

Wave-Mask/Mix: Exploring Wavelet-Based Augmentations for Time Series Forecasting

Dona Arabi^{*}, Jafar Bakhshaliyev^{*}, Ayse Coskuner^{*}, Kiran Madhusudhanan^{*},
and Kami Serdar Uckardes^{*}

University of Hildesheim, 31141 Hildesheim, Germany
{arabi, bakhshaliyev, coskuner, ueckardes}@uni-hildesheim.de
{kiranmadhusud}@ism11.de

Abstract. Data augmentation is important for improving machine learning model performance when faced with limited real-world data. In time series forecasting (TSF), where accurate predictions are crucial in fields like finance, healthcare, and manufacturing, traditional augmentation methods for classification tasks are insufficient to maintain temporal coherence. This research introduces two augmentation approaches using the discrete wavelet transform (DWT) to adjust frequency elements while preserving temporal dependencies in time series data. Our methods, Wavelet Masking (WaveMask) and Wavelet Mixing (WaveMix), are evaluated against established baselines across various forecasting horizons. To the best of our knowledge, this is the first study to conduct extensive experiments on multivariate time series using Discrete Wavelet Transform as an augmentation technique [15]. Experimental results demonstrate that our techniques achieve competitive results with previous methods. We also explore cold-start forecasting using downsampled training datasets, comparing outcomes to baseline methods. Code is available at <https://github.com/jafarbakhshaliyev/Wave-Augs>.

Keywords: Time Series Forecasting · Data Augmentation · Cold-start Forecasting · Wavelet Transform · WaveMask · WaveMix

1 Introduction

Synthetic data generation is crucial when real-world data is scarce or insufficient. Data augmentation creates additional training samples by applying different transformations to existing data. This expands the dataset to include a wider variety of data patterns, strengthening model resilience and improving its ability to generalize to new data [13].

Time series forecasting is crucial in diverse fields such as finance, healthcare, meteorology, and manufacturing, and data augmentation techniques have become essential for tasks such as classification, forecasting, anomaly detection, and clustering [2]. Data augmentation (DA) is increasingly essential in the realm

^{*} Equal Contribution

of time series forecasting due to its ability to enhance model performance and improve generalization. Time series data, unlike image data, presents unique characteristics in both the time and frequency domains, making its analysis more intricate. While current augmentation methods are tailored for classification purposes, it is crucial in the context of time series forecasting to emphasize diversity and consistency with the original temporal patterns. In Time Series Forecasting (TSF), data points within the look-back window and forecasting horizon serve as both the data and the label, enabling the modeling of complex temporal relationships by accurately capturing data within the specified time frame [2,20].

The existing fundamental augmentation techniques encompass scaling, flipping [14], window cropping or slicing [7], Gaussian noise injection [16], dynamic time warping [15], and other methods. However, these methods have a significant limitation as they may introduce missing values or alter periodic patterns within the series, rendering them unsuitable for TSF research. Consequently, the precision required for augmented data-label pairs in TSF exceeds that needed for other time series analysis tasks [2]. STAUG, a decomposition-based augmentation approach [20], is effective for TSF but is inefficient and memory-intensive, limiting its application to large datasets. One of the baseline methods is the frequency-domain augmentation technique, FrAug [2], which motivated our study. Due to its utilization of Fourier transform to decompose the signal, a key drawback is its exclusive focus on frequency components at the expense of time resolution [2]. Our methods, employing wavelet decomposition, exhibit potential for superior results compared to alternative time-frequency analysis techniques. These methods, along with others for comparison, are depicted in Fig. 1. To grasp the limitations of each approach, we consider a signal with unknown frequency components. The Fourier transform provides detailed information about frequency components over time, showcasing high resolution in the frequency domain while overlooking variations in the time domain. Short-time Fourier Transform (STFT) addresses this limitation by analyzing localized frequency variations in the time domain. Nonetheless, STFT’s fixed resolution poses a challenge, as its window boundaries remain constant [9,12]. Overcoming this limitation, the wavelet transform offers variable window sizes, providing high frequency domain resolution and low time domain resolution for small frequency values and large frequency values respectively [1,9,8]. Given this information, adjusting the signal’s frequency components at different resolutions is expected to yield favorable outcomes.

In this study, we introduce two innovative yet direct and simple data augmentation methods, termed as *wavelet masking* (WaveMask) and *wavelet mixing* (WaveMix). These techniques utilize the discrete wavelet transform (DWT) to obtain wavelet coefficients (both approximation and detail coefficients) by breaking down the signal and adjusting these coefficients, in line with modifying frequency components across different time scales. The DWT accomplishes this decomposition by employing a sequence of filtering and downsampling procedures, resulting in a structured representation of the signal at different levels of resolu-

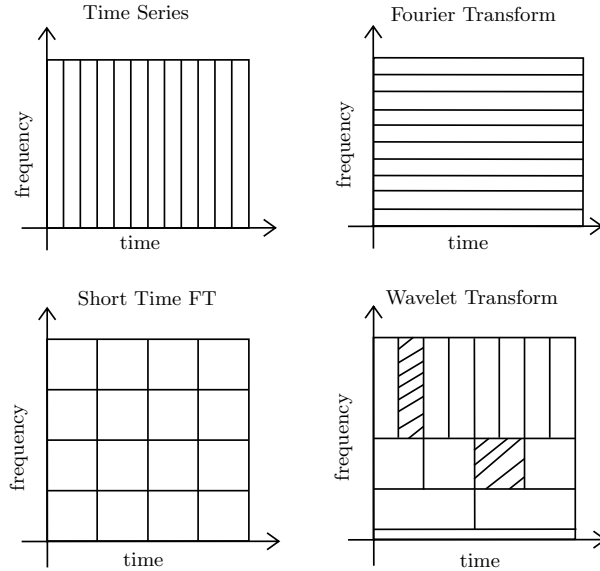


Fig. 1. Illustration of time-frequency analysis methods [5]

tion. More extensive insights into DWT can be found in Appendix A. WaveMask selectively eliminates specific wavelet coefficients at each decomposition level, thereby introducing variability in the augmented data. Conversely, WaveMix exchanges wavelet coefficients from two distinct instances of the dataset, thereby enhancing the diversity of the augmented data.

The key contributions of this study include:

- WaveMask and WaveMix techniques provide a richer understanding of a signal compared to FrAug methods, making them more competitive and straightforward.
- Our methods demonstrate superior performance in 12 out of 16 forecasting horizon tasks and rank second best in the remaining four tasks, using datasets ETTh1, ETTh2, Weather, and ILI.
- On cold-start forecasting, our methods have consistently achieved remarkable results, resulting in a higher discrepancy.

2 Related Work

Basic Augmentation Methods: As mentioned earlier, basic augmentation operations are scaling, flipping [14], window cropping or slicing [7], Gaussian noise injection [16], dynamic time warping [15], and others. Scaling operations alter the magnitude of data within a defined range by using a randomly generated scalar multiplier, while flipping reverses the sequence of data points along the time axis [14]. Window cropping is a method that selects random sections of

a specific length from time series data, and each section is classified using a majority voting approach during testing. Gaussian noise injection is a technique that involves adding small amounts of noise or outliers to time series data while keeping the original labels intact [15]. These data augmentation techniques are mostly suitable for time series classification or anomaly detection as they do not maintain temporal coherence, which is crucial for time series forecasting. When these methods are applied for forecasting, they yield inferior results [2].

Decomposition-based Augmentation Methods: Some decomposition-based approaches break down time series data into constituent elements such as trend, seasonality, and residual components through techniques such as RobustSTL or STL. Subsequently, these elements are transformed and merged to generate new time series data by modifying weights and implementing statistical models [15]. Some methods that rely on decomposition use Empirical mode decomposition (EMD), and one particular method is known as STAug method. The EMD algorithm breaks down a signal into multiple components called Intrinsic Mode Functions (IMFs). The initial IMFs capture high-frequency components, while the final IMFs represent low-frequency trend information. STAug utilizes two distinct instances, applies EMD to break them into components, and then reconstructs these subcomponents using randomly sampled weights from a uniform distribution. It then applies linear interpolation of these two weights to obtain augmented data for time series forecasting [20]. We have conducted experiments using the STAug method, which serves as one of our baseline approaches.

Frequency-Domain Augmentation Methods: Some studies investigate data augmentation techniques that work on only the frequency domain; one of the most recent works was conducted by Gao et al. [6]. They use the Fourier transform to manipulate frequency components, such as adding perturbations or eliminating a few of them [15]. One of the recent works is also FrAug methods, which consist of two methods: *frequency masking* (FreqMask) and *frequency mixing* (FreqMix) [2]. These techniques form the fundamental basis of our paper and serve as our baselines. FreqMask employs the Fast Fourier transform (FFT) to compute the frequency domain representations of input data. Subsequently, it generates augmented data by randomly masking specific frequency components and applies the inverse real FFT for signal reconstruction. The procedure for FreqMix is similar, except that it randomly substitutes the frequency components in one training instance with the frequency components of another training instance [2].

3 Problem Formulation

A multivariate time series sequence of length T and channels K is denoted as $\mathcal{X} = (x_1, \dots, x_T) \in \mathbb{R}^{T \times K}$. As it is a **Time Series Forecasting Problem**, we only observe data up to timestamp $t < T$. Based on the forecasting horizon, we aim to forecast future values starting from $t + 1, \dots, T$. Given input $X = (x_1, \dots, x_t) \in \mathbb{R}^{t \times K}$, the objective is to learn a model f that forecasts $F =$

$(x_{t+1}, \dots, x_T) \in \mathbb{R}^{(T-t) \times K}$ to obtain predictions for the future part. We learn a model for TSF as:

$$f_\theta : \mathbb{R}^{t \times K} \rightarrow \mathbb{R}^{(T-t) \times K}, \quad X \mapsto f_\theta(X)$$

The data X may be limited or insufficient for time series forecasting, the synthetic time series data $\bar{X} = (\bar{x}_1, \dots, \bar{x}_t) \in \mathbb{R}^{t \times K}$ is generated by applying data augmentation methods. Therefore, we want to learn a model:

$$f_\theta : \mathbb{R}^{t \times K} \times \mathbb{R}^{t \times K} \rightarrow \mathbb{R}^{(T-t) \times K}, \quad X' \mapsto f_\theta(X')$$

where the data:

$$X' = [X, \bar{X}] = [[x_1, \dots, x_t], [\bar{x}_1, \dots, \bar{x}_t]]$$

The loss functions are the mean squared error (MSE) and the mean absolute error (MAE) averaged over future time points.

4 Methodology

This section provides a detailed structure of how the training framework works with our proposed methods. We will also detail these two simple methods with illustrations and pseudocodes.

4.1 Overview

The discrete wavelet transform (DWT), as discussed in Appendix A, is proposed for enhancing time series data by preserving intricate data characteristics and facilitating multi-resolution analysis that accounts for various frequencies at different resolutions. The illustration in Fig. 2 depicts a framework of the training stages incorporating wavelet augmentations, which involve the concatenation of the look-back window and the forecasting horizon prior to transformation and augmentation. Batch sampling of the generated synthetic data is conducted according to a predefined hyperparameter called the *sampling rate*. These batches are subsequently used to split the data into the look-back window and target horizon, after which they are concatenated with the original data. By maintaining consistency with the notations presented in the Problem Formulation section, the random batch sampling operation can be defined as:

$$f_{\text{sample}} : \mathbb{R}^{b \times t \times K} \rightarrow \mathbb{R}^{n \times t \times K}, \quad \bar{X} \mapsto f_{\text{sample}}(\bar{X})$$

where b is our initial batch size and n is determined based on the hyperparameter *sampling rate* with $n \leq b$.

Our selected model, DLinear [19], is characterized by a decomposition scheme and a linear layers. Given the look-back window as $X \in \mathbb{R}^{L \times K}$, DLinear model initially decomposes it into a trend component $X_t \in \mathbb{R}^{L \times K}$ and a residual part

$X_r \in \mathbb{R}^{L \times K}$. In order to produce the forecasting output $F \in \mathbb{R}^{T \times K}$, linear layers are employed for both the trend and residual components as follows: $H_t = W_t X_t$ where $W_t \in \mathbb{R}^{T \times L}$ and $H_r = W_r X_r$ where $W_r \in \mathbb{R}^{T \times L}$ respectively. The forecast output is obtained by summing both components H_t and H_r [19]. Augmentations are exclusively implemented during the training phase, while the original test samples remain unaltered for evaluation purposes.

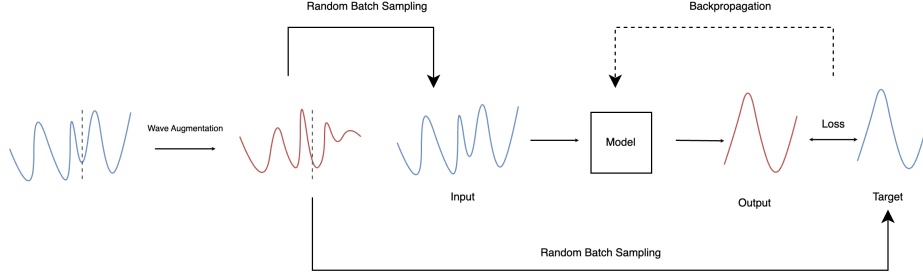


Fig. 2. Wavelet-Integrated Training Framework

4.2 Wavelet Masking (WaveMask)

Our first method, **WaveMask**, creates a synthetic dataset based on the input, which is a concatenation of the look-back window and the target horizon (Line 1 in Alg. 1). The method is illustrated in Fig. 3. WaveDec (Wavelet Decomposition) performs multilevel discrete wavelet transform (DWT) on the input signal. It decomposes the signal into the last approximation and all detail coefficients for each resolution level \mathcal{L} (Line 3 in Alg. 1). We mask the wavelet coefficients based on the array of hyperparameter *rates* ($\mu^{\mathcal{L}+1}$), which contains parameter *rate* for the last approximation and all detail coefficients; it has a length of $\mathcal{L} + 1$ (Line 4 in Alg. 1). It determines the probability of each coefficient being masked (set to zero) at that specific level, allowing for level-specific control over the masking process. CreateRandomMask generates boolean values (True or False) to determine whether to mask or not mask sections according to the *rate*, and the Masking function replaces True elements with zeros (Line 5 & 6 in Alg. 1). WaveRec (Wavelet Reconstruction) applies the inverse discrete wavelet transform (IDWT) to reconstruct the signal from its wavelet coefficients. It synthesizes the original signal structure from the modified/masked wavelet coefficients, effectively reversing the decomposition process (Line 8 in Alg. 1). We implemented it for each channel (c) separately (Line 2 in Alg. 1). Masking wavelet coefficients allow us to eliminate frequency components at various time scales selectively. When we reconstruct the signal using the masked coefficients, we obtain a modified version of the original signal in which specific frequency components have been eliminated.

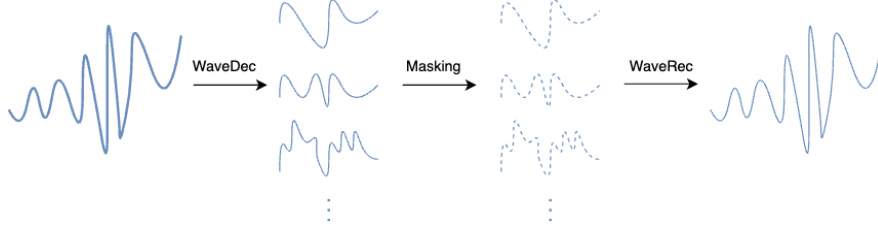


Fig. 3. The proposed wavelet masking (WaveMask) augmentation method

Algorithm 1: WaveMask

Input: Look-back window x , target horizon y , resolution level \mathcal{L} ,
mask rate $\mu^{\mathcal{L}+1}$
Output: Masked sequence \tilde{s}

```

1  $s \leftarrow x || y$  ; // Concatenate  $x$  and  $y$ 
2 for each channel  $c$  do
3    $W^{\mathcal{L}+1} \leftarrow \text{WaveDec}(s[:, :, c])$  ; // Multilevel decomposition
4   for  $i \in \{1, \dots, \mathcal{L} + 1\}$  do
5      $m \leftarrow \text{CreateRandomMask}(\text{len}(W[i]), \mu[i])$ 
6      $\tilde{W}[i] \leftarrow \text{Masking}(W[i], m)$  ; // Apply mask to coefficients
7   end
8    $\tilde{s}[c] \leftarrow \text{WaveRec}(\tilde{W})$  ; // Multilevel reconstruction
9 end
10 return  $\tilde{s}$ 

```

4.3 Wavelet Mixing (WaveMix)

The second method, **WaveMix**, extends the **WaveMask** concept to mix two input signals using wavelet transformation. It takes two different input pairs, concatenates these look-back windows with the corresponding target horizons, and outputs augmented data after applying the transformation pipeline. The pipeline is illustrated in Fig. 4. In the same way as our first method, WaveMask, we implement wavelet decomposition for both input pairs separately. CreateRandomMask generates a binary mask based on the mix rate $\mu[i]$ ($i = 1, \dots, \mathcal{L} + 1$) for each wavelet level, determining which coefficients to select from each input signal. BitwiseNOT operation creates a complementary mask, ensuring that coefficients not selected from one signal are taken from the other. Masking applies the generated masks to the wavelet coefficients of each input signal and we add both to select portions from each effectively (Lines 5-7 in Alg. 1). We apply the same procedure as wavelet reconstruction to obtain the augmented data. By exchanging wavelet coefficients between inputs, the frequency components of one data instance are integrated into the other, and vice versa. This process enables the mixing of frequency information across different time granularities between the two data instances. We ran the process for each channel separately.

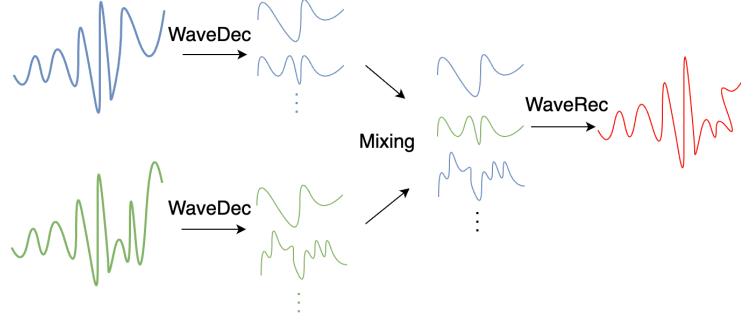


Fig. 4. The proposed wavelet mixing (WaveMix) augmentation method

Algorithm 2: WaveMix

Input: Look-back window x_1 , target horizon y_1 , another training sample pair x_2, y_2 , resolution level \mathcal{L} , mix rate $\mu^{\mathcal{L}+1}$

Output: Masked sequence \tilde{s}

```

1  $s_1 \leftarrow x_1 || y_1$ ;  $s_2 \leftarrow x_2 || y_2$ ; // Concatenate  $x$  and  $y$ 
2 for each channel  $c$  do
3    $W_1^{\mathcal{L}+1} \leftarrow \text{WaveDec}(s_1[:, :, c])$ ;  $W_2^{\mathcal{L}+1} \leftarrow \text{WaveDec}(s_2[:, :, c])$ ;
4   for  $i \in \{1, \dots, \mathcal{L} + 1\}$  do
5      $m_1 \leftarrow \text{CreateRandomMask}(\text{len}(W_1[i]), \mu[i])$ ;
6      $m_2 \leftarrow \text{BitwiseNOT}(m_1)$ ;
7      $\tilde{W}[i] \leftarrow \text{Masking}(W_1[i], m_1) + \text{Masking}(W_2[i], m_2)$ ;
8   end
9    $\tilde{s}[c] \leftarrow \text{WaveRec}(\tilde{W})$ ; // Multilevel reconstruction
10 end
11 return  $\tilde{s}$ 

```

5 Experiments

The first section provides an overview of the datasets, their statistical information, the baseline methods, and the experimental setup. Section 5.2 presents the main results regarding four datasets. The final section presents the ablation study of cold-start forecasting, wherein models are trained on down-sampled training datasets.

5.1 Datasets, Baselines and Experimental Setup

Our methods are assessed using four datasets: ETTh1, ETTh2, Weather, and ILI [21,10]. We selected these datasets as they represent a diverse range of time series forecasting tasks across different domains. The statistical details for the datasets can be found in Table 1. Our approach follows the methodology delineated in the prior research [2,19], which utilized a 24-length look-back window

and forecasting horizons of (24, 36, 48, 60) for ILI. For the rest, a 336-length look-back window and forecasting horizons of (96, 192, 336, 720) were used. The forecasting horizons were chosen to evaluate performance on short-, medium-, and long-term predictions, aligning with common practices in time series forecasting literature [15]. The datasets are segmented into train, validation, and test sets at a ratio of 6:2:2. The comparison includes the FreqMask, FreqMix [2], and STAug [20] techniques. We implemented an optimized version of STAug that performs Empirical Mode Decomposition (EMD) for the entire sequence. The evaluation employs the following metrics:

- Mean Squared Error (MSE) - In the context of multivariate time series forecasting with multiple channels, MSE is calculated by squaring the differences between actual and predicted values for each channel, averaging these squared errors across all data points, and then averaging the results across all channels.
- Mean Absolute Error (MAE) - Similar to MSE, MAE is derived by taking the absolute differences between actual and predicted values for each channel, averaging these absolute errors across all data points, and then averaging the results across all channels.

We utilized DLinear model [19] and treated each forecasting horizon as a distinct task. Our approaches fine-tune the parameters for each task through Random Search, while the baseline methods employ Grid Search [11]. The optimized hyperparameters are provided in Appendix B. In contrast to the experimental setup of the baseline methods, we executed a 30-epoch run with a patience level of 12 and selected the optimal one based on validation loss. The batch size is configured to 64, and a learning schedule is incorporated. A random sampling strategy is utilized to choose a subset of the generated data.

Table 1. The statistical information of datasets

Datasets	ETTh1	ETTh2	Weather	ILI
Variates	7	7	21	7
Frequency	1 hour	1 hour	10 min	1 week
Total Timesteps	17420	17420	52696	966

5.2 Main Results

Table 2 and Table 3 present comparisons between our methods and baselines in terms of the metrics Mean Squared Error (MSE) and Mean Absolute Error (MAE). The best result is indicated in bold, while the second most favorable outcome is underlined. We conduct the experiments a total of 10 times and report standard deviations in contrast to the experiments conducted by the previous studies [2,20]. Every forecasting horizon is treated as an individual task, and

the parameters are optimized independently. Among the 16 tasks, our methods, WaveMask and WaveMix, outperform all baselines in 12. Among these four tasks, they rank as the second most effective methods. The STAug in the Weather dataset exceeds the memory capacity of the GPU due to creating a tensor-based on various factors such as data length, maximum Intrinsic Mode Function (IMF), sequence length, forecasting horizon, and channels. We can observe that STAug method is not stable and standard deviations are high, especially in the ETTh2 dataset.

Table 2. Comparison of augmentation methods for various forecasting horizons under MSE. * denotes out-of-memory issue

Methods		None	FreqMask	FreqMix	STAug	WaveMask	WaveMix
ETTh1	96	0.3708 \pm 1.66e-05	0.3698 \pm 7.69e-05	0.3698 \pm 0.0001	0.3707 \pm 0.007	<u>0.3698 \pm 2.07e-05</u>	0.3696 \pm 1.58e-05
	192	0.453 \pm 0.0002	0.4559 \pm 0.0008	0.4537 \pm 0.0004	0.4552 \pm 0.06	<u>0.4484 \pm 0.0002</u>	0.4464 \pm 0.0003
	336	0.5382 \pm 0.0007	0.5170 \pm 0.002	0.5307 \pm 0.01	0.5199 \pm 0.01	<u>0.4995 \pm 0.0002</u>	0.4967 \pm 0.0001
	720	0.5109 \pm 0.0002	0.5106 \pm 0.002	0.4981 \pm 0.0002	<u>0.4866 \pm 0.02</u>	0.5089 \pm 0.0002	0.5076 \pm 0.0002
ETTh2	96	0.3035 \pm 3.69e-05	0.3096 \pm 0.003	0.3031 \pm 0.0003	0.4179 \pm 0.03	0.3018 \pm 7.76e-05	<u>0.303 \pm 0.0004</u>
	192	0.4171 \pm 0.002	0.4153 \pm 0.07	<u>0.4152 \pm 0.003</u>	0.5778 \pm 0.13	0.4152 \pm 0.002	0.4230 \pm 0.005
	336	0.4799 \pm 0.003	0.4512 \pm 0.005	0.4764 \pm 0.002	1.1409 \pm 0.23	0.47 \pm 0.003	<u>0.4644 \pm 0.003</u>
	720	0.5471 \pm 0.0004	<u>0.54 \pm 0.003</u>	0.5473 \pm 0.0007	1.9252 \pm 0.28	0.5469 \pm 0.0004	0.5117 \pm 0.0007
Weather	96	0.1858 \pm 2.27e-06	0.1879 \pm 0.002	0.1895 \pm 0.002	*	<u>0.1855 \pm 1.34e-05</u>	0.1852 \pm 2.42e-05
	192	0.2389 \pm 8.16e-07	0.2382 \pm 0.0002	0.2393 \pm 5.8e-05	*	0.2352 \pm 3.02e-05	0.2353 \pm 4.21e-05
	336	0.2921 \pm 5.49e-07	0.2908 \pm 0.0002	0.2922 \pm 3.1e-05	*	0.2893 \pm 5.34e-07	0.2896 \pm 5.69e-05
	720	0.3555 \pm 2.18e-07	0.3505 \pm 0.0002	0.3553 \pm 4.33e-05	*	0.3553 \pm 6.45e-07	<u>0.3515 \pm 4.04e-05</u>
ILI	24	2.7759 \pm 0.008	2.7855 \pm 0.01	2.7797 \pm 0.008	2.7385 \pm 0.02	2.7768 \pm 0.009	<u>2.775 \pm 0.008</u>
	36	3.0018 \pm 0.01	3.053 \pm 0.008	3.034 \pm 0.01	3.0627 \pm 0.03	<u>3.0014 \pm 0.008</u>	2.8912 \pm 0.003
	48	2.9205 \pm 0.009	2.9553 \pm 0.009	2.9428 \pm 0.008	2.9598 \pm 0.04	<u>2.9082 \pm 0.008</u>	2.819 \pm 0.001
	60	3.0579 \pm 0.006	3.0576 \pm 0.007	3.0595 \pm 0.007	3.1042 \pm 0.02	<u>3.0322 \pm 0.005</u>	3.0195 \pm 0.004

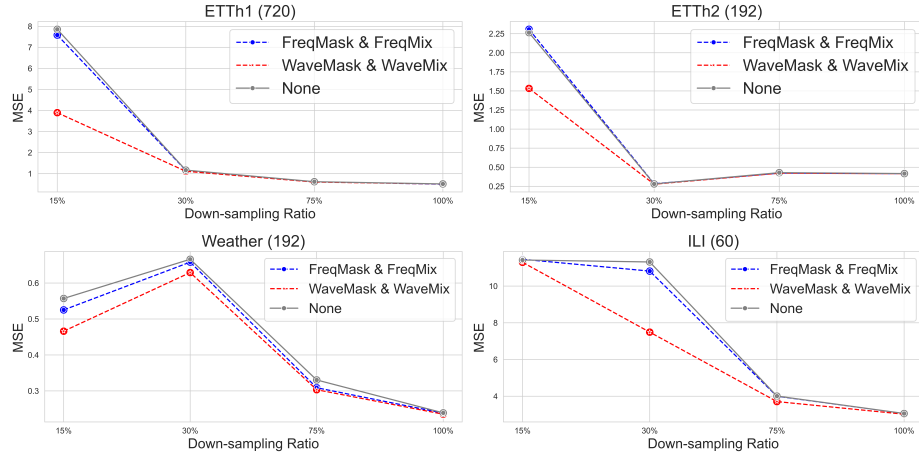
5.3 Ablation Study

We carried out an evaluation of the cold-start forecasting task utilizing the ETTh1, ETTh2, Weather, and ILI datasets across different forecasting horizons. The downsampling rates indicate the percentage of the training dataset that was retained while keeping the test dataset unchanged. We chose different down-sampling rates of 15%, 30%, and 75% for the evaluations. The performance, as depicted in Fig. 5, is presented using the Mean Squared Error (MSE) metric, with the forecasting horizon specified in parentheses. The optimal performance, indicated by the minimum value of MSE values, was chosen from the FreqMask & FreqMix and WaveMask & WaveMix combinations, respectively. It is obvious that our methods deliver superior outcomes, particularly noticeable when the downsampling rate is as low as 15%. Furthermore, it is noteworthy that we did

Table 3. Comparison of augmentation methods for various forecasting horizons under MAE. * denotes out-of-memory issue

Methods		None	FreqMask	FreqMix	STAug	WaveMask	WaveMix
ETTh1	96	$0.3946 \pm 2.06\text{e-}05$	0.3942 ± 0.0001	0.3947 ± 0.0002	0.3992 ± 0.008	<u>$0.3939 \pm 2.96\text{e-}05$</u>	$0.3936 \pm 1.69\text{e-}05$
	192	0.4660 ± 0.0002	0.4682 ± 0.0007	0.4667 ± 0.0004	0.4645 ± 0.003	<u>0.4619 ± 0.0002</u>	0.4601 ± 0.0003
	336	0.5257 ± 0.0004	0.5112 ± 0.001	0.5163 ± 0.01	0.5064 ± 0.008	<u>0.4967 ± 0.0001</u>	0.4944 ± 0.0001
	720	0.5229 ± 0.0001	0.5209 ± 0.0008	0.5105 ± 0.0002	0.5164 ± 0.02	0.5212 ± 0.0001	<u>0.5189 ± 0.0001</u>
ETTh2	96	$0.3652 \pm 3.84\text{e-}05$	0.3646 ± 0.002	<u>0.3648 ± 0.0003</u>	0.4393 ± 0.02	$0.3641 \pm 6.87\text{e-}05$	0.3650 ± 0.0004
	192	0.4317 ± 0.001	0.4311 ± 0.005	<u>0.4309 ± 0.002</u>	0.5179 ± 0.07	0.4296 ± 0.002	0.4367 ± 0.003
	336	0.479 ± 0.002	0.4638 ± 0.003	0.4771 ± 0.001	0.7465 ± 0.09	0.4735 ± 0.002	<u>0.4701 ± 0.002</u>
	720	0.5276 ± 0.0002	<u>0.5239 ± 0.002</u>	0.5278 ± 0.0003	1.0115 ± 0.08	0.5274 ± 0.0002	0.5090 ± 0.0004
Weather	96	$0.2595 \pm 5.24\text{e-}06$	0.2629 ± 0.004	0.2662 ± 0.004	*	<u>$0.2590 \pm 2.87\text{e-}05$</u>	$0.2584 \pm 5.7\text{e-}05$
	192	$0.3138 \pm 3.84\text{e-}06$	0.3127 ± 0.0003	$0.3144 \pm 8.08\text{e-}05$	*	$0.3090 \pm 4.47\text{e-}05$	<u>$0.3094 \pm 5.5\text{e-}05$</u>
	336	$0.3583 \pm 2.64\text{e-}06$	0.3569 ± 0.0002	$0.3586 \pm 3.90\text{e-}05$	*	$0.3542 \pm 3.41\text{e-}06$	<u>$0.3556 \pm 7.57\text{e-}05$</u>
	720	$0.4079 \pm 5.42\text{e-}07$	0.4022 ± 0.0002	$0.4082 \pm 6.19\text{e-}05$	*	$0.4077 \pm 9.80\text{e-}07$	<u>$0.4039 \pm 4.78\text{e-}05$</u>
ILI	24	1.1371 ± 0.002	1.1385 ± 0.003	1.1373 ± 0.002	<u>1.1333 ± 0.005</u>	1.1354 ± 0.002	1.1333 ± 0.002
	36	1.1604 ± 0.003	1.1710 ± 0.002	1.1678 ± 0.003	1.1781 ± 0.007	<u>1.1583 ± 0.002</u>	1.1263 ± 0.0008
	48	1.1575 ± 0.002	1.1604 ± 0.002	1.1601 ± 0.002	1.1668 ± 0.006	<u>1.1531 ± 0.002</u>	1.1256 ± 0.0002
	60	1.2029 ± 0.001	1.1995 ± 0.002	1.2 ± 0.002	1.2110 ± 0.006	<u>1.1953 ± 0.001</u>	1.1880 ± 0.0009

not fine-tune hyperparameters for the augmentation techniques; instead, we utilized the same parameters optimized on the full training data. We hypothesize that our methods, WaveMask and WaveMix, may generate exceptional results at lower downsampling rates through hyperparameter optimization, and this area of research holds potential for future investigation.

**Fig. 5.** Cold-start forecasting for different datasets. Parentheses show the forecasting horizon

6 Discussion & Conclusion

All of these data augmentation methods can be assessed across a wide range of diverse dataset domains, in addition to the experiments already conducted. To the best of our knowledge, there are no state-of-the-art techniques other than those we evaluated, except TimeGAN [17], to compare with our methods as the methods we used are also evaluated with little data in cold-start forecasting setting. The evaluations cannot be directly compared to the paper [2] because we conducted them using 30 epochs instead of 10 epochs and selected the best model from the validation dataset. In addition, we performed the running task 10 times and reported the standard deviation. Instead of utilizing Random Search, more sophisticated search techniques could be employed to discover optimal parameters for WaveMask and WaveMix that may result in exceptional outcomes. We initially employed DLinear model [19], however, for further progress, TSMixer [3] can be utilized as it demonstrates superior outcomes and attains state-of-the-art performance. Furthermore, the augmentation methods can be implemented following linear projection layers and techniques that function on representations of time series, such as in TS2Vec [18].

In conclusion, both our methods, WaveMask and WaveMix, outperform the baseline methods [2,20] and are more straightforward and comprehensive techniques compared to methods that solely focus on the frequency domain without considering the time domain. Based on the ablation study, our methods have also been shown to achieve outstanding results in cold-start forecasting.

A Discrete Wavelet Transform (DWT)

During a wavelet transform, a signal is subjected to multiplication by a wavelet function, which is a localized wave with finite energy. The resulting transform is then analyzed for each segment. The equation that defines a continuous wavelet transform (CWT) is as follows:

$$H(x) = \frac{1}{|\sqrt{\zeta}|} \int x(t) \cdot \psi^* \left(\frac{t - \tau}{\zeta} \right) dt \quad (1)$$

where $H(x)$ is the wavelet transform for the signal $x(t)$ as a function of time t , ζ is the scale parameter, τ is the time parameter, and ψ is the mother wavelet or the basis function. High frequencies, or small scales, compress the signal and convey global information, while low frequencies, or large scales, expand the signal and reveal hidden detailed information within it. Nevertheless, the cost remains high, and DWT offers effective computation through subband coding, where the signal undergoes filtering with distinct cutoff frequencies at various scales. The DWT is calculated sequentially by applying high-pass and low-pass filters to a signal, resulting in detail and approximation coefficients. The half-band filters reduce the signal's sampling rate by a factor of 2 at each level of decomposition. This method of breaking down and refining can be iterated until the desired level has been achieved. The original signal can be reconstructed by up-sampling the

approximation and detail coefficients at each level, filtering them through high- and low-pass synthesis filters, and adding them together [1,4].

B Hyperparameters

Random Search is applied to optimize the hyperparameters for our methods, WaveMask and WaveMix, while Grid Search is used for the remaining methods. The parameters that have been optimized are provided below:

Table 4. Hyperparameters for each dataset

Datasets		WaveMask				WaveMix				FreqMask	FreqMix	STAug	
		Mother Wavelet	Level	Rates	Sampling rate	Mother Wavelet	Level	Rates	Sampling rate	Rate	Rate	Rate	nJMF
ETTh1	96	db2	3	[0.5, 0.3, 0.9, 0.9]	0.2	db3	1	[0.0, 0.9]	0.2	0.1	0.2	0.9	100
	192	db3	1	[0.0, 1.0]	0.2	db3	1	[1.0, 0.4]	0.8	0.5	0.1	0.9	900
	336	db25	1	[0.1, 0.9]	0.8	db3	1	[0.0, 0.9]	0.8	0.5	0.1	0.8	200
	720	db1	1	[0.4, 0.9]	0.2	db5	1	[0.1, 0.9]	0.8	0.4	0.6	0.7	1000
ETTh2	96	db26	2	[0.4, 0.9, 0.0]	0.5	db25	2	[0.9, 0.9, 0.1]	0.2	0.6	0.9	0.4	2000
	192	db26	2	[0.6, 0.7, 0.5]	0.8	db1	3	[0.9, 0.4, 0.1, 0.8]	0.5	0.6	0.8	0.9	200
	336	db1	3	[0.2, 0.7, 0.9, 0.4]	0.8	db25	3	[0.9, 0.1, 0.2, 0.5]	0.8	0.1	0.8	0.6	100
	720	db5	4	[0.8, 0.9, 0.4, 0.9, 0.4]	0.2	db5	1	[0.5, 0.1]	1.0	0.1	0.1	0.4	700
Weather	96	db2	2	[0.2, 1.0, 0.4]	0.5	db3	1	[0.1, 0.5]	1.0	0.1	0.9	*	*
	192	db2	1	[0.1, 0.7]	0.5	db3	1	[0.2, 0.7]	1.0	0.1	0.1	*	*
	336	db1	1	[1.0, 1.0]	1.0	db2	1	[0.8, 0.6]	1.0	0.1	0.1	*	*
	720	db2	1	[1.0, 0.8]	0.5	db1	1	[0.1, 0.1]	1.0	0.9	0.9	*	*
ILI	24	db25	1	[0.4, 0.8]	0.2	db1	1	[0.1, 0.8]	0.2	0.2	0.1	0.7	200
	36	db25	1	[0.6, 0.8]	0.2	db25	1	[0.1, 1.0]	0.8	0.1	0.1	0.3	300
	48	db2	1	[0.2, 0.7]	0.2	db3	1	[0.1, 1.0]	1.0	0.1	0.1	0.9	300
	60	db25	1	[0.2, 0.8]	0.2	db1	1	[0.1, 0.9]	0.5	0.1	0.1	0.7	1000

References

1. Chaovalit, P., Gangopadhyay, A., Karabatis, G., Chen, Z.: Discrete wavelet transform-based time series analysis and mining. *ACM Comput. Surv.* **43**(2), Article 6 (January 2011). <https://doi.org/10.1145/1883612.1883613>, 37 pages
2. Chen, M., Xu, Z., Zeng, A., Xu, Q.: Fraug: Frequency domain augmentation for time series forecasting (2023), <https://arxiv.org/abs/2302.09292>
3. Chen, S.A., Li, C.L., Yoder, N., Arik, S.O., Pfister, T.: Tsmixer: An all-mlp architecture for time series forecasting (2023), <https://arxiv.org/abs/2303.06053>
4. Deepika, T.: Analysis and comparison of different wavelet transform methods using benchmarks for image fusion (2020), <https://arxiv.org/abs/2007.11488>
5. Gabry, M.A., Eltaleb, I., Soliman, M.Y., Farouq-Ali, S.M.: A new technique for estimating stress from fracture injection tests using continuous wavelet transform. *Energies* **16**(2), 764 (2023), <https://www.mdpi.com/1996-1073/16/2/764>
6. Gao, J., Song, X., Wen, Q., Wang, P., Sun, L., Xu, H.: Robusttad: Robust time series anomaly detection via decomposition and convolutional neural networks (2021), <https://arxiv.org/abs/2002.09545>
7. Guennec, A.L., Malinowski, S., Tavenard, R.: Data augmentation for time series classification using convolutional neural networks (2016), <https://api.semanticscholar.org/CorpusID:3907864>
8. Kirichenko, L., Radivilova, T., Pichugina, O., Chala, L., Khandak, D.: Trend detection in short time series using discrete wavelet transform. In: *Proceedings of the International Workshop on Trends in Machine Learning and Data Analysis*. p. Page numbers. Kharkiv, Ukraine (Month 2022), <https://ceur-ws.org/Vol-3347>
9. Kuyuk, H.: On the use of stockwell transform in structural dynamic analysis. *Sadhana* **40**, 295–306 (February 2015). <https://doi.org/10.1007/s12046-014-0301-2>
10. Lai, G., Chang, W.C., Yang, Y., Liu, H.: Modeling long- and short-term temporal patterns with deep neural networks (2018), <https://arxiv.org/abs/1703.07015>
11. Liashchynskyi, P., Liashchynskyi, P.: Grid search, random search, genetic algorithm: A big comparison for nas (2019), <https://arxiv.org/abs/1912.06059>
12. Percival, D.B., Mondal, D.: A wavelet variance primer. In: Rao, T.S., Rao, S.S., Rao, C.R. (eds.) *Handbook of Statistics*, vol. 30, pp. 623–657. Elsevier (2012). <https://doi.org/10.1016/B978-0-444-53858-1.00022-3>, <https://www.sciencedirect.com/science/article/pii/B9780444538581000223>
13. Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. *Journal of Big Data* **6**(60) (2019). <https://doi.org/10.1186/s40537-019-0197-0>
14. Um, T.T., Pfister, F.M.J., Pichler, D., Endo, S., Lang, M., Hirche, S., Fietzek, U., Kulić, D.: Data augmentation of wearable sensor data for parkinson’s disease monitoring using convolutional neural networks. In: *Proceedings of the 19th ACM International Conference on Multimodal Interaction (ICMI ’17)*. pp. 1–10. ACM (2017). <https://doi.org/10.1145/3136755.3136817>
15. Wen, Q., Sun, L., Yang, F., Song, X., Gao, J., Wang, X., Xu, H.: Time series data augmentation for deep learning: A survey. In: *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-2021)*. pp. 1–10. International Joint Conferences on Artificial Intelligence Organization (2021). <https://doi.org/10.24963/ijcai.2021/631>
16. Wen, T., Keyes, R.: Time series anomaly detection using convolutional neural networks and transfer learning (2019), <https://arxiv.org/abs/1905.13628>

17. Yoon, J., Jarrett, D., van der Schaar, M.: Time-series generative adversarial networks. In: *Advances in Neural Information Processing Systems*. vol. 32, pp. 5508–5518. NeurIPS Foundation (2019)
18. Yue, Z., Wang, Y., Duan, J., Yang, T., Huang, C., Tong, Y., Xu, B.: Ts2vec: Towards universal representation of time series. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. vol. 36, pp. 8980–8987. AAAI Press (2022). <https://doi.org/10.1609/aaai.v36i8.20881>
19. Zeng, A., Chen, M., Zhang, L., Xu, Q.: Are transformers effective for time series forecasting? (2022), <https://arxiv.org/abs/2205.13504>
20. Zhang, X., Chowdhury, R.R., Shang, J., Gupta, R., Hong, D.: Towards diverse and coherent augmentation for time-series forecasting (2023), <https://arxiv.org/abs/2303.14254>
21. Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., Zhang, W.: Informer: Beyond efficient transformer for long sequence time-series forecasting. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. vol. 35, pp. 11106–11115. AAAI Press (2021). <https://doi.org/10.1609/aaai.v35i12.17325>