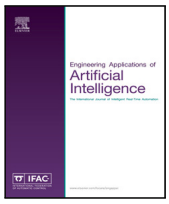
Engineering Applications of Artificial Intelligence 181 (2026) 115449

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



Research paper

Learnable multi-scale decomposition with integrated autocorrelation and confidence-aware time series forecasting

Ibrahim Delibasoglu [a,b,*], Sanjay Chakraborty [a], Fredrik Heintz [a]

[a] Department of Computer and Information Science (IDA), REAL, AIICS, Linköping University, Linköping, Sweden
[b] Department of Software Engineering, Sakarya University, Sakarya, Türkiye

ARTICLE INFO

Dataset link: https://github.com/mribrahim/lms-tsf

Keywords:
Time series
Forecasting
Deep learning

ABSTRACT

In this work, we introduce a time series forecasting architecture that incorporates autocorrelation while leveraging dual encoders operating at multiple scales. Unlike traditional models that rely on predefined trend and seasonal components, our architecture applies learnable decomposition in the frequency domain and employs two separate encoders per scale: one focusing on low-pass filtering to capture trends and the other utilizing high-pass filtering to model seasonal variations. This learnable filtering allows the model to dynamically adapt and isolate trend and seasonal components directly in the frequency domain. A central and distinguishing innovation in our approach is the introduction of an integrated confidence scoring mechanism based on the reconstruction error of the same encoder backbone used for forecasting. This confidence score provides a direct, learned metric of prediction certainty, which we use to dynamically adjust the final forecasting output, effectively mitigating risks from uncertain predictions. We achieve the integration of autocorrelation by computing lagged differences in temporal processing, enabling the model to capture dependencies across time more effectively. In our architecture, each encoder processes the input through fully connected layers to handle temporal and channel interactions, and the same encoder backbone is used for both reconstruction and forecasting outputs. By combining frequency-domain filtering, autocorrelation-based temporal modeling, channel-wise transformations, and proposed confidence scoring mechanism, our model not only accurately captures long-term dependencies and fine-grained patterns, but also operates more efficiently compared to other state-of-the-art methods. Its lightweight design ensures faster processing while maintaining high precision in forecasting across diverse datasets and time horizons.

1. Introduction

Time series data typically comprises an ordered sequence of observed or measured outcomes from a process at fixed time intervals. These datasets are designed to capture relevant information and activities within a specific subject matter. The primary goal of time series applications is to estimate future values by the identification of underlying patterns in historical data. Time series forecasting (TSF) is a frequently used technique in fields like traffic congestion anticipation, weather forecasting, and stock market prediction. Making more informed and efficient decisions is made possible by accurate forecasting, which gives decision-makers the ability to recognize and reduce risks (Li and Law, 2024). Due to the complex and ever-changing nature of real-world systems, observed time series often exhibit intricate temporal patterns. These patterns can involve a mix of increasing, decreasing, and fluctuating trends, making forecasting highly challenging. Multivariate time series forecasting involves predicting future values of multiple interrelated time-dependent variables based on their historical data. Unlike univariate forecasting, which deals with a single variable, multivariate forecasting considers the interactions and dependencies between multiple variables, allowing for more complex and potentially more accurate predictions across several dimensions (Zerveas et al., 2021). In time series data, the trend component represents the long-term directional movement, showing consistent upward or downward patterns, while the seasonal component captures regular, repeating fluctuations over fixed intervals, such as daily, monthly, or yearly cycles. Variations in trends can be linear or nonlinear and may shift or reverse over time, whereas seasonal variations are typically periodic but can vary in amplitude or timing due to external factors. Both components often overlap, making the task of isolating and forecasting them more complex (Chen et al., 2023). In time series analysis, frequency domain filtering has two types. A low-pass filtering is commonly used to capture seasonal variations by allowing low frequency components of the data to pass through while attenuating high frequency

* Corresponding author at: Department of Computer and Information Science (IDA), REAL, AIICS, Linköping University, Linköping, Sweden.
E-mail addresses: ibrahimdelibasoglu@sakarya.edu.tr (I. Delibasoglu), sanjay.chakraborty@liu.se (S. Chakraborty), fredrik.heintz@liu.se (F. Heintz).

https://doi.org/10.1016/j.engappai.2026.115449
Received 29 May 2025; Received in revised form 4 December 2025; Accepted 13 June 2026
Available online 19 June 2026
0952-1976/© 2026 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

noise. This process helps isolate recurring seasonal patterns, such as annual or monthly cycles, which are important in fields like finance and climatology (Hyndman and Athanasopoulos, 2018). Conversely, high-pass filtering is employed to model trends by passing high frequency components and filtering out lower frequency signals, which represent long-term, smooth variations. This technique is particularly useful for identifying short-term fluctuations and trends, removing the impact of long-term cyclical patterns (Shumway and Stoffer, 2006). Both methods provide a clearer decomposition of time series data, enabling analysts to focus on specific components of interest. Autocorrelation serves as another foundational feature in forecasting. It quantifies the internal memory of a series by measuring the similarity between observations at different lags. The autocorrelation function directly characterizes the components identified by decomposition, such as slow decay for trends or rapid drop-off for noise. Consequently, autocorrelation statistics themselves are often used as a feature in time series forecasting models to capture these temporal dependencies.

Transformer is emerging in time series forecasting, driven by the tremendous success in the natural language processing field. Transformer has the ability to extract multi-level representations from sequences and illustrate pairwise connections (Liu et al., 2024). In time series data analysis, Transformers that combine frequency-domain filtering with temporal and channel-wise transformations offer a powerful approach to capturing complex dependencies and patterns. By leveraging frequency-domain filtering, these models can efficiently separate and analyze the signal's various frequency components, isolating key seasonal or trend-related elements (Zerveas et al., 2021). Additionally, temporal and channel-wise transformations allow the model to process both the sequential nature of time series data and the relationships across multiple channels, making it particularly adept at handling multivariate time series and discovering intricate dependencies between variables (Zhang et al., 2022a). This dual capability enables the model to address both local and global patterns, improving performance in tasks such as forecasting, anomaly detection, and classification. Our main contributions are as follows:

- The dynamic decomposition enhances the model's ability to capture intricate patterns, enabling it to learn the trend and seasonality features of each dataset without relying on fixed assumptions.
- By leveraging the concept of autocorrelation in the time domain, the architecture effectively incorporates temporal dependencies into the forecasting process. This integration allows the model to recognize and utilize historical relationships within the data, thereby improving forecasting accuracy.
- LMSAutoTSF achieves state-of-the-art (SOTA) performance in long-term forecasting tasks by integrating learnable decomposition, autocorrelation, and multi-scale processing.
- A confidence score, derived from reconstruction error, is integrated to represent the trustworthiness of the forecast. This score adjusts the final prediction, further enhancing the model's reliability.

We do comprehensive tests on different time-series forecasting benchmarks to support our motivation and hypothesis. When compared to various forecasting techniques, the performance of our proposed model is at the cutting edge. Extensive ablation investigations and analysis trials support the viability of suggested designs and their consequent benefit over earlier methods.

This paper is organized as follows. Section 2 explains a set of typical times series forecasting works from the literature which sets the notion and motivation of this paper. Section 3 briefly explains the problem statement of this paper. In Section 4, we provide a detailed discussion of our proposed architecture called LMSAutoTSF which is suitable for multivariate time series forecasting. In Section 5, we discuss the result analysis of the proposed methodology and describe a detailed comparison with the state-of-the-art time series forecasting models. A conclusion of the work's findings is provided in Section 6.

2. Literature review

In both long-term and short-term time series forecasting, Transformer-based architectures have demonstrated exceptional performance and significant potential (Zerveas et al., 2021; Z. Zhang et al., 2024; Nie et al., 2023; Zeng et al., 2023). This section reviews key developments in Transformer-based forecasting models, linear decomposition-based models, and MLP-based alternatives, positioning our proposed architecture within this evolving research landscape.

2.1. Transformer-based models for time series forecasting

The Transformer architecture has rapidly become a foundational model for time series forecasting due to its capacity to capture long-range dependencies and temporal dynamics. The Informer (Zhou et al., 2021) was the first widely recognized Transformer variant for time series forecasting, introducing a generative-style decoder and the ProbSparse self-attention mechanism to address the quadratic time complexity challenge inherent in standard Transformers. Following Informer, numerous Transformer variants were developed to improve efficiency and performance. Autoformer (Chen et al., 2021) incorporated auto-correlation and decomposition mechanisms to enhance long-term forecasting capabilities. Pyraformer (Liu et al., 2021) focused on multi-resolution attention for efficient signal processing, while FEDFormer (Zhou et al., 2022) integrated frequency-domain analysis with Transformer attention to improve representation learning. Reformer (Kitaev et al., 2020) further optimized attention through locality-sensitive hashing for memory efficiency. Recent studies such as iTransformer (Liu et al., 2024) and PatchTST (Nie et al., 2023) emphasize improved sequence representation and local–global pattern recognition. In particular, PatchTST enhances the model's capacity to capture hierarchical dependencies by leveraging patch-based embeddings. Similarly, Crossformer (Zhang and Yan, 2023) introduces a Dimension-Segment-Wise (DSW) embedding approach that converts multivariate time series into 2D vector arrays, preserving both temporal and dimensional relationships. Its Two-Stage Attention (TSA) layer effectively captures cross-time and cross-dimension dependencies, forming the basis for its Hierarchical Encoder-Decoder (HED) design. DyGraphformer (Han et al., 2025) combines Transformer architecture with a dynamic spatio-temporal graph network to model multivariate time series. By capturing both temporal dependencies and dynamic inter-variable relationships, DyGraphformer effectively represents complex interactions across multiple series. Its hybrid approach enhances forecasting accuracy for high-dimensional and highly interdependent datasets, demonstrating the benefits of integrating graph-based modeling with Transformer architectures for multivariate time series prediction. TOEformer (Li et al., 2025) enhances the Transformer by incorporating temporal order information into attention, improving long-term forecasting accuracy and robustness for multivariate time series. TimeXer (Y. Wang et al., 2024) is a Transformer-based model designed for the practical forecasting scenario where a target variable is influenced by auxiliary exogenous variables which are external factors that influence the system but are not influenced by them. It introduces a dual-attention mechanism, using patch-wise self-attention for temporal patterns and variate-wise cross-attention to inject relevant external information from exogenous series into the endogenous (core internal variables of a system) forecast.

2.2. Advances in multi-scale and non-stationary modeling

Recent architectures have emphasized multi-scale and non-stationary feature modeling to better capture complex temporal structures. N-HITS (Challu et al., 2023) adopts hierarchical learning and interpolation-based aggregation, leveraging multi-scale representations, a philosophy conceptually aligned with our proposed design. The Multi-resolution Time Series Transformer (MTST) (Y. Zhang et al., 2024) further extends

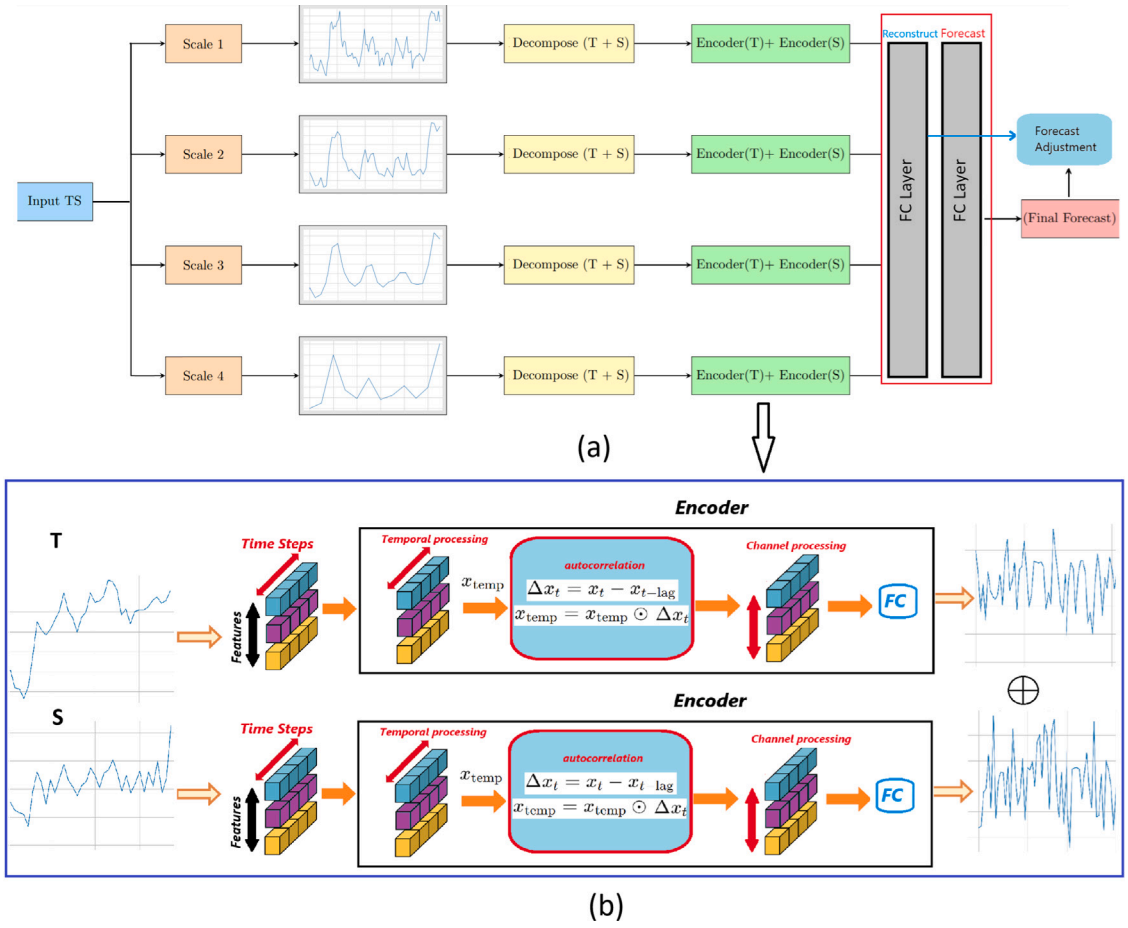


Fig. 1. (a) Overall architecture of LMSAutoTSF. (b) Encoder module operations for trend and seasonal components in Scale-2.

this idea with a multi-branch structure, modeling patterns at multiple resolutions using relative positional encoding. This enables effective representation of periodic components across scales (Woo et al., 2022). The Non-Stationary Transformer (Liu et al., 2022) addresses the challenge of changing statistical properties in time series, incorporating normalization techniques to manage evolving trends and seasonality. This adaptation enhances its ability to handle both short-term and long-term dependencies in dynamic environments, such as financial forecasting and climate prediction. MICN (Multi-scale Local and Global Context Network) (Wang et al., 2023) introduces an efficient framework for long-term time series forecasting by jointly modeling local and global temporal dependencies across multiple scales. The model integrates convolutional blocks for local context extraction and attention-based modules for capturing global dependencies, effectively balancing fine-grained short-term variations with broader long-term patterns. By employing multi-scale feature aggregation, MICN enhances forecasting robustness and adaptability across diverse datasets. Its design philosophy aligns with recent trends in hierarchical and multi-resolution modeling, demonstrating that combining local context and global temporal representations can significantly improve predictive accuracy.

2.3. MLP-based forecasting architectures

Beyond Transformers, MLP-based architectures have shown strong potential in time series forecasting. TSMixer (Ekambaram et al., 2023) replaces traditional recurrent or convolutional structures with token-mixing layers that capture dependencies across time and features. This simple yet effective architecture supports parallel computation, achieving competitive performance while maintaining efficiency. Similarly, TimeMixer (S. Wang et al., 2024) employs a multi-layer perceptron-based framework that leverages multi-scale time series decomposition in both past extraction and future prediction stages. Its Past-Decomposable-Mixing (PDM) and Future-Multipredictor-Mixing (FMM) components collaboratively model macroscopic trends and microscopic seasonal patterns, enabling robust multi-horizon forecasting. The core architectural philosophies of LMSAutoTSF and TimeMixer diverge significantly, despite their shared use of multi-scale decomposition. TimeMixer is built on a foundation of hierarchical processing. Its mixing blocks enable bottom-up seasonal mixing and top-down trend mixing, with learnable linear layers ensuring explicit communication and progressive aggregation of information across scales. LMSAutoTSF, on the other hand, adopts a more modular approach. It processes each scale separately using parameterized frequency-domain filters, decomposes them into seasonal and trend components, and then concatenates all outputs for a direct projection to the forecast horizon. Another distinctive feature of LMSAutoTSF is its reconstruction-based confidence scoring, which weights the final forecast. TimeMixer++ (Wang et al., 2025) extends the original TimeMixer framework into a more general and versatile architecture for universal time series predictive analysis. It introduces enhanced multi-scale mixing mechanisms capable of capturing temporal patterns across heterogeneous datasets and forecasting horizons. By unifying trend, seasonal, and irregular components within a common representation space, TimeMixer++ achieves superior generalization while maintaining computational efficiency. Its fully MLP-based design eliminates the need for explicit recurrence or convolution, positioning it as a strong, lightweight alternative to Transformer-based models for long-term time series forecasting. WP-Mixer (Murad et al., 2025) is an efficient multi-resolution mixing

architecture for long-term time series forecasting. It employs parallel token-mixing operations at different resolutions to capture both local and global temporal dependencies while maintaining computational efficiency. By integrating multi-scale features in a lightweight MLP-based framework, WPMixer achieves strong predictive performance across diverse datasets, demonstrating that carefully designed multi-resolution mixing can rival more complex Transformer-based approaches. TCM (Jiang et al., 2025) proposes a lightweight MLP-based architecture for long-term time series forecasting, incorporating affine transformations to enhance representation learning. Its design efficiently captures temporal dependencies while keeping computational costs low, making it suitable for large-scale and real-time applications. TCM demonstrates competitive performance compared to more complex Transformer-based models, highlighting the effectiveness of streamlined MLP architectures for long-horizon forecasting tasks.

2.4. Linear and decomposition-based methods

Decomposition remains a fundamental concept in time series analysis (Cleveland et al., 1990). Classical approaches decompose data into trend, seasonal, and residual components for interpretability and improved predictability. Models such as Prophet (Sean and Taylor, 2018), N-BEATS (Oreshkin et al., 2020), and DeepGLO (Sen et al., 2019) rely on explicit decomposition or matrix factorization to model historical dependencies before prediction (Asadi and Regan, 2020). However, such pre-processing often limits adaptability, as they cannot capture hierarchical interactions among temporal components in long-term forecasting. More recently, lightweight linear models like DLinear (Zeng et al., 2023) have revisited decomposition-based modeling. DLinear separates trend and seasonal components and models them with independent linear layers, achieving efficient training and competitive accuracy. These models often outperform more complex Transformer architectures in large-scale forecasting tasks due to their simplicity and computational efficiency. A work (Qiao et al., 2025) proposes a multivariate time series forecasting framework combining temporal decomposition with graph neural networks. The method captures both temporal patterns and inter-variable dependencies, improving prediction accuracy for complex, high-dimensional datasets while leveraging structured relationships among variables.

2.5. Positioning of the proposed work

Our proposed architecture builds upon these advancements by integrating learnable frequency-domain decomposition, autocorrelation-based temporal modeling, and multi-scale dual encoders. While conceptually related to models like Autoformer (Chen et al., 2021), FEDFormer (Zhou et al., 2022), and N-HITS (Challu et al., 2023), our method differs in several key aspects:

- It employs Fourier-based low-pass and high-pass filtering with learnable parameters for trend and seasonal decomposition.
- It incorporates autocorrelation explicitly through lag-based temporal modeling.
- It introduces a unique confidence scoring mechanism derived from reconstruction error to dynamically calibrate forecasts.

This combination of learnable filtering, autocorrelation modeling, and integrated confidence estimation enables our model to achieve superior forecasting accuracy while maintaining lightweight computational efficiency.

3. Problem statement

This paper addresses the challenge of long-term forecasting for multivariate time series, using historical data. We define a multivariate time series at time t as $\mathbf{X}_t = [x_{t,1}, x_{t,2}, \ldots, x_{t,N}]$, where $x_{t,n}$ represents the value of the nth variable at time t, for $n = 1, 2, \ldots, N$. We use the notation $X_{t:t+h}$ to represent the X-values from the time point t to $t + H$, inclusive. The aim is to develop a model for forecasting the future values of the series over the next T time steps, based on the most recent L time steps. The parameters L and H are referred to as the *look-back window* and the *prediction horizon*, respectively. Specifically, for a given initial time t_0, the model takes as input the sequence $\mathbf{X}_{t_0-L:t_0}$, corresponding to the past L time steps, and outputs the predicted sequence $\hat{\mathbf{X}}_{t_0:t_0+H}$, representing the forecasted values for the next H time steps. The predicted value of $\mathbf{X}_t$ at time t is denoted by $\hat{\mathbf{X}}_t$. In brief, the goal of multivariate time series forecasting is to predict future values $\mathbf{X}_{t:t+H}$ given past observations $\mathbf{X}_{t-L:t}$:

$$\hat{\mathbf{X}}_{t:t+H} = f(\mathbf{X}_{t-L:t}) \tag{1}$$

The forecasting performance of the model is assessed by computing the mean squared error (MSE) and the mean absolute error (MAE) between the prediction and the ground truth on the test set.

4. Methodology

In this work, we introduce an efficient approach for time series forecasting that leverages a combination of multi-scale input processing, frequency domain filtering (Yi et al., 2024), and autocorrelation. The general overview of the proposed architecture is represented in Fig. 1. To capture both local and global patterns in the time series, we apply multi-scale down-sampling of the input data. The input sequence is progressively down-sampled using an average pooling layer, with each scale indexed by k representing different temporal resolutions. This enables the model to capture variations across multiple time horizons. In the rest of the article, we use the abbreviation X_t^{LB} to represent $X_{t-L:t}$, where "LB" stands for "loopback".

$$\mathbf{X}_{t:t+H}^{(k)} = \text{Downsample}_k(\mathbf{X}_t^{LB}) \tag{2}$$

For each scale k, downsampled time series is decomposed into trend (T) and seasonality (S) components with learnable (differentiable) layers in the frequency domain. The decomposition is achieved by applying a frequency-domain transformation (via FFT), followed by learnable low-pass and high-pass filters. The cutoff frequency and steepness of the filters are trainable parameters, and different filters are learned for each feature of the time series. The filtering process is defined as:

- **Frequency transform**: Fast Fourier Transform (FFT) efficiently transforms a time series X_t from the time domain into its frequency domain representation $X(f)$, and vice versa, by decomposing the series into a sum of sinusoidal components.

$$\text{FFT}(X_{\text{LB}}) = \sum_{i=0}^{L-1} X_{t-L+i} \cdot e^{-j\frac{2\pi i}{L}} \tag{3}$$

$$X_{freq} = FFT(X_t^{LB}) \tag{4}$$

- **Low-pass filter**: Captures the trend **(T)** by retaining lower frequencies, defined as:

$$T = X_{\text{low}} = \text{FFT}^{-1}(X_{\text{freq}} \cdot \sigma(-(f - f_{\text{cutoff}}) \cdot s)) \tag{5}$$

where f is the frequency, f_{cutoff} is the learnable cutoff frequency, s is the steepness, and σ is the sigmoid function.

- **High-pass filter**: Extracts the seasonal component **(S)** by retaining higher frequencies, defined similarly:

$$S = X_{\text{high}} = \text{FFT}^{-1}(X_{\text{freq}} \cdot \sigma((f - f_{\text{cutoff}}) \cdot s)) \tag{6}$$

The low-pass filter is applied to emphasize low-frequency components (i.e., trend), while the high-pass filter is used to isolate high-frequency components (i.e., seasonality). This approach allows for the effective decomposition of the time series into its underlying trend and seasonal components, improving forecasting accuracy. This trainable decomposition approach enables the model to learn trend and seasonality

components directly from the data, rather than relying on a predefined method. This flexibility allows the model to capture patterns that might be missed by traditional techniques, resulting in more precise forecasts. Furthermore, this adaptability supports better generalization across different datasets, as the model can learn the unique trend and seasonality features of each dataset without depending on fixed assumptions.

$$\mathbf{X}_t^{LB}(k) = T^{(k)} + S^{(k)} \tag{7}$$

Encoder module details are represented in Fig. 1(b). In the Encoder module, we apply fully connected (FC) layers for temporal and channel processing. In addition, we apply autocorrelation via lagged difference, so the model takes this feature into account in addition to trend or season to emphasize how much temporal variation matter for forecasting. Autocorrelation in the time domain refers to the correlation between a time series and its lagged version, capturing how past values influence future values. A simple way to compute this is by taking the difference between a time series at time t and its value at a previous time $t-1$, which highlights how much the time series changes over consecutive steps. By incorporating autocorrelation in the model, the network can more effectively learn temporal dependencies, improving its ability to forecast future values based on past behavior. In the Encoder, firstly input (consider T for scale k: $T_t^{(k)}$) is passed through a temporal processing FC_{temp} layer, and then processed temporal data x_{temp} is multiplied by the autocorrelation which is the lagged difference of the input of the Encoder module as shown below.

$$x_{temp} = \mathrm{FC}_{\mathrm{temp}}(T^{(k)}) \tag{8}$$

$$x_{temp} = x_{temp} \odot \Delta T^{(k)} \tag{9}$$

where $\odot$ represents element-wise multiplication. Then, the channel processing operation is performed using the FC_{channel} layer to obtain x_{channel}, which captures feature interactions. This is followed by a projection layer $FC_{\mathrm{projection}}$ that produces the final forecasting outputs: $\hat{T}t^{(k)}$ for forecasting from T, and $\hat{S}t^{(k)}$ for forecasting from S.

$$\begin{aligned} x_{channel} &= \mathrm{FC}_{\mathrm{channel}}(x_{temp}) \\ \hat{T}^{(k)} &= \mathrm{FC}_{\mathrm{projection}}(x_{temp} + x_{channel}) \end{aligned} \tag{10}$$

The $Encoder$ module applying FC_{temp}, $FC_{channel}$ and $FC_{projection}$ is used to forecast from trend and seasonal components separately, and each forecast is summed to find the final forecast output of the corresponding scale as shown with $\oplus$ operation in Fig. 1. The forecasting outputs of encoder for trend and seasonality components are summed to form the final output for each scale:

$$\begin{aligned} \hat{T}^{(k)} &= \mathrm{Encoder}_T^{(k)}(T^{(k)}) \\ \hat{S}^{(k)} &= \mathrm{Encoder}_S^{(k)}(S^{(k)}) \end{aligned} \tag{11}$$

$$\hat{\mathbf{X}}_{t:t+H}^{(k)} = \hat{T}^{(k)} + \hat{S}^{(k)} \tag{12}$$

The final prediction $\hat{\mathbf{X}}_{t:t+H}$ is obtained by applying a final FC layer to the concatenated forecasts from all scales k:

$$\hat{\mathbf{X}}_{t:t+H} = \mathrm{FC}\left(\left[\hat{\mathbf{X}}_{t:t+H}^{(1)}\ \hat{\mathbf{X}}_{t:t+H}^{(2)}\ \cdots\ \hat{\mathbf{X}}_{t:t+H}^{(K)}\right]\right) \tag{13}$$

Where: $\left[\hat{\mathbf{X}}_{t:t+H}^{(1)}\ \hat{\mathbf{X}}_{t:t+H}^{(2)}\ \cdots\ \hat{\mathbf{X}}_{t:t+H}^{(K)}\right]$ represents the concatenation of forecasts from all scales, - K is the total number of scales.

4.1. Confidence score and forecast adjustment

To further enhance the reliability of the final forecast, we adjust it using a confidence score derived from the reconstruction error. First, reconstruction outputs from different scales are obtained and concatenated. Let the set of reconstruction outputs from the individual scales be denoted as $\{\mathbf{X}_{\mathrm{recon}}^{(k)}\}_{k=1}^{K}$. These outputs are concatenated along the scale dimension and this concatenated tensor is then transformed by a projection layer to align the feature dimensions:

$$\mathbf{X}_{\mathrm{recon}} = \mathrm{FC}\left(\left[\mathbf{X}_{\mathrm{recon}}^{(1)}, \mathbf{X}_{\mathrm{recon}}^{(2)}, \ldots, \mathbf{X}_{\mathrm{recon}}^{(K)}\right]\right) \tag{14}$$

Next, we compute the reconstruction error as the mean squared error between the encoded input x_{enc} and the reconstructed output $\mathbf{X}_{\mathrm{recon}}$:

$$\mathrm{rec_error} = \frac{1}{L}\sum_{i=1}^{L}\left(x_{\mathrm{enc}}^{(i)} - \mathbf{X}_{\mathrm{recon}}^{(i)}\right)^2, \tag{15}$$

A confidence score is then derived from the reconstruction error. This score, which reflects the trustworthiness of the forecast (with a lower error leading to higher confidence), is defined as:

$$\mathrm{confidence_score} = \exp(-\mathrm{rec_error}) \tag{16}$$

Finally, the initial forecast output $\hat{\mathbf{X}}_{t:t+H}$ is adjusted by an elementwise multiplication with the confidence score:

$$\hat{\mathbf{X}}_{t:t+H}^{\mathrm{adj}} = \mathrm{confidence_score} \odot \hat{\mathbf{X}}_{t:t+H}, \tag{17}$$

where $\odot$ denotes elementwise multiplication.

This reconstruction error based adjustment enables the model to modulate the final forecast according to its confidence, thereby enhancing overall prediction reliability.

5. Experiments

In this section, we perform in-depth experiments to assess how well our proposed LMSAutoTSF model performs in forecasting tasks, in comparison with state-of-the-art time-series forecasting architectures. The model is trained using a batch-based approach, and key hyperparameters are summarized in Table 1. Specifically, the model dimension (**d_model**) determines the capacity of the model to capture complex temporal patterns. Different values are selected based on task complexity: 512 for long-term forecasting (1024 for Traffic), 128 for short-term forecasting, and 32 for anomaly detection. We did not apply hyperparameter search for our own model to ensure a fair and reproducible comparison. For all baseline models, we adopt the hyperparameter configurations from their available code repositories. The **number of scales** ($k = 4$) facilitates multi-scale processing by enabling down-sampling at different levels to capture both short- and long-term dependencies. A **batch size** of 32 is used for forecasting tasks and 128 for anomaly detection, balancing training stability and generalization. The **learning rate** is set to 0.0001 for long-term forecasting and 0.003 for short-term and anomaly tasks, allowing for an optimal trade-off between convergence speed and stability. We employ **early stopping** with a **patience** of 3 epochs to prevent overfitting and limit the total **number of epochs** to 20 for all tasks. Additionally, input data is normalized across all models and datasets to ensure consistency and stabilize training.

All experiments are implemented in PyTorch (CUDA version 12.2) and conducted on a single NVIDIA GeForce RTX 3090 GPU with 24 GB memory. We replicate all baseline models using the benchmark Time-Series Library (TSLib) (Wu et al., 2023), following the official configurations or corresponding papers. The proposed LMSAutoTSF model is trained on all datasets using the specified settings, and we additionally perform ablation studies to measure the impact of the learnable decomposition and autocorrelation modules.

5.1. Datasets

We evaluate the performance of our proposed architecture on eight widely-used public benchmark datasets, including Weather, Electricity, Traffic, Exchange and four ETT datasets (ETTh1, ETTh2, ETTm1, ETTm2) for long-term forecasting. The PEMS dataset (Chen et al., 2001), which is used for short-term traffic flow forecasting, includes four public traffic network datasets: PEMS03, PEMS04, PEMS07, and

Table 1
Hyperparameters of LMSAutoTSF architecture and training.

Parameter	Forecasting: long-term	Forecasting: short-term	Anomaly
d_model	512 *except Traffic (1024)*	128	32
Number of scales (k)	4	4	4
Batch size	32	32	128
Learning rate	0.0001	0.003	0.003
Patience (early stopping)	3	3	3
Number of epochs	20	20	20

Table 2
Detailed dataset descriptions for forecasting and anomaly detection datasets.

Category	Dataset	Dim	Frequency	Information
Long-term	ETTh1, ETTh2	7	Hourly	Electricity
	ETTm1, ETTm2	7	15 min	Electricity
	Weather	21	10 min	Weather
	Electricity	321	Hourly	Electricity
	Traffic	862	Hourly	Transportation
	Exchange	8	Daily	Economy
	Solar-Energy	137	10 min	Electricity
	Beijing PM2.5	12	Hourly	Meteorological
Short-term	PEMS03	358	5 min	Transportation
	PEMS04	307	5 min	Transportation
	PEMS07	883	5 min	Transportation
	PEMS08	170	5 min	Transportation
Anomaly detection	MSL	55	1 min	Spacecraft telemetry
	PSM	25	1 min	Server machine data
	SMAP	25	1 min	Satellite telemetry
	SMD	38	1 min	Server machine data
	SWAT	51	1 min	Water treatment

PEMS08. The Beijing PM2.5 Dataset (Chen, 2015) is an hourly dataset containing PM2.5 air quality measurements recorded by the US Embassy in Beijing, alongside meteorological data from Beijing Capital International Airport, providing valuable insights into air quality and weather conditions. This dataset includes multiple features to predict one target feature, making it distinct from other datasets that focus on a single target. We assess anomaly detection performance on datasets: MSL (Hundman et al., 2018), SMAP (Hundman et al., 2018), SMD (Su et al., 2019), SWAT (Mathur and Tippenhauer, 2016), and PSM dataset (Abdulaal et al., 2021). The details of the datasets are shared in Table 2.

5.2. Forecasting results

Table 3 presents the ablation and sensitivity analysis of LMSAutoTSF across different datasets and architectural configurations. Each variant isolates the effects of fixed decomposition, learnable decomposition, autocorrelation, and the number of multi-scale branches (k). The combination of learnable decomposition and autocorrelation consistently achieves the best performance, demonstrating the model's ability to capture complex temporal dependencies. The detailed results in Tables 4 and 6 show that lower MSE/MAE values indicate higher precision, with LMSAutoTSF consistently outperforming state-of-the-art models such as TimeMixer, iTransformer, and PatchTST, particularly in high-dimensional long- and short-term forecasting. Fig. 2 shows sample long-term predictions for best four architectures: LMSAutoTSF, iTransformer, TimeMixer, and PatchTST. Fig. 4 shows the sample prediction on PEMS03 dataset, and it is seen that proposed LMSAutoTSF has closer predictions to the ground truth. More details for the PEMS prediction comparison are represented in the appendix. Table 5 presents an extensive comparative analysis of multivariate long-term forecasting performance across eight benchmark datasets (ETTh1, ETTh2, ETTm1, ETTm2, Weather, Electricity, Traffic, and Exchange) using recent state-of-the-art (SOTA) MLP-based models. The proposed LMSAutoTSF and its Confidence-adjusted variant are compared with DLinear (Zeng et al., 2023), MICN (Wang et al., 2023), LightTS (Zhang et al., 2022b), ATFNet (Ye et al., 2024), and TimeMixer++ (Wang et al., 2025) under fixed look-back (96) and multiple forecast horizons (96, 192, 336, 720). Evaluation metrics include MSE and MAE, with red and blue highlighting the best and second-best scores, respectively. Across datasets, LMSAutoTSF consistently outperforms competing methods, achieving the lowest overall average MSE (0.332) and MAE (0.345), followed closely by its confidence-adjusted variant (0.338/0.349). Traditional MLP-based models like DLinear and MICN exhibit higher error rates, while advanced architectures like ATFNet and TimeMixer++ perform competitively but still lag behind LMSAutoTSF. The model shows notable robustness across both short- and long-term horizons, particularly on ETTh1, ETTh2, and Weather datasets, confirming its superior generalization and stability in multivariate forecasting tasks. Fig. 3 shows sample long-term predictions for five sample MLP based architectures: LMSAutoTSF, MICN, TimeMixer++, and LightTS, on the ETTh1 dataset for a prediction horizon of 96 with an equal input sequence length of 96. The figure illustrates how closely each model's predicted values (typically represented by a colored line) follow the ground truth time series (often shown in black). Among all, LMSAutoTSF demonstrates the most accurate and stable predictions, maintaining tight alignment with the true signal and capturing both short-term fluctuations and long-term trends effectively. TimeMixer++ and LightTS perform reasonably well but exhibit noticeable deviations and smoother approximations in rapidly changing regions. In contrast, MICN and especially DLinear display larger phase shifts and higher amplitude errors, indicating poorer temporal sensitivity and trend adaptation. Overall, the visualization highlights the superior fitting ability and temporal consistency of LMSAutoTSF compared to other MLP-based baselines. For short-term forecasting performance, proposed method has competitive performance with TimeMixer, and outperforms iTransformer and PatchTST as shown in Table 6.Table 7 presents the performance comparison on the PM2.5 dataset for the multivariate input and single-feature detection problem. It can be observed that the proposed approach performs well in this scenario as well. The confidence score based adjustment not only enhances the reliability of our outputs but also delivers the best performance on the PM2.5 dataset, while achieving competitive results on other long-term and short-term forecasting benchmarks.

5.2.1. Model reliability and interpretability

Fig. 5 presents a comparison between the confidence scores and loss values for the 500 sequences of the Weather, and Solar datasets, respectively. Fig. 5(a) illustrates the relationship between prediction loss and confidence scores on the Weather dataset. Notably, a major loss spike occurs between timesteps 1400 and 1470, with values approaching 1.0, indicating a significant forecasting failure. During this period, confidence scores drop markedly (to 0.1–0.2), suggesting that the model correctly identifies this region as highly uncertain. Although the alignment between loss and confidence is not always exact, this is expected since loss measures prediction error on future targets, while confidence reflects reconstruction quality of the input or latent space. Consequently, confidence may lead or lag loss fluctuations. Rather than ensuring pointwise correspondence, the confidence score

Table 3
Ablation study for the average results of different prediction horizons.

Number of scales (k)	Fixed decompose	Learnable decompose	Auto correlation	Etth1		Ettm1		Weather		Electricity		PM2.5	
				MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
4	✓	–	–	0.457	0.440	0.398	0.406	0.242	0.273	0.180	0.279	0.616	0.509
4	–	✓	–	0.448	0.435	0.392	0.402	0.240	0.270	0.174	**0.271**	0.617	0.509
4	–	✓	✓	**0.441**	**0.432**	**0.377**	**0.394**	**0.238**	**0.269**	**0.175**	**0.271**	**0.615**	**0.508**
1	–	✓	✓	0.447	0.436	0.383	0.398	0.244	0.272	0.184	0.280	0.627	0.512

aims to highlight uncertain or challenging regions in forecasting. As shown in Fig. 5(b), notable loss spikes (e.g., around timesteps 1050 and 1200) coincide with decreased confidence, despite some volatility due to the dynamic nature of solar data. Overall, confidence scores approximate difficult forecasting intervals across datasets, offering a lightweight encoder-based uncertainty proxy without requiring ground truth. Table 8 presents the average confidence and loss values across seven widely used long-term forecasting datasets for both the test and validation splits with prediction length 96. As expected, higher confidence scores generally coincide with lower loss values, indicating that the learned confidence measure effectively reflects the model's predictive reliability. Across datasets, the inverse relationship between confidence and loss highlights that the confidence estimation captures uncertainty dynamics consistently with model performance, validating its interpretability as a measure of forecast certainty. Fig. 6 illustrates the learned frequency band masks across multiple features. Unlike a fixed spectral cutoff, the learned masks vary in both cutoff frequency and smoothness, indicating that the model adapts its spectral filtering based on each feature's temporal characteristics. The visualization applies frequency shifting, so the center corresponds to low frequency components. The sigmoid based masking mechanism ensures smooth and differentiable transitions between low and high frequency regions, enabling the model to softly control spectral emphasis rather than enforcing a hard cutoff. This design stabilizes training and allows flexible, feature-specific frequency selection for adaptive trend and seasonal decomposition. As observed, the learned frequency-band masks exhibit slight asymmetry around the zero frequency. This occurs because the sigmoid-based masking is applied over one-sided frequency indices before FFT shifting, leading to directionally smooth roll-offs on either side. The sigmoid ensures a continuous transition between low- and high-frequency components, avoiding abrupt spectral cutoffs.

Table 9 presents the comparison of average confidence and loss between test data and its Gaussian noise based out of distribution (OOD) variant. Although OOD perturbations typically increase the forecasting loss, the confidence score does not always decrease accordingly. In certain datasets, both loss and confidence increase simultaneously, indicating that the model can become overconfident under distributional shifts. This observation highlights the need for improved calibration mechanisms in confidence estimation while still demonstrating the sensitivity of the proposed score to OOD variations. Fig. 7 illustrates the trend and seasonal decomposition behavior across four representative samples. Each subplot compares fixed and learned cutoff settings, showing that the learned cutoffs adaptively adjust the separation between low and high frequency components. This flexibility enables the model to better capture varying temporal patterns and improves interpretability of the disentangled trend and seasonal dynamics.

5.3. Anomaly detection

For anomaly detection, we employ a reconstruction error-based approach. In time-series anomaly evaluation, point-based adjustments are commonly used. However, simple adjustments may artificially increase the accuracy without reflecting the actual performance. To ensure a rigorous evaluation of time-series anomaly detection, we implemented a coverage-based adjustment strategy instead of a simple threshold-based adjustment, ensuring a fair comparison. The coverage-based adjustment method refines anomaly predictions by evaluating the proportion of detected anomalies within true anomaly segments. If the predicted anomaly coverage is below a predefined threshold, adjustments are made accordingly. The details of this method are provided in Algorithm 1. This method ensures that predicted anomalies are validated by requiring a minimum overlap ratio with ground truth anomaly segments. Only predictions with sufficient coverage are considered correct, leading to a more stringent evaluation process. Additionally, we adapt the error feedback mechanism implemented in the AE-FAR (Delibasoglu and Heintz, 2024) for the anomaly detection case in our architecture. We define the F1 score after applying coverage-based adjustment as F_{1CPA}, with Precison and Recall denoted as P_{CPA} and R_{CPA}, respectively. Table 10 presents the results for public benchmark datasets, demonstrating the effectiveness of our evaluation method.

Algorithm 1 Coverage-based adjustment for anomaly detection

Require: y_{true}: Ground truth labels $(0, 1)$
Require: y_{pred}: Predicted labels $(0, 1)$
Require: coverage: Minimum coverage ratio (0.05)
Require: erase_tp: Whether to erase true positives outside adjusted segments (False)
Ensure: Adjusted predicted labels y_{adj}
$y_{adj} \leftarrow y_{pred}$
Identify anomaly segment boundaries from y_{true}
for each anomaly segment $(start, end)$ **do**
  Compute segment width: $w \leftarrow end - start$
  Compute predicted anomaly count: $c \leftarrow \sum_{i=start}^{end} y_{pred}[i]$
  Compute coverage ratio: $\rho \leftarrow c/w$
  if $\rho \geq$ coverage **then**
    Mark entire segment as anomaly: $y_{adj}[start : end] \leftarrow 1$
  else if erase_tp **then**
    Erase predictions outside valid coverage: $y_{adj}[start : end] \leftarrow 0$
  end if
end for
return y_{true}, y_{adj}

5.4. Model efficiency

Table 11 compares the execution time and computational complexity (measured in FLOPs) of several multivariate short-term forecasting models across four PEMS datasets. These datasets were specifically chosen for this comparison due to their high-dimensional input features, ranging from 170 to 883, which pose greater computational challenges. LMS-AutoTSF consistently achieves the lowest execution time across all datasets, outperforming TimeMixer, iTransformer, and PatchTST. It also requires the fewest FLOPs in three out of four cases, highlighting its computational efficiency. FLOPs were calculated using the *torch-profile* library, confirming LMS-AutoTSF's lightweight architecture and scalability for real-time, high-dimensional forecasting tasks.

Table 4
Comparison of multivariate long-term forecasting results with prediction horizons (96,192,336,720) and fixed look-back 96. The values highlighted in red and blue indicates the best and second-best scores for each metric, respectively.

Models		LMSAutoTSF (Confidence adj.)		LMSAutoTSF		TimeMixer		iTransformer		PatchTST		ETSformer		Autoformer	
Database	Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.379	0.398	0.383	0.397	0.378	0.399	0.386	0.404	0.386	0.405	0.521	0.477	0.505	0.482
	192	0.427	0.424	0.435	0.426	0.440	0.431	0.443	0.434	0.447	0.440	0.621	0.593	0.477	0.471
	336	0.462	0.442	0.469	0.442	0.499	0.459	0.492	0.463	0.489	0.473	0.721	0.692	0.525	0.507
	720	0.472	0.467	0.478	0.464	0.549	0.510	0.508	0.491	0.506	0.496	0.579	0.534	0.512	0.509
Avg		0.435	0.433	0.441	0.432	0.466	0.449	0.457	0.448	0.457	0.453	0.610	0.582	0.504	0.492
ETTh2	96	0.294	0.343	0.293	0.344	0.292	0.343	0.294	0.346	0.318	0.361	0.439	0.452	0.375	0.409
	192	0.368	0.391	0.368	0.393	0.383	0.403	0.379	0.398	0.389	0.407	0.369	0.436	0.463	0.461
	336	0.423	0.431	0.411	0.427	0.439	0.437	0.434	0.435	0.428	0.439	0.498	0.451	0.472	0.494
	720	0.435	0.448	0.432	0.447	0.441	0.450	0.467	0.466	0.439	0.454	0.459	0.481	0.481	0.489
Avg		0.380	0.403	0.376	0.403	0.389	0.408	0.394	0.411	0.394	0.415	0.441	0.455	0.447	0.463
ETTm1	96	0.327	0.360	0.318	0.357	0.323	0.362	0.341	0.376	0.324	0.364	0.201	0.269	0.539	0.494
	192	0.356	0.376	0.357	0.378	0.368	0.386	0.381	0.395	0.345	0.386	0.293	0.342	0.566	0.511
	336	0.390	0.401	0.385	0.403	0.401	0.406	0.419	0.418	0.395	0.408	0.337	0.391	0.621	0.534
	720	0.449	0.437	0.450	0.441	0.431	0.441	0.486	0.455	0.397	0.409	0.388	0.435	0.561	0.515
Avg		0.381	0.393	0.377	0.394	0.380	0.398	0.406	0.411	0.365	0.391	0.304	0.359	0.571	0.513
ETTm2	96	0.172	0.255	0.171	0.254	0.175	0.256	0.184	0.266	0.186	0.268	0.189	0.280	0.295	0.337
	192	0.237	0.299	0.236	0.298	0.238	0.299	0.252	0.311	0.253	0.311	0.253	0.319	0.292	0.349
	336	0.298	0.338	0.295	0.337	0.299	0.339	0.313	0.349	0.324	0.358	0.314	0.357	0.329	0.365
	720	0.397	0.396	0.406	0.402	0.401	0.405	0.409	0.405	0.406	0.402	0.414	0.413	0.436	0.424
Avg		0.276	0.322	0.277	0.323	0.278	0.325	0.290	0.333	0.292	0.335	0.293	0.342	0.338	0.368
Weather	96	0.155	0.202	0.154	0.202	0.162	0.208	0.179	0.219	0.175	0.218	0.221	0.286	0.319	0.369
	192	0.201	0.244	0.202	0.247	0.207	0.251	0.225	0.258	0.222	0.257	0.269	0.302	0.295	0.353
	336	0.259	0.288	0.259	0.288	0.263	0.292	0.279	0.298	0.279	0.298	0.271	0.334	0.346	0.385
	720	0.338	0.340	0.338	0.340	0.343	0.344	0.328	0.351	0.353	0.346	0.294	0.357	0.559	0.524
Avg		0.238	0.269	0.238	0.269	0.244	0.274	0.253	0.282	0.257	0.280	0.264	0.320	0.379	0.407
Electricity	96	0.156	0.253	0.154	0.252	0.158	0.249	0.149	0.240	0.180	0.273	0.187	0.304	0.216	0.335
	192	0.167	0.261	0.165	0.260	0.169	0.259	0.164	0.255	0.194	0.289	0.199	0.315	0.274	0.363
	336	0.178	0.274	0.177	0.274	0.189	0.281	0.183	0.275	0.217	0.322	0.212	0.329	0.243	0.353
	720	0.202	0.295	0.204	0.299	0.228	0.314	0.228	0.312	0.258	0.352	0.233	0.345	0.287	0.372
Avg		0.176	0.271	0.175	0.271	0.186	0.276	0.181	0.270	0.212	0.309	0.208	0.323	0.255	0.355
Traffic	96	0.475	0.322	0.485	0.330	0.469	0.291	0.417	0.287	0.526	0.347	0.607	0.392	0.668	0.400
	192	0.481	0.319	0.483	0.322	0.480	0.299	0.432	0.292	0.522	0.332	0.621	0.399	0.658	0.417
	336	0.496	0.325	0.493	0.323	0.498	0.301	0.449	0.305	0.531	0.339	0.622	0.396	0.667	0.414
	720	0.527	0.338	0.524	0.336	0.506	0.313	0.481	0.318	0.552	0.352	0.632	0.396	0.650	0.403
Avg		0.495	0.326	0.496	0.328	0.488	0.301	0.445	0.301	0.533	0.343	0.621	0.396	0.661	0.408
Exchange	96	0.081	0.198	0.082	0.199	0.088	0.208	0.098	0.226	0.097	0.216	0.085	0.204	0.149	0.279
	192	0.171	0.293	0.174	0.296	0.184	0.307	0.188	0.312	0.182	0.303	0.182	0.303	0.285	0.389
	336	0.321	0.409	0.324	0.410	0.376	0.442	0.337	0.422	0.346	0.425	0.348	0.428	0.969	0.728
	720	0.826	0.683	0.834	0.686	0.880	0.701	0.885	0.714	0.950	0.731	1.025	0.774	1.112	0.820
Avg		0.350	0.396	0.354	0.398	0.382	0.415	0.377	0.419	0.394	0.419	0.410	0.427	0.628	0.554
Solar	96	0.213	0.265	0.214	0.264	0.259	0.293	0.228	0.271	0.236	0.287	0.573	0.595	0.573	0.512
	192	0.247	0.287	0.251	0.287	0.276	0.312	0.267	0.293	0.271	0.303	0.652	0.660	0.725	0.632
	336	0.263	0.298	0.270	0.299	0.296	0.326	0.286	0.310	0.285	0.318	0.753	0.707	0.751	0.649
	720	0.265	0.297	0.269	0.302	0.297	0.318	0.301	0.320	0.283	0.317	0.761	0.718	0.838	0.706
Avg		0.247	0.287	0.251	0.288	0.282	0.312	0.271	0.299	0.269	0.306	0.685	0.670	0.722	0.625
Overall Avg		0.338	0.349	0.332	0.345	0.344	0.351	0.341	0.353	0.353	0.361	0.426	0.430	0.501	0.465

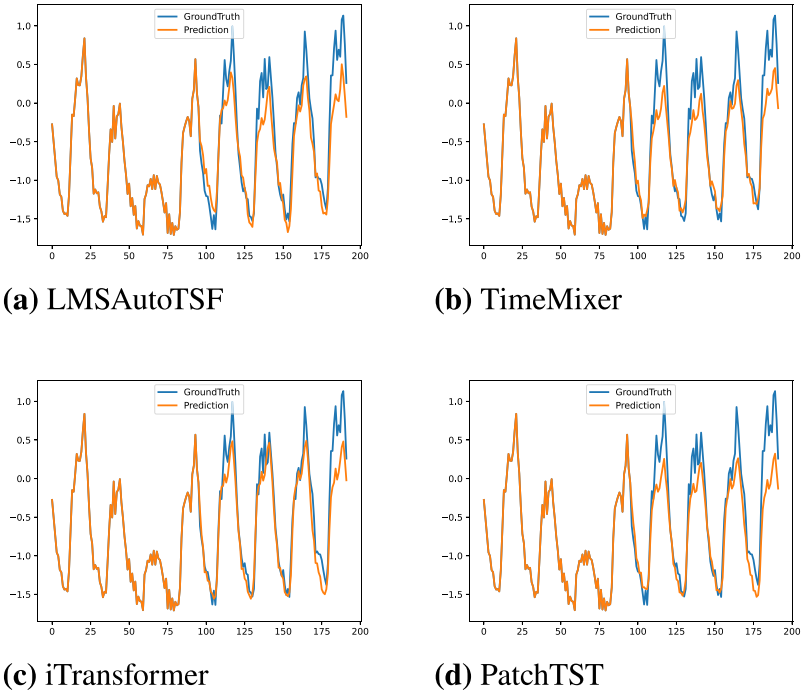


Fig. 2. Visualization of prediction results on Electricity dataset.

Table 5
Extended comparison of multivariate long-term forecasting results with recent SOTA MLP based models in terms of prediction horizons (96,192,336,720) and fixed look-back 96. The values highlighted in red and blue indicates the best and second-best scores for each metric, respectively.

Models		LMSAutoTSF		WPMixer(Murad et al., 2025)		DLinear (Zeng et al., 2023)		MICN(Wang et al., 2023)		LightTS(Zhang et al., 2022b)		ATFNet(Ye et al., 2024)		TimeMixer++(Wang et al., 2025)	
Database	Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.383	0.397	0.387	0.399	0.396	0.411	0.413	0.442	0.424	0.432	0.418	0.442	0.385	0.403
	192	0.435	0.426	0.426	0.424	0.445	0.440	0.451	0.462	0.475	0.462	0.487	0.493	0.442	0.430
	336	0.469	0.442	0.488	0.451	0.487	0.465	0.556	0.528	0.518	0.488	0.516	0.522	0.512	0.470
	720	0.478	0.464	0.511	0.484	0.513	0.510	0.658	0.607	0.547	0.533	0.640	0.597	0.490	0.472
Avg		**0.441**	**0.432**	**0.452**	**440**	**0.460**	**0.456**	**0.520**	**0.509**	**0.491**	**0.479**	**0.515**	**0.514**	**0.457**	**0.444**
ETTh2	96	0.293	0.344	0.300	0.350	0.341	0.395	0.303	0.364	0.397	0.437	0.303	0.398	0.290	0.343
	192	0.368	0.393	0.372	0.395	0.482	0.479	0.403	0.446	0.520	0.504	0.400	0.451	0.382	0.402
	336	0.411	0.427	0.425	0.432	0.593	0.542	0.603	0.550	0.626	0.559	0.493	0.511	0.416	0.431
	720	0.432	0.447	0.434	0.449	0.840	0.661	1.106	0.852	0.863	0.672	0.451	0.489	0.419	0.441
Avg		**0.376**	**0.403**	**0.383**	**0.407**	**0.564**	**0.519**	**0.604**	**0.553**	**0.602**	**0.543**	**0.412**	**0.462**	**0.377**	**0.404**
ETTm1	96	0.318	0.357	0.330	0.364	0.346	0.374	0.375	0.427	0.374	0.400	0.376	0.402	0.326	0.363
	192	0.357	0.378	0.385	0.397	0.382	0.391	0.389	0.435	0.400	0.407	0.427	0.432	0.369	0.388
	336	0.385	0.403	0.397	0.407	0.415	0.415	0.451	0.469	0.438	0.438	0.480	0.464	0.401	0.408
	720	0.450	0.441	0.486	0.542	0.472	0.451	0.480	0.476	0.527	0.502	0.551	0.513	0.468	0.442
Avg		**0.377**	**0.393**	**0.400**	**0.428**	**0.403**	**0.407**	**0.424**	**0.452**	**0.435**	**0.437**	**0.408**	**0.453**	**0.391**	**0.400**
ETTm2	96	0.171	0.254	0.176	0.257	0.193	0.293	0.191	0.295	0.209	0.308	0.201	0.318	0.175	0.258
	192	0.236	0.298	0.241	0.300	0.284	0.361	0.262	0.339	0.311	0.382	0.265	0.369	0.239	0.301
	336	0.295	0.337	0.303	0.341	0.385	0.429	0.311	0.379	0.442	0.466	0.330	0.394	0.295	0.339
	720	0.406	0.402	0.391	0.392	0.556	0.523	0.431	0.448	0.675	0.587	0.472	0.451	0.394	0.397
Avg		**0.277**	**0.323**	**0.278**	**0.323**	**0.354**	**0.401**	**0.298**	**0.365**	**0.409**	**0.436**	**0.317**	**0.383**	**0.276**	**0.324**
Weather	96	0.154	0.202	0.162	0.208	0.196	0.256	0.178	0.249	0.182	0.242	0.172	0.228	0.167	0.212
	192	0.202	0.247	0.207	0.250	0.239	0.299	0.243	0.269	0.227	0.287	0.220	0.269	0.208	0.252
	336	0.259	0.288	0.262	0.289	0.281	0.331	0.278	0.338	0.282	0.334	0.281	0.319	0.264	0.294
	720	0.338	0.340	0.343	0.345	0.345	0.382	0.320	0.360	0.352	0.386	0.354	0.405	0.343	0.344
Avg		**0.238**	**0.269**	**0.243**	**0.273**	**0.265**	**0.317**	**0.255**	**0.304**	**0.261**	**0.312**	**0.256**	**0.306**	**0.245**	**0.275**
Electricity	96	0.154	0.252	0.166	0.258	0.199	0.287	0.190	0.299	0.207	0.307	0.205	0.319	0.153	0.245
	192	0.165	0.260	0.178	0.267	0.201	0.290	0.209	0.328	0.213	0.316	0.220	0.338	0.167	0.256
	336	0.177	0.274	0.188	0.275	0.212	0.305	0.230	0.345	0.230	0.333	0.229	0.349	**0.186**	**0.275**
	720	0.204	0.299	0.221	0.312	0.248	0.338	0.265	0.378	0.265	0.360	0.261	0.371	0.223	0.319
Avg		**0.175**	**0.271**	**0.188**	**0.278**	**0.215**	**0.305**	**0.224**	**0.337**	**0.229**	**0.329**	**0.229**	**0.344**	**0.182**	**0.274**
Traffic	96	0.485	0.330	0.521	0.345	0.696	0.428	0.473	0.293	0.615	0.319	0.478	0.310	0.481	0.316
	192	0.483	0.322	0.525	0.349	0.650	0.409	0.483	0.298	0.601	0.382	0.498	0.321	0.491	0.317
	336	0.493	0.323	0.539	0.352	0.657	0.412	0.491	0.303	0.613	0.386	0.513	0.330	0.514	0.322
	720	0.524	0.336	0.552	0.365	0.679	0.428	0.559	0.327	0.658	0.407	0.529	0.345	0.530	0.327
Avg		**0.496**	**0.328**	**0.534**	**0.353**	**0.670**	**0.419**	**0.501**	**0.305**	**0.622**	**0.392**	**0.504**	**0.326**	**0.504**	**0.321**
Exchange	96	0.082	0.199	0.083	0.200	0.098	0.232	0.278	0.315	0.116	0.262	0.114	0.251	0.085	0.214
	192	0.174	0.296	0.180	0.299	0.186	0.325	0.392	0.441	0.215	0.359	0.465	0.511	0.175	0.313
	336	0.324	0.410	0.354	0.429	0.327	0.434	0.551	0.592	0.377	0.466	0.559	0.603	0.316	0.420
	720	0.834	0.686	0.885	0.709	0.749	0.663	0.612	0.683	0.831	0.699	0.609	0.711	0.851	0.689
Avg		**0.251**	**0.288**	**0.376**	**0.409**	**0.282**	**0.312**	**0.271**	**0.299**	**0.269**	**0.306**	**0.685**	**0.670**	**0.722**	**0.625**

Table 6
Comparison of multivariate short-term forecasting results with different prediction horizons (12, 24, 48) and fixed look-back 96. The values highlighted in red and blue indicates the best and second-best scores for each metric, respectively.

Models		LMSAutoTSF Confidence adj. (ours)		LMSAutoTSF (ours)		TimeMixer		iTransformer		PatchTST	
Dataset	Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
PEMS03	12	0.0640	0.1683	0.0647	0.1706	0.0607	0.1659	0.0732	0.1804	0.0915	0.2018
	24	0.0964	0.2087	0.0776	0.1898	0.0822	0.1939	0.1013	0.2139	0.1457	0.2573
	48	0.1523	0.2631	0.1022	0.2191	0.0982	0.2138	0.1663	0.2789	0.3014	0.3791
	Avg	0.1042	0.2134	0.0815	0.193	0.0803	0.1912	0.1136	0.2244	0.179	0.2794
PEMS04	12	0.0718	0.1742	0.0664	0.1681	0.0664	0.1689	0.0868	0.1925	0.1075	0.2262
	24	0.0903	0.1996	0.0749	0.1813	0.0746	0.1803	0.1221	0.2313	0.1725	0.2817
	48	0.1241	0.2406	0.0862	0.1975	0.0905	0.2039	0.1935	0.2977	0.3845	0.4423
	Avg	0.0954	0.2048	0.0758	0.1823	0.0771	0.184	0.134	0.2405	0.2215	0.316
PEMS07	12	0.0598	0.1570	0.0565	0.1582	0.0555	0.1538	0.0711	0.1739	0.098	0.217
	24	0.0812	0.1818	0.0715	0.1815	0.0661	0.1688	0.1039	0.2100	0.1486	0.2628
	48	0.1243	0.2310	0.0928	0.2091	0.0886	0.2013	0.1656	0.2695	0.3215	0.3941
	Avg	0.0884	0.1899	0.0736	0.1829	0.070	0.174	0.113	0.2178	0.189	0.2913
PEMS08	12	0.760	0.1791	0.064	0.1699	0.0617	0.1647	0.0817	0.1881	0.1492	0.2125
	24	0.1113	0.2192	0.0798	0.1948	0.0779	0.1872	0.1220	0.2347	0.1639	0.2798
	48	0.1740	0.2693	0.1039	0.2270	0.1015	0.2197	0.2217	0.3279	0.3126	0.3934
	Avg	0.1204	0.2225	0.082	0.197	0.080	0.190	0.1418	0.250	0.208	0.295
Overall Avg		0.1021	0.2077	0.0782	0.1888	0.0769	0.1848	0.1256	0.2332	0.1994	0.2954

Table 7
Comparison of models on PM2.5 Dataset with multivariate input single target. (prediction horizons (12,24,48) and fixed look-back 24). The values highlighted in red and blue indicates the best and second-best scores for each metric, respectively.

	LMSAutoTSF Confidence adj. (ours)		LMSAutoTSF (ours)		TimeMixer		iTransformer		PatchTST	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
12	0.3607	0.3734	0.3656	0.3780	0.3731	0.3791	0.4108	0.4024	0.4202	0.4093
24	0.5900	0.5006	0.5891	0.5020	0.6402	0.5254	0.6510	0.5294	0.6586	0.5275
48	0.8887	0.6439	0.8911	0.6443	0.9271	0.6623	0.9767	0.6751	0.9753	0.6728
Avg.	0.6131	0.5060	0.6152	0.5081	0.6468	0.5223	0.6795	0.5356	0.6847	0.5365

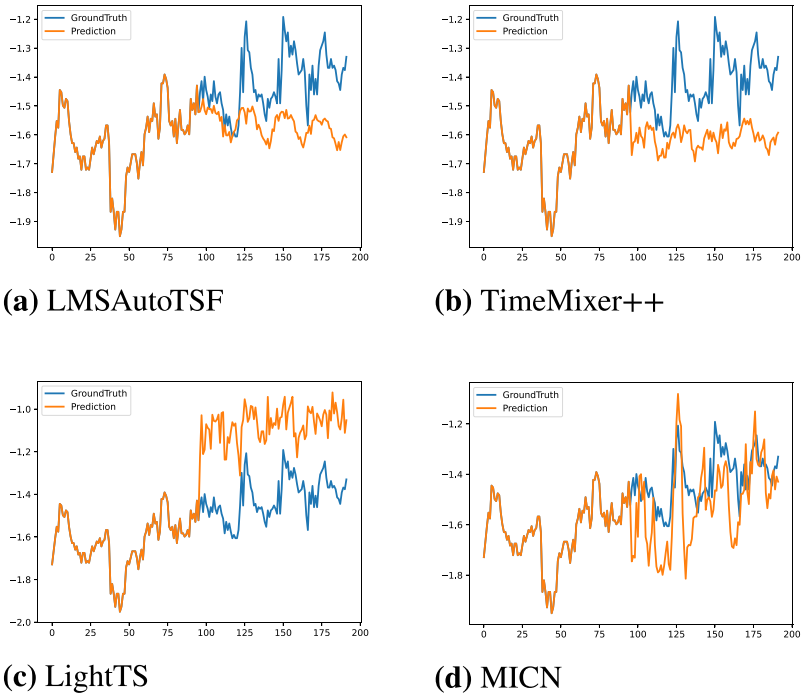


Fig. 3. Visualization of prediction results on ETTh1 dataset (prediction length:96, sequence length:96) with MLP based models.

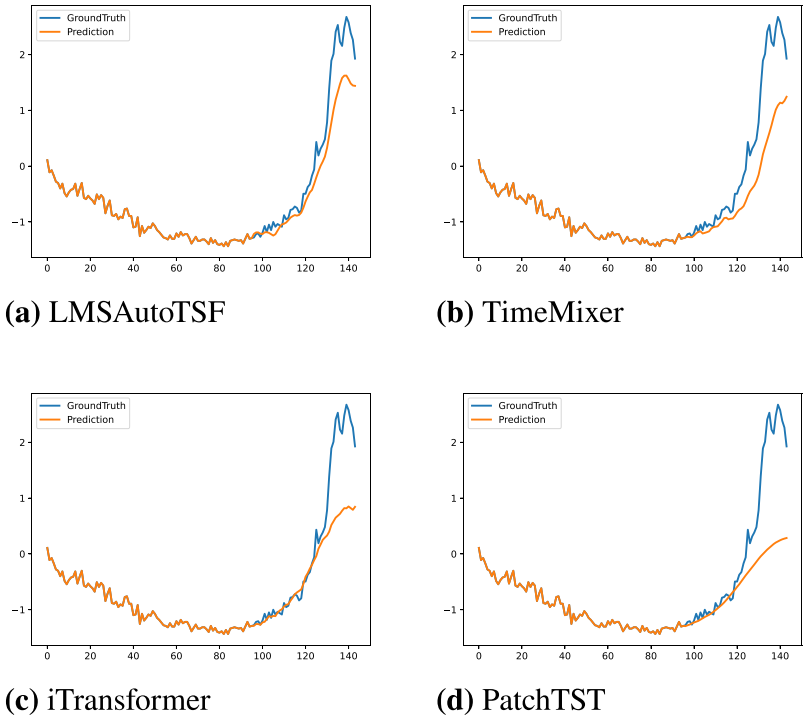
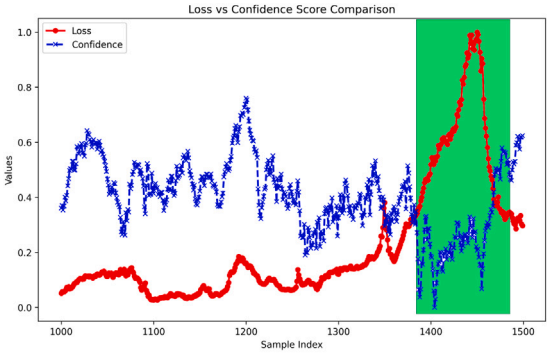


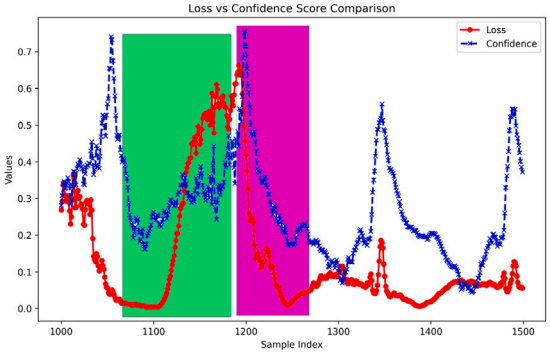
Fig. 4. Visualization of prediction results on PEMS03 dataset.

6. Conclusions

In this work, we introduced LMSAutoTSF, a novel and efficient architecture for time series forecasting that unifies learnable frequency-domain decomposition, multi-scale dual encoders, and autocorrelation-based temporal modeling. Unlike conventional methods that rely on predefined assumptions for trend and seasonality, LMSAutoTSF leverages learnable low-pass and high-pass filters to dynamically extract trend and seasonal components directly in the frequency domain. This allows the model to flexibly adapt to various time series characteristics across different temporal scales. A central feature of LMSAutoTSF is the explicit integration of autocorrelation, a fundamental property of time



(a) Weather dataset: High-loss regions (e.g., timesteps between 1400 and 1470) generally align with low confidence. A temporal lag arises as loss reflects future prediction error, while confidence is reconstruction-based.



(b) Solar dataset: The green-shaded region shows low confidence preceding high loss, while the purple region shows confidence rising after loss decreases - highlighting the temporal lag between the two.

Fig. 5. Comparison of confidence scores and prediction loss on the Weather and Solar datasets. In both cases, the confidence score exhibits a lag relative to the loss, reflecting its reconstruction-based nature versus the loss's future-oriented objective.

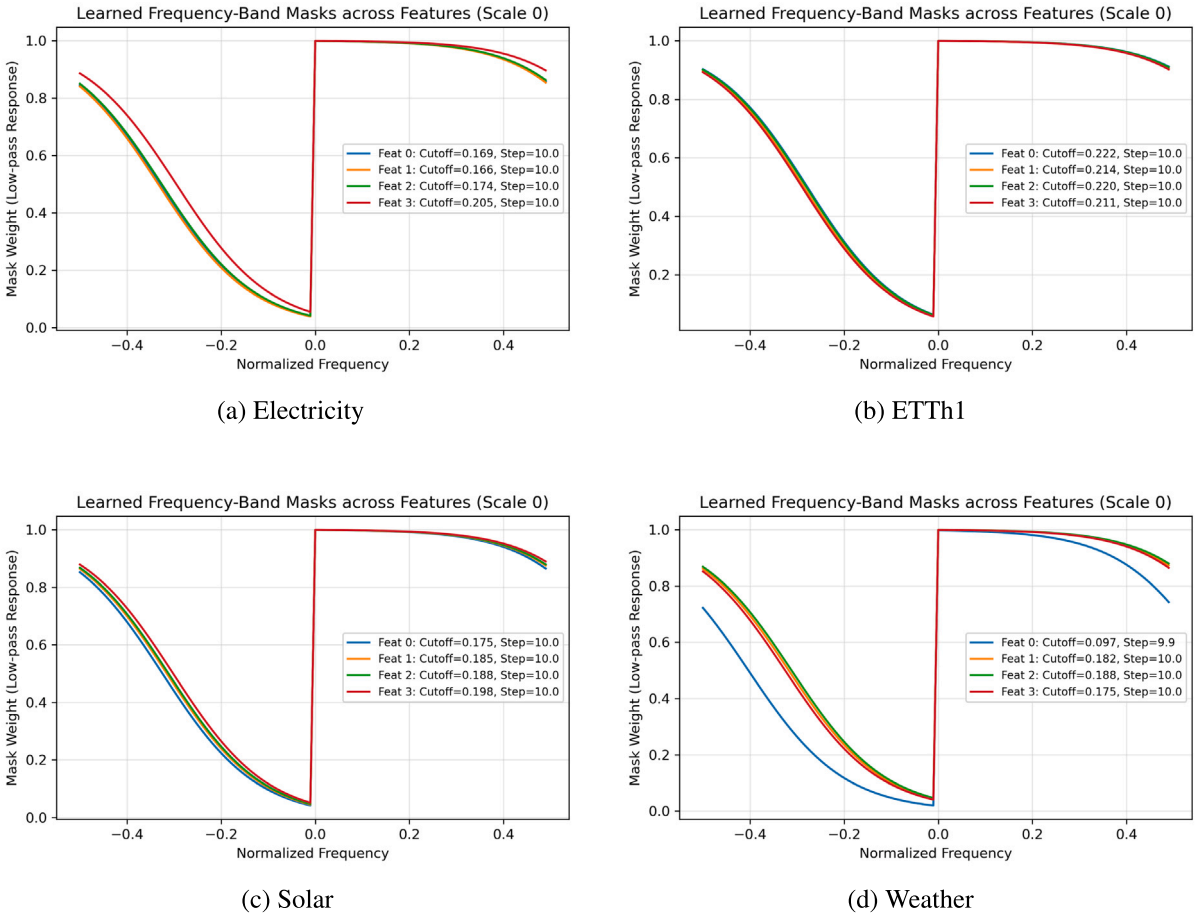
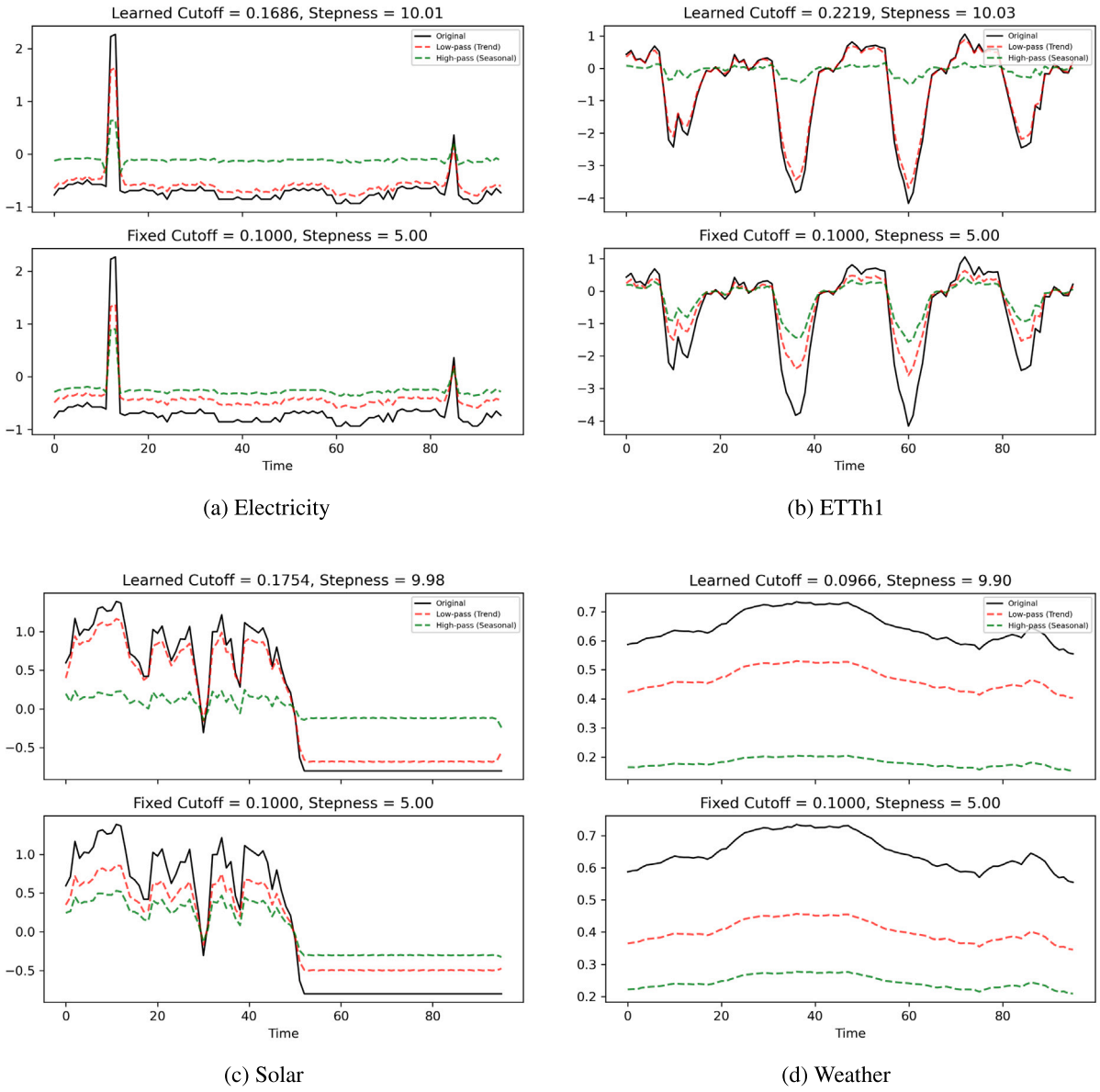


Fig. 6. Visualization of learned frequency-band masks across different features on different datasets.

series data, into the temporal encoding process. By computing lagged differences and incorporating skip connections, the model captures time step dependencies more effectively while preserving important temporal information throughout the network. This autocorrelation-aware design significantly improves the model's ability to forecast complex patterns over extended horizons.

Moreover, we propose a confidence score, derived from the reconstruction error, to assess the reliability of each forecast. This score enhances the model's transparency and enables uncertainty-aware predictions, which are especially valuable in high-stakes applications. To the best of our knowledge, LMSAutoTSF is the only state-of-the-art method offering such an informative reliability measure, further enhancing both the transparency and robustness of predictions. However, the current mechanism may be less effective on extremely volatile datasets where uncertainty is inherently high. Future work will focus on developing adaptive methods, such as dynamic confidence



(a) Electricity (b) ETTh1

(c) Solar (d) Weather

Fig. 7. Visualization of trend and seasonal decomposition for four representative samples. Each subplot shows the original signal, the low-pass (trend) and high-pass (seasonal) components, using both learned and fixed cutoff frequencies. Learned cutoffs adaptively separate frequency bands based on data characteristics, yielding more interpretable and distinct trend and seasonal representations.

Table 8
Comparison of average confidence and loss values for the test and validation sets across seven benchmark datasets in long-term forecasting. Higher confidence values indicate stronger model certainty, while lower loss values correspond to better predictive accuracy. An opposite trend between confidence and loss within the same dataset reflects the consistency of the confidence estimation with model performance.

Dataset	Avg Conf. test	Avg. Loss	Avg. Conf. validation	Avg. Loss
ETTh1	0.6254	0.3793	0.6350	0.6834
ETTh2	0.6419	0.2939	0.6297	0.2163
ETTm1	0.6488	0.3272	0.6556	0.4050
ETTm2	0.6302	0.1717	0.5998	0.1261
Weather	0.6191	0.1547	0.6133	0.4025
Electricity	0.6695	0.1561	0.6705	0.1379
Solar	0.5636	0.2137	0.5386	0.1698

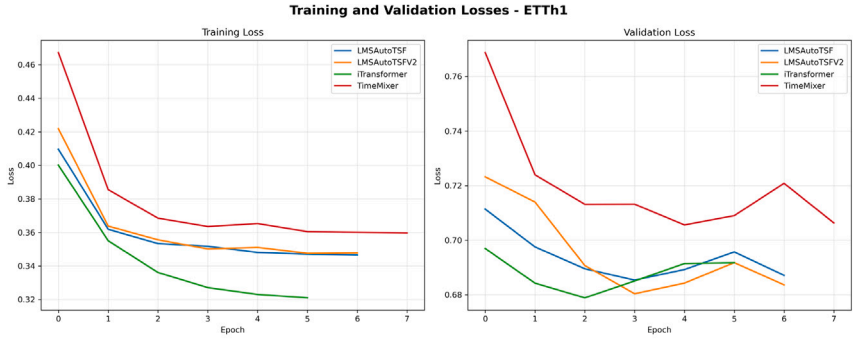
thresholding, to better mitigate the impact of uncertain predictions in such challenging regimes.

Funding

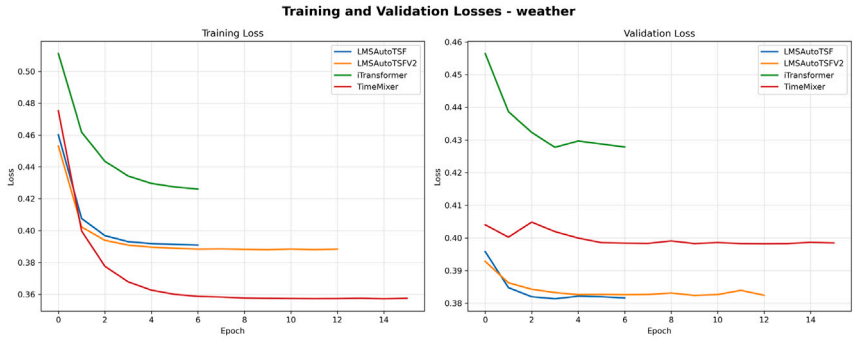
This work is partially supported by the 'Resurssmarta Processor (RSP)', the Wallenberg AI, and Autonomous Systems and Software Program (WASP), all funded by the Knut and Alice Wallenberg Foundation. We would also like to thank Vinnova and BioInnovation program.

CRediT authorship contribution statement

Ibrahim Delibasoglu: Writing – review & editing, Writing – original draft, Validation, Software, Investigation, Conceptualization. **Sanjay Chakraborty:** Writing – original draft, Validation. **Fredrik Heintz:** Writing – review & editing, Validation, Supervision.



(a) ETTh1 dataset



(b) ETTh2 dataset

Fig. 8. Training and validation loss convergence.

Table 9
Comparison of average confidence scores and forecasting losses on in-distribution test data and OOD inputs corrupted with Gaussian noise across seven benchmark datasets. While the loss generally increases under OOD perturbation, the behavior of the confidence score differs across datasets. In some cases (ETTh2, ETTm1, Electricity), confidence slightly decreases as expected under noise corruption, whereas in others (e.g., ETTh1, Solar) confidence increases despite higher loss. This reflects a well-known mismatch between error and confidence under distribution shift: the model remains overconfident on certain corrupted inputs even when its predictive accuracy degrades. Nevertheless, the loss remains stable without sharp degradation, indicating baseline OOD robustness, while also highlighting the potential value of extending the model with better-calibrated uncertainty estimation in future work.

Dataset	Test data		OOD with Gaussian noise	
	Avg Conf.	Avg. Loss	Avg. Conf.	Avg. Loss
ETTh1	0.4515	0.3793	0.5065	0.3936
ETTh2	0.4425	0.2939	0.4310	0.3034
ETTm1	0.5402	0.3272	0.4827	0.3383
ETTm2	0.3868	0.1717	0.4548	0.1912
Weather	0.5936	0.1547	0.6194	0.1818
Electricity	0.6070	0.1561	0.5920	0.1840
Solar	0.3844	0.2137	0.5106	0.3183

Declaration of competing interest

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

Appendix. Training dynamics and convergence analysis

This appendix provides a detailed analysis of the training behavior, convergence patterns, and optimization stability of the proposed LMSAutoTSF framework compared to other baseline models. The experiments and results shown here help us understand important aspects of model training that are crucial for both practical use and theoretical understanding of deep learning models for time series forecasting.

While the main paper focuses on forecasting accuracy and computational performance, this additional analysis looks at the fundamental training properties that support model performance. Specifically, we examine how gradients flow during training, how parameters settle to their final values, and how stable the training process remains across different datasets and model versions. This type of analysis is especially important for models that include learnable components in non-standard areas, like our frequency-adaptive filters, where we need to ensure they train properly. Our investigation covers two main areas: first, we compare training efficiency by looking at loss convergence between our methods and other state-of-the-art models; second, we carefully study the learnable frequency filter components to confirm they optimize stably and adapt meaningfully to different time series patterns. This two-part approach ensures that our new architecture not only delivers good results but also maintains reliable training behavior needed for real-world applications. The results presented in this appendix support the claims in the main paper about model stability and generalization ability, providing solid evidence for the reliability of our approach. By systematically analyzing multiple datasets, we show that our framework maintains stable training while effectively adapting to different time series characteristics, confirming both its practical value and mathematical soundness. Training analysis was conducted across multiple datasets including ETTh1, ETTh2, Weather and Solar to evaluate:

- Training and validation loss convergence for proposed and selected best models
- Parameter convergence and gradient propagation of cutoff frequencies and stepness parameters inside learnable frequency filters

A.1. Loss convergence analysis

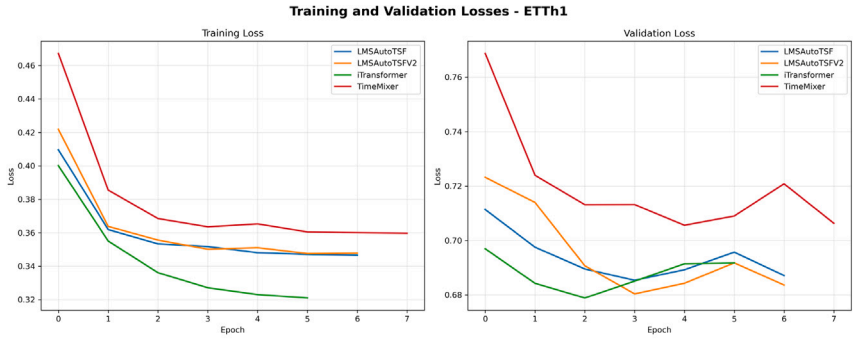
Figs. 8 and 9 present training and validation loss during training across multiple datasets including ETTh1, ETTh2, Weather, and Solar. Training analysis was conducted to evaluate both training/validation loss convergence for our proposed models and selected baselines,

Table 10
Anomaly detection performance with coverage-based point adjustment and fixed threshold. The values highlighted in red and blue indicates the best and second-best scores for each metric, respectively.

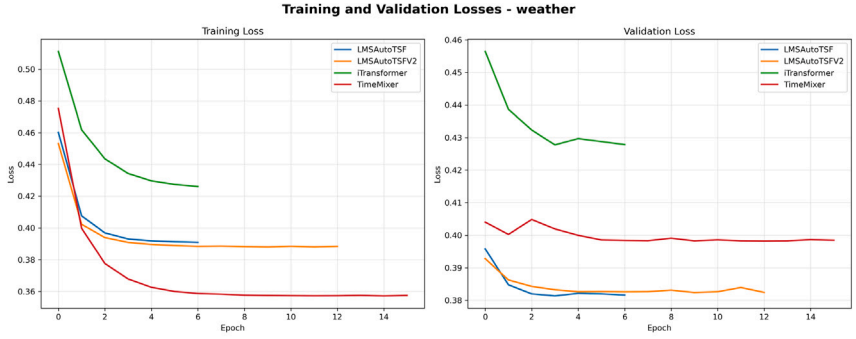
Dataset	Metric	LMSAutoTSF (MSE-RNN)	LMSAutoTSF (ours)	TimeMixer	iTransformer	PatchTST
MSL	P_{CPA}	0.4890	0.4942	0.4821	0.4569	0.4846
	R_{CPA}	0.1017	0.0979	0.0851	0.0801	0.0876
	F_{1CPA}	0.1684	0.1634	0.1446	0.1363	0.1484
PSM	P_{CPA}	0.7445	0.7454	0.8447	0.7640	0.8049
	R_{CPA}	0.0493	0.0483	0.0597	0.0531	0.0469
	F_{1CPA}	0.0925	0.0907	0.1115	0.0992	0.0886
SMAP	P_{CPA}	0.4509	0.3667	0.4153	0.4263	0.4135
	R_{CPA}	0.0471	0.0335	0.0411	0.0412	0.0409
	F_{1CPA}	0.0853	0.0614	0.0748	0.0752	0.0745
SMD	P_{CPA}	0.6232	0.6196	0.6553	0.6182	0.6180
	R_{CPA}	0.4450	0.4478	0.4207	0.4237	0.4276
	F_{1CPA}	0.5192	0.5199	0.5125	0.5028	0.5054
SWAT	P_{CPA}	0.4524	0.4587	0.4287	0.4342	0.4364
	R_{CPA}	0.0633	0.0635	0.0597	0.0601	0.0607
	F_{1CPA}	0.1111	0.1116	0.1048	0.1056	0.1065
AVG	P_{CPA}	0.5520	0.5369	0.5652	0.5399	0.5515
	R_{CPA}	0.1413	0.1382	0.1333	0.1316	0.1327
	F_{1CPA}	0.1953	0.1894	0.1896	0.1838	0.1847

Table 11
Comparison of multivariate short-term forecasting results in terms of execution time (sec) and no. of flops.

	Datasets	LMS-AutoTSF	TimeMixer	iTransformer	PatchTST
Execution Time (Sec) (For test data)	PEMS03	0.8233	0.893	1.0696	1.3208
	PEMS04	0.7945	0.8608	1.1218	1.6141
	PEMS07	0.9783	1.0231	1.4294	2.2553
	PEMS08	0.7762	0.8142	0.8227	0.9495
No. of Flops	PEMS03	151.52M	278.96M	4581.79M	13809.65M
	PEMS04	129.93M	270.17M	15604.76M	46861.69M
	PEMS07	373.72M	369.48M	5721.62M	34061.23M
	PEMS08	71.95M	246.55M	1102.88M	6557.65M



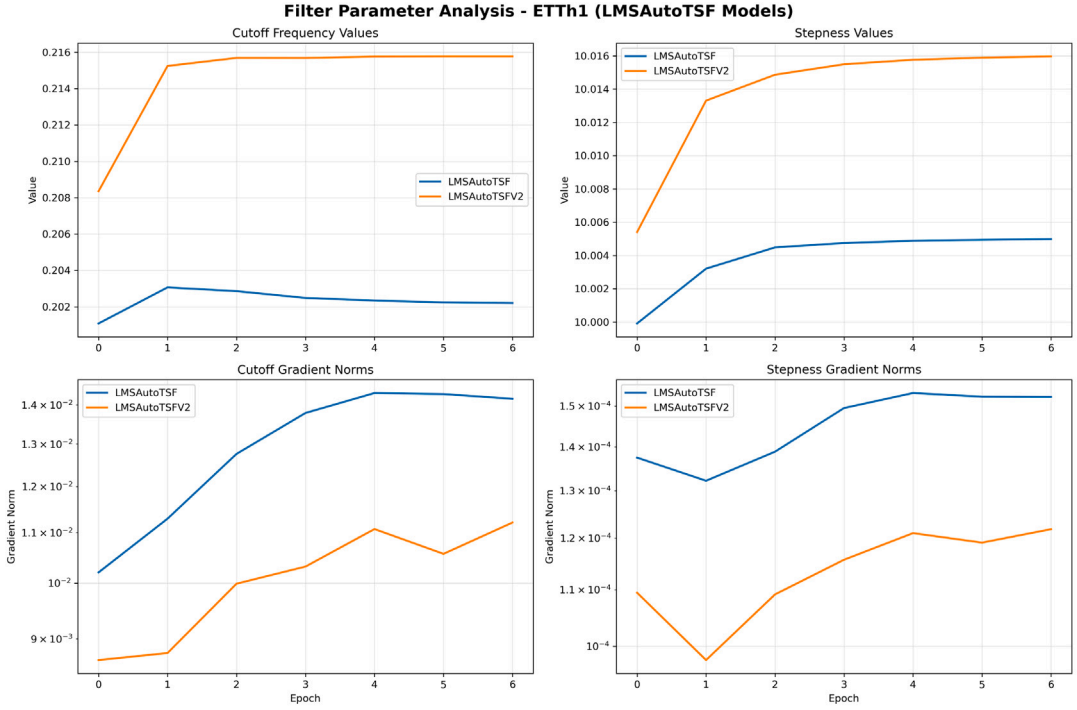
(a) Weather dataset



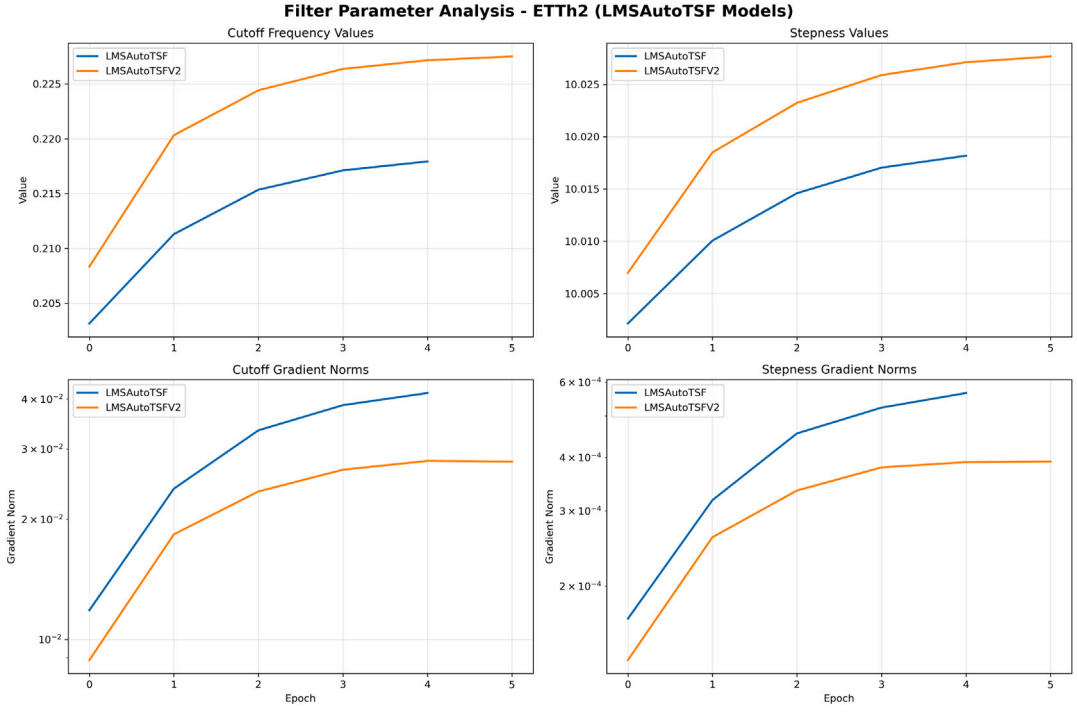
(b) Solar dataset

Fig. 9. Training and validation loss convergence.

as well as parameter convergence and gradient propagation of the learnable frequency filters. The results demonstrate that our proposed models achieve strong performance on both training and validation sets across most datasets. Notably, while some baseline models occasionally achieve lower training loss, our method consistently shows superior performance on validation sets, indicating better generalization capability. This validation superiority aligns with the test set results presented in the main paper, where our approach predominantly outperforms comparative methods, further substantiating its enhanced generalization properties rather than mere training set optimization.



(a) ETTh1 dataset

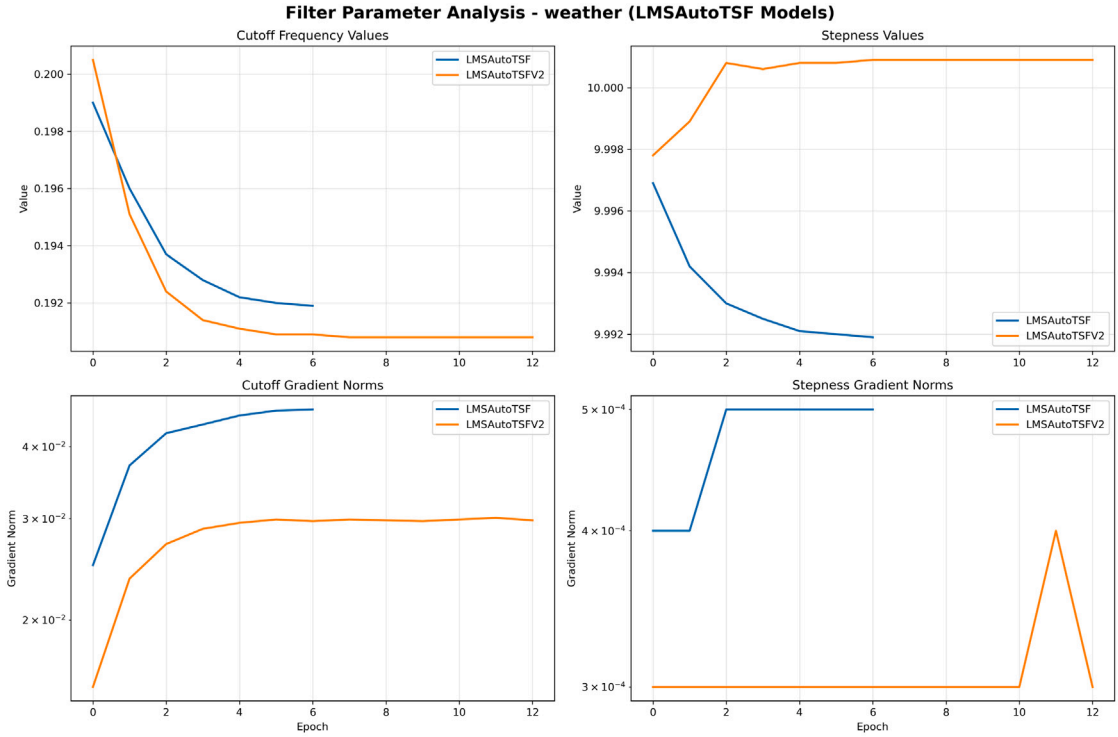


(b) ETTh2 dataset

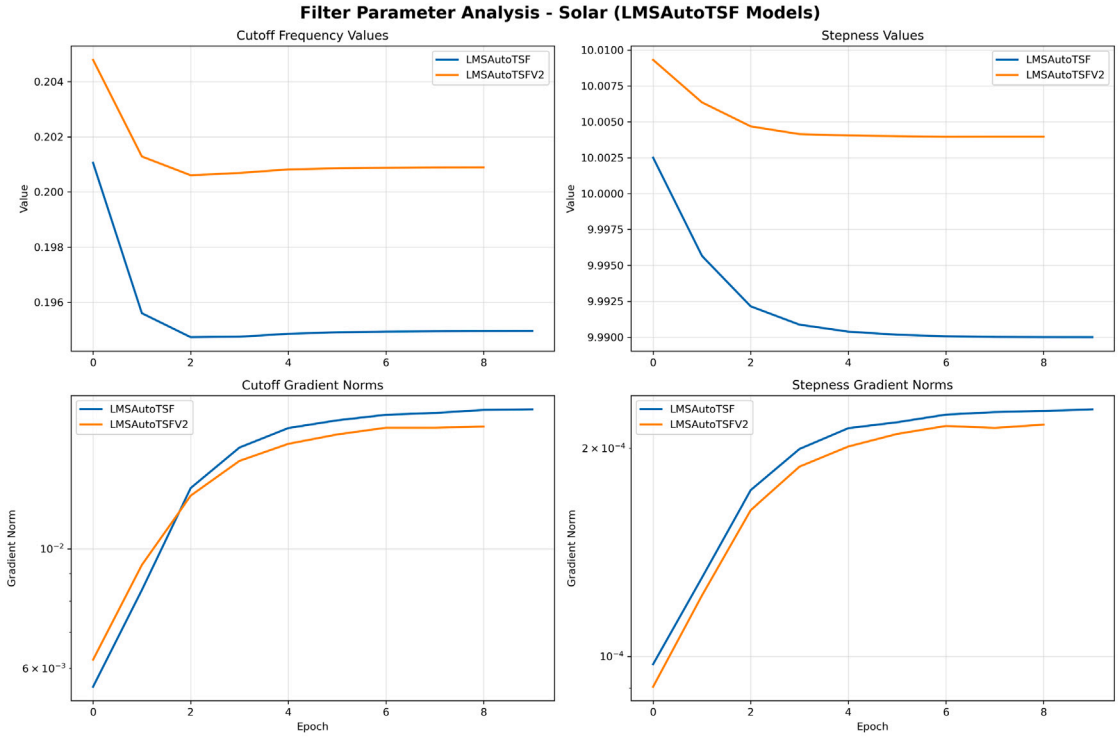
Fig. 10. Evolution of learnable filter parameters and their gradients for proposed models. LMSAutoTSFV2 indicates Confidence Adjustment version of our model LMSAutoTSF.

A.2. Learnable filter analysis

Figs. 10 and 11 illustrate the evolution of learnable filter parameters across different datasets. The cutoff frequencies demonstrate smooth convergence to stable configurations with minimal oscillation, indicating a robust optimization landscape. Notably, the specific convergence patterns exhibit dataset-dependent characteristics for some datasets, cutoff values gradually increase from initial to final iterations (e.g., 0.208 to 0.216), while for Weather dataset they show a slight decrease (e.g., 0.200 to 0.192), reflecting adaptive tuning to distinct temporal frequency structures. Gradient propagation remains well behaved throughout training, with bounded gradient norms maintained within stable ranges (10^{-2} to 10^{-4}) and showing smooth decay patterns that confirm convergence without vanishing or exploding gradient



(a) Weather dataset



(b) Solar dataset

Fig. 11. Evolution of learnable filter parameters and their gradients for proposed models. LMSAutoTSFV2 indicates Confidence Adjustment version of our model LMSAutoTSF.

problems. Similarly, stepness parameters exhibit stable convergence with minimal drift, maintaining consistent gradient flow while adapting to optimize frequency response characteristics for each dataset's unique temporal patterns. These results substantiate the training stability and convergence properties of the LMSAutoTSF architecture, validating its practical deployability and theoretical foundation.

Data availability

The source code is publicly available at https://github.com/mribrahim/lms-tsf.

References

Abdulaal, A., Liu, Z., Lancewicki, T., 2021. Practical approach to asynchronous multivariate time series anomaly detection and localization. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. pp. 2485–2494.

Asadi, R., Regan, A.C., 2020. A spatio-temporal decomposition based deep neural network for time series forecasting. Appl. Soft Comput. 87, 105963.

Challu, C., Olivares, K.G., Oreshkin, B.N., Ramirez, F.G., Canseco, M.M., Dubrawski, A., 2023. Nhits: Neural hierarchical interpolation for time series forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, (6), pp. 6989–6997.

Chen, S., 2015. Beijing PM2.5. http://dx.doi.org/10.24432/C5JS49, UCI Machine Learning Repository.

Chen, Z., Ma, M., Li, T., Wang, H., Li, C., 2023. Long sequence time-series forecasting with deep learning: A survey. Inf. Fusion 97, 101819.

Chen, M., Peng, H., Fu, J., Ling, H., 2021. Autoformer: Searching transformers for visual recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12270–12280.

Chen, C., Petty, K., Skabardonis, A., Varaiya, P., Jia, Z., 2001. Freeway performance measurement system: mining loop detector data. Transp. Res. Rec. 1748 (1), 96–102.

Cleveland, R.B., Cleveland, W.S., McRae, J.E., Terpenning, I., et al., 1990. STL: A seasonal-trend decomposition. J. Off. Stat. 6 (1), 3–73.

Delibasoglu, I., Heintz, F., 2024. Time series anomaly detection leveraging MSE feedback with AutoEncoder and RNN. In: 31st International Symposium on Temporal Representation and Reasoning (TIME 2024). Schloss Dagstuhl–Leibniz-Zentrum für Informatik, pp. 17:1–17:12.

Ekambaram, V., Jati, A., Nguyen, N., Sinthong, P., Kalagnanam, J., 2023. Tsmixer: Lightweight mlp-mixer model for multivariate time series forecasting. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 459–469.

Han, S., Xun, Y., Cai, J., Yang, H., Li, Y., 2025. DyGraphformer: Transformer combining dynamic spatio-temporal graph network for multivariate time series forecasting. Neural Netw. 181, 106776.

Hundman, K., Constantinou, V., Laporte, C., Colwell, I., Soderstrom, T., 2018. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 387–395.

Hyndman, R.J., Athanasopoulos, G., 2018. Forecasting: Principles and Practice. OTexts.

Jiang, H., Liu, D., Ding, X., Chen, Y., Li, H., 2025. TCM: An efficient lightweight MLP-based network with affine transformation for long-term time series forecasting. Neurocomputing 617, 128960.

Kitaev, N., Kaiser, Ł., Levskaya, A., 2020. Reformer: The efficient transformer. In: International Conference on Learning Representation. ICLR, pp. 1–12.

Li, W., Law, K.E., 2024. Deep learning models for time series forecasting: A review. IEEE Access 12, 92306–92327.

Li, Q., Qin, J., Wang, D., Qin, X., Cui, D., Xie, J., Sun, D., 2025. TOEformer: Temporal order enhanced transformer for time series forecasting. Eng. Appl. Artif. Intell. 155, 110965.

Liu, Y., Hu, T., Zhang, H., Wu, H., Wang, S., Ma, L., Long, M., 2024. Itransformer: Inverted transformers are effective for time series forecasting. In: International Conference on Learning Representation. ICLR, pp. 1–25.

Liu, Y., Wu, H., Wang, J., Long, M., 2022. Non-stationary transformers: Exploring the stationarity in time series forecasting. Adv. Neural Inf. Process. Syst. 35, 9881–9893.

Liu, S., Yu, H., Liao, C., Li, J., Lin, W., Liu, A.X., Dustdar, S., 2021. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In: International Conference on Learning Representations.

Mathur, A.P., Tippenhauer, N.O., 2016. SWaT: A water treatment testbed for research and training on ICS security. In: 2016 International Workshop on Cyber-Physical Systems for Smart Water Networks. CySWater, IEEE, pp. 31–36.

Murad, M.M.N., Aktukmak, M., Yilmaz, Y., 2025. Wpmixer: Efficient multi-resolution mixing for long-term time series forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 39, (18), pp. 19581–19588.

Nie, Y., Nguyen, N.H., Sinthong, P., Kalagnanam, J., 2023. A time series is worth 64 words: Long-term forecasting with transformers. In: International Conference on Learning Representation. ICLR, pp. 1–24.

Oreshkin, B., Carpov, D., Chapados, N., Bengio, Y., 2020. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. arxiv 2019. In: International Conference on Learning Representation. ICLR, pp. 1–31.

Qiao, Y., Zhao, P., Wang, J., Hu, R., Li, M., Yuan, X., Li, M., Wei, Z., Feng, C., 2025. Multivariate time series forecasting based on temporal decomposition and graph neural network. Eng. Appl. Artif. Intell. 161, 112074.

Sean, J., Taylor, J., 2018. Forecasting at scale. Am. Stat. 72 (1), 37–45.

Sen, R., Yu, H.-F., Dhillon, I.S., 2019. Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting. Adv. Neural Inf. Process. Syst. 32.

Shumway, R.H., Stoffer, D.S., 2006. Time Series Analysis and Its Applications: with R Examples. Springer.

Su, Y., Zhao, Y., Niu, C., Liu, R., Sun, W., Pei, D., 2019. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 2828–2837.

Wang, S., Li, J., Shi, X., Ye, Z., Mo, B., Lin, W., Ju, S., Chu, Z., Jin, M., 2025. Timemixer++: A general time series pattern machine for universal predictive analysis. In: International Conference on Learning Representation. ICLR, pp. 1–37.

Wang, H., Peng, J., Huang, F., Wang, J., Chen, J., Xiao, Y., 2023. Micn: Multi-scale local and global context modeling for long-term series forecasting. In: The Eleventh International Conference on Learning Representations.

Wang, Y., Wu, H., Dong, J., Qin, G., Zhang, H., Liu, Y., Qiu, Y., Wang, J., Long, M., 2024. Timexer: Empowering transformers for time series forecasting with exogenous variables. Adv. Neural Inf. Process. Syst. 37, 469–498.

Wang, S., Wu, H., Shi, X., Hu, T., Luo, H., Ma, L., Zhang, J.Y., Zhou, J., 2024. Timemixer: Decomposable multiscale mixing for time series forecasting. In: International Conference on Learning Representation. ICLR, pp. 1–28.

Woo, G., Liu, C., Sahoo, D., Kumar, A., Hoi, S., 2022. Etsformer: Exponential smoothing transformers for time-series forecasting. arXiv preprint arXiv:2202.01381.

Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., Long, M., 2023. Timesnet: Temporal 2d-variation modeling for general time series analysis. In: International Conference on Learning Representation. ICLR, pp. 1–23.

Ye, H., Chen, J., Gong, S., Jiang, F., Zhang, T., Chen, J., Gao, X., 2024. ATFNet: Adaptive time-frequency ensembled network for long-term time series forecasting. arXiv preprint arXiv:2404.05192.

Yi, K., Zhang, Q., Fan, W., Wang, S., Wang, P., He, H., An, N., Lian, D., Cao, L., Niu, Z., 2024. Frequency-domain MLPs are more effective learners in time series forecasting. Adv. Neural Inf. Process. Syst. 36.

Zeng, A., Chen, M., Zhang, L., Xu, Q., 2023. Are transformers effective for time series forecasting? In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, (9), pp. 11121–11128.

Zerveas, G., Jayaraman, S., Patel, D., Bhamidipaty, A., Eickhoff, C., 2021. A transformer-based framework for multivariate time series representation learning. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. pp. 2114–2124.

Zhang, Z., Li, W., Liu, H., 2024. Multivariate time series forecasting by graph attention networks with theoretical guarantees. In: International Conference on Artificial Intelligence and Statistics. PMLR, pp. 2845–2853.

Zhang, Y., Ma, L., Pal, S., Zhang, Y., Coates, M., 2024. Multi-resolution time-series transformer for long-term forecasting. In: International Conference on Artificial Intelligence and Statistics. PMLR, pp. 4222–4230.

Zhang, Y., Yan, J., 2023. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In: The Eleventh International Conference on Learning Representations.

Zhang, T., Zhang, Y., Cao, W., Bian, J., Yi, X., Zheng, S., Li, J., 2022a. Less is more: Fast multivariate time series forecasting with light sampling-oriented MLP structures. arXiv preprint arXiv:2207.01186.

Zhang, T., Zhang, Y., Cao, W., Bian, J., Yi, X., Zheng, S., Li, J., 2022b. Less is more: Fast multivariate time series forecasting with light sampling-oriented mlp structures. arXiv preprint arXiv:2207.01186.

Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., Jin, R., 2022. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In: International Conference on Machine Learning. PMLR, pp. 27268–27286.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., Zhang, W., 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, (12), pp. 11106–11115.