green dots) cover our model's predictions (blue dots), and then the Informer model's forecasts (yellow dots) are overlaid on top. Figure 8 shows the data fitting for the two sites.

From the fitting diagrams, it can be observed that in areas with significant data fluctuations, such as extreme points, our model shows better conformity compared to the Informer. Additionally, for sudden tidal changes (as shown in the zoomed-in areas of the diagrams), our model also demonstrates robustness. Overall, our model can be effectively applied to predictive tasks for tidal data at various

sites under real-world conditions.

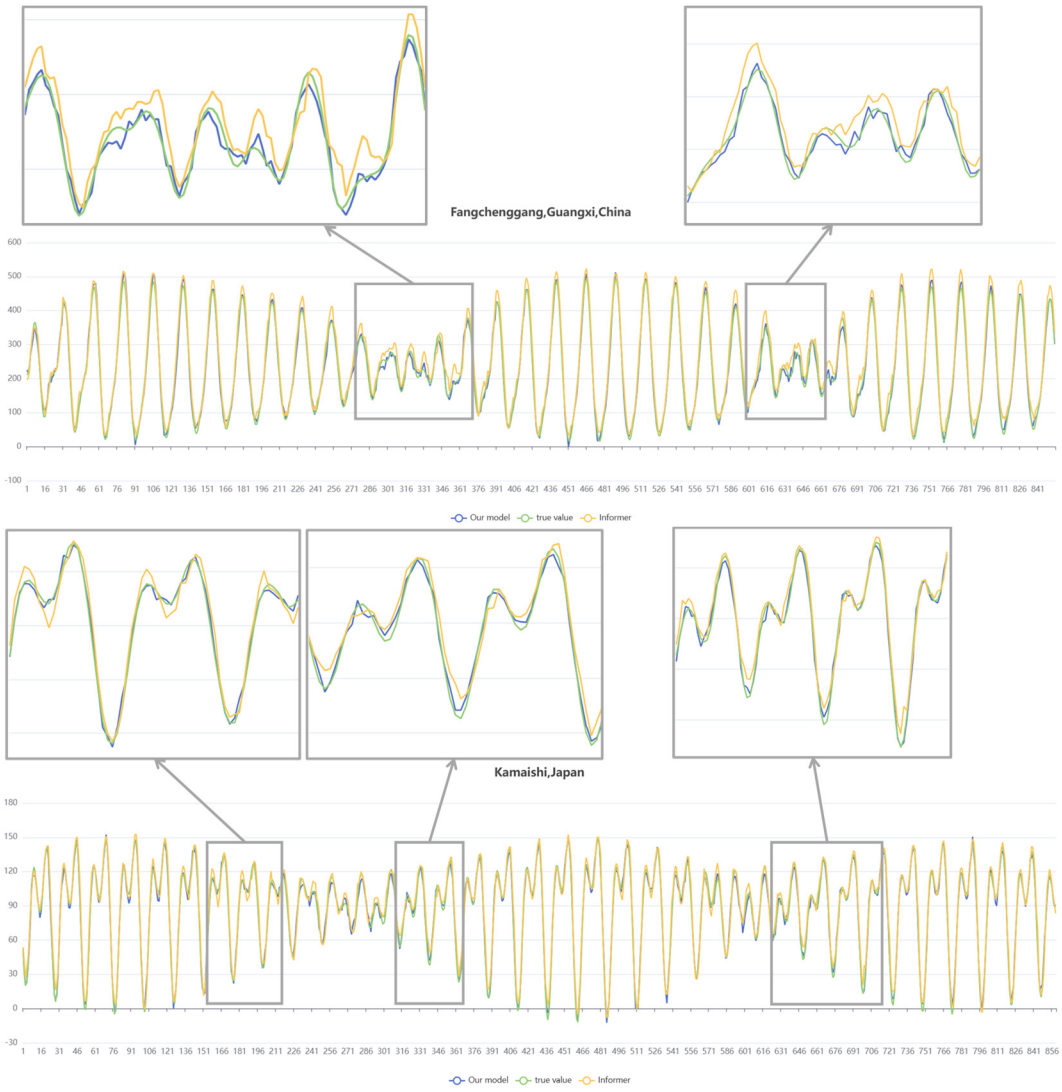

Figure 8. Data fitting across sites.

6. Summary and outlook

In this paper, we propose a multi-scale Informer-based model that fuses memory factors and wavelet denoising that is capable of accurately predicting tidal series data characterized by non-stationarity and multiple periodicities. Some conclusions can be drawn from the study:

• By integrating normalized data with mean and standard deviation characteristics, comprehensive modeling of multiple statistical features is achieved. The introduction of memory factors not only enhances the predictability of the data, but also effectively avoids the issue of over-stabilization, ensuring the effective capture of key temporal dependencies and sudden events in the raw series. Furthermore, feature fusion enhances the data representation capability, enabling the model to effectively capture deep patterns in the data from a more comprehensive perspective.

• The set of signal frequencies defined by the Fourier transform, periodogram, and autocorrelation coefficients together reinforces the core periodic structure of the signal as well as

improves the model's overall understanding of the dynamic properties of periodic signals.

• The combination of progressive extraction and iterative optimization offers greater flexibility and accuracy than traditional one-time trend extraction. Employing larger convolutional kernels to extract macro trends, followed by smaller kernels to identify micro-dynamic changes, allows each analytical stage to focus on features at different levels of the data.

• The periodic multi-level wavelet block employs dynamic data partitioning, combined with multi-level wavelets and cross-attention mechanisms, significantly enhancing the accuracy of long-term prediction and the model's ability to capture complex periodic features.

These mechanisms collectively enable the model to achieve excellent predictive performance, validated at multiple representative tidal stations with wide geographic distribution, varying tidal timings, and significant altitude differences. Our model's design and algorithms also demonstrate broad applicability to different time-span predictive tasks. However, the model has the following limitations. For each limitation, we propose corresponding directions for future research: Although the model demonstrates good performance in long-term predictions, its accuracy inevitably declines as the prediction horizon extends. Future research could explore the incorporation of advanced time series analysis methods or the optimization of the model structure to improve its ability to capture long-term patterns. Additionally, our existing model relies solely on data from a single station, limiting its ability to consider spatial correlations between different tidal stations. A potential avenue for future work is to integrate concepts from graph theory, leveraging data from neighboring stations to inform and adjust the final predictions. Furthermore, we plan to extend the application of this model to other practical domains to thoroughly assess its generalizability.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This research was funded by the Research on Key Technology of Intelligent Extraction for Remote Sensing Monitoring of Marine Areas and Islands, grant number SDGP37000000020240200 1009A_001.

Conflict of interest

The authors declare there is no conflict of interest.

References

1. T. Bongarts Lebbe, H. Rey-Valette, É. Chaumillon, G. Camus, R. Almar, A. Cazenave, et al., Designing coastal adaptation strategies to tackle sea level rise, *Front. Mar. Sci.*, **8** (2021), 740602. https://doi.org/10.3389/fmars.2021.740602
2. G. Griggs, B. G. Reguero, Coastal adaptation to climate change and sea-level rise, *Water*, **13** (2021), 2151. https://doi.org/10.3390/w13162151

3. A. T. Doodson, The analysis and predictions of tides in shallow water, *Int. Hydrogr. Rev.*, **33** (1958), 85–126.
4. R. E. Kalman, A new approach to linear filtering and prediction problems, *J. Basic Eng.*, **82** (1960), 35–45. https://doi.org/10.1115/1.3662552
5. G. Evensen, Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics, *J. Geophys. Res.*, **99** (1994), 10143–10162. https://doi.org/10.1029/94JC00572
6. Y. Yang, Y. Gao, Z. Wang, X. Li, H. Zhou, J. Wu, Multiscale-integrated deep learning approaches for short-term load forecasting, *Int. J. Mach. Learn. Cybern.*, **15** (2024), 6061–6076. https://doi.org/10.1007/s13042-024-02302-4
7. S. Zhang, Z. Zhao, J. Wu, Y. Jin, D. S. Jeng, S. Li, et al., Solving the temporal lags in local significant wave height prediction with a new VMD-LSTM model, *Ocean Eng.*, **313** (2024), 119385. https://doi.org/10.1016/j.oceaneng.2024.119385
8. J. C. Yin, A. N. Perakis, N. Wang, An ensemble real-time tidal level prediction mechanism using multiresolution wavelet decomposition method, *IEEE Trans. Geosci. Remote Sensing*, **56** (2018), 4856–4865. https://doi.org/10.1109/TGRS.2018.2841204
9. T. L. Lee, Back-propagation neural network for long-term tidal predictions, *Ocean Eng.*, **31** (2004), 225–238. https://doi.org/10.1016/S0029-8018(03)00115-X
10. N. Portillo Juan, V. Negro Valdecantos, Review of the application of artificial neural networks in ocean engineering, *Ocean Eng.*, **259** (2022), 111947. https://doi.org/10.1016/j.oceaneng.2022.111947
11. C. L. Giles, G. M. Kuhn, R. J. Williams, Dynamic recurrent neural networks: Theory and applications, *IEEE Trans. Neural Netw.*, **5** (1994), 153–156. https://doi.org/10.1109/TNN.1994.8753425
12. S. Fan, N. Xiao, S. Dong, A novel model to predict significant wave height based on long short-term memory network, *Ocean Eng.*, **205** (2020), 107298. https://doi.org/10.1016/j.oceaneng.2020.107298
13. L. H. Bai, H. Xu, Accurate estimation of tidal level using bidirectional long short-term memory recurrent neural network, *Ocean Eng.*, **235** (2021), 108765. https://doi.org/10.1016/j.oceaneng.2021.108765
14. Q. R. Luo, H. Xu, L. H. Bai, Prediction of significant wave height in hurricane area of the Atlantic Ocean using the Bi-LSTM with attention model, *Ocean Eng.*, **266** (2022), 112747. https://doi.org/10.1016/j.oceaneng.2022.112747
15. J. Oh, K. D. Suh, Real-time forecasting of wave heights using EOF-wavelet-neural network hybrid model, *Ocean Eng.*, **150** (2018), 48–59. https://doi.org/10.1016/j.oceaneng.2017.12.044
16. H. H. H. Aly, Intelligent optimised deep learning hybrid models of neuro wavelet, Fourier series and recurrent Kalman filter for tidal currents constitutions forecasting, *Ocean Eng.*, **218** (2020), 108254. https://doi.org/10.1016/j.oceaneng.2020.108254
17. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, et al., Attention is all you need, in *Advances in Neural Information Processing Systems*, **30** (2017), 5998–6008. https://doi.org/10.48550/arXiv.1706.03762
18. S. Wang, Z. Huang, B. Zhang, X. Heng, Y. Jiang, X. Sun, Plot-aware Transformer for recommender systems, *Electron. Res. Arch.*, **31** (2023), 3169–3186. https://doi.org/10.3934/era.2023160

19. Y. Li, X. Wang, Y. Guo, CNN-Trans-SPP: A small Transformer with CNN for stock price prediction, *Electron. Res. Arch.*, **32** (2024), 6717–6732. https://doi.org/10.3934/era.2024314
20. J. Wan, N. Xia, Y. Yin, X. Pan, J. Hu, J. Yi, TCDformer: A transformer framework for non-stationary time series forecasting based on trend and change-point detection, *Neural Netw.*, **173** (2024), 106196. https://doi.org/10.1016/j.neunet.2024.106196
21. H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, et al., Informer: Beyond efficient transformer for long sequence time-series forecasting, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **35** (2021), 11106–11115. https://doi.org/10.48550/ARXIV.2012.07436
22. H. Wu, J. Xu, J. Wang, M. Long, Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting, *Adv. Neural Inf. Process. Syst.*, **34** (2021), 22419–22430. https://doi.org/10.48550/arXiv.2106.13008
23. T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, R. Jin, FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting, *J. Int. Law Policy*, **3** (2022), 321–322. https://doi.org/10.48550/arXiv.2201.12740
24. Y. Nie, N. H. Nguyen, P. Sinthong, J. Kalagnanam, A time series is worth 64 words: Long-term forecasting with transformers, in *International Conference on Learning Representations*, (2022). https://doi.org/10.48550/arXiv.2211.14730
25. Y. Liu, H. Wu, J. Wang, M. Long, Non-stationary transformers: exploring the stationarity in time series forecasting, preprint, arXiv:2205.14415. https://doi.org/10.48550/arXiv.2205.14415
26. Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, M. Long, iTransformer: inverted transformers are effective for time series forecasting, *J. Int. Law Policy*, **3** (2023), 321–322. https://doi.org/10.48550/arXiv.2310.06625
27. Y. H. H. Tsai, S. Bai, M. Yamada, L. P. Morency, R. Salakhutdinov, Transformer dissection: A unified understanding of transformer's attention via the lens of kernel, preprint, arXiv:1908.11775. https://doi.org/10.48550/arXiv.1908.11775
28. S. G. Mallat, A theory for multiresolution signal decomposition: the wavelet representation, *IEEE Trans. Pattern Anal. Mach. Intell.*, **11** (1989), 674–693. https://doi.org/10.1109/34.192463
29. S. G. Venkatesh, S. K. Ayyaswamy, S. Raja Balachandar, The Legendre wavelet method for solving initial value problems of Bratu-type, *Comput. Math. Appl.*, **63** (2012), 1287–1295. https://doi.org/10.1016/j.camwa.2011.12.069
30. A. A. Abdulrahman, F. S. Tahir, Face recognition using enhancement discrete wavelet transform based on MATLAB, *Int. J. Eng. Comput. Sci.*, **23** (2021), 1128–1136. https://doi.org/10.11591/ijeecs.v23.i2.pp1128-1136
31. N. Zheng, H. Chai, Y. Ma, L. Chen, P. Chen, Hourly sea level height forecast based on GNSS-IR by using ARIMA model, *Remote Sens.*, **43** (2022), 3387–3411. https://doi.org/10.1080/01431161.2022.2091965
32. D. Lee, S. Lee, J. Lee, Standardization in building an ANN-based mooring line top tension prediction system, *Int. J. Nav. Archit. Ocean Eng.*, **14** (2022), 100421. https://doi.org/10.1016/j.ijnaoe.2021.11.004
33. J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2018), 7132–7141. https://doi.org/10.1109/CVPR.2018.00745

34. H. Han, Z. Liu, M. Barrios, J. Li, Z. Zeng, N. Sarhan, et al., Time series forecasting model for non-stationary series pattern extraction using deep learning and GARCH modelling, *J. Cloud Comput.*, **13** (2024), 2. https://doi.org/10.1186/s13677-023-00576-7
35. I. Yanovitzky, A. VanLear, Time series analysis: Traditional and contemporary approaches, in *The SAGE Sourcebook of Advanced Data Analysis Methods for Communication Research* (eds. A. F. Hayes, M. D. Slater, L. B. Snyder), Sage Publications, (2008), 89–124. https://doi.org/10.4135/9781452272054.n4
36. R. B. Cleveland, W. S. Cleveland, J. E. McRae, I. Terpenning, STL: A seasonal-trend decomposition procedure based on loess, *J. Off. Stat.*, **6** (1990), 3–73.
37. C. Nontapa, C. Kesamoon, N. Kaewhawong, P. Intrapaiboon, A new time series forecasting using decomposition method with SARIMAX model, in *Neural Information Processing*, *Communications in Computer and Information Science*, **1333** (2020), 743–751. https://doi.org/10.1007/978-3-030-63823-8_84
38. T. Kim, B. R. King, Time series prediction using deep echo state networks, *Neural Comput. Appl.*, **32** (2020), 17769–17787. https://doi.org/10.1007/s00521-020-04948-x
39. H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, M. Long, TimesNet: Temporal 2D-variation modeling for general time series analysis, in *International Conference on Learning Representations*, (2022). https://doi.org/10.48550/arXiv.2210.02186
40. A. R. Abdullah, N. M. Saad, A. Zuri, Power quality monitoring system utilizing periodogram and spectrogram analysis techniques, in *Asean Virtual Instrumentation Applications Contest*, (2007). https://doi.org/10.13140/2.1.3109.1841
41. C. Samson, V. U. K. Sastry, A novel image encryption supported by compression using multilevel wavelet transform, *Int. J. Adv. Comput. Sci. Appl.*, **3** (2012). https://doi.org/10.14569/IJACSA.2012.030926
42. S. Liu, H. Yu, C. Liao, J. Li, W. Lin, A. X. Liu, et al., Pyraformer: low-complexity pyramidal attention for long-range time series modelling and forecasting, in *Proceedings of the 10th International Conference on Learning Representations*, 2022. Available from: https://openreview.net/forum?id=0EXmFzUn5I.
43. H. Wang, J. Peng, F. Huang, J. Wang, J. Chen, Y. Xiao, MICN: Multi-scale local and global context modelling for long-term series forecasting, in *International Conference on Learning Representations*, 2023. Available from: https://openreview.net/forum?id=zt53IDUR1U.
44. M. Liu, A. Zeng, M. Chen, Z. Xu, Q. Lai, L. Ma, et al., SCINet: time series modelling and forecasting with sample convolution and interaction, in *Advances in Neural Information Processing Systems*, **35** (2022), 5816–5828. https://doi.org/10.48550/arXiv.2106.09305
45. A. Zeng, M. Chen, L. Zhang, Q. Xu, Are transformers effective for time series forecasting?, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **37** (2023), 11121–11128. https://doi.org/10.48550/arXiv.2205.13504

AIMS Press