WPMixer: Efficient Multi-Resolution Mixing for Long-Term Time Series Forecasting

Md Mahmuddun Nabi Murad[1], **Mehmet Aktukmak**[2], **Yasin Yilmaz**[1]

[1]University of South Florida
[2]Intel Corporation
mmurad@usf.edu, mehmet.aktukmak@intel.com, yasiny@usf.edu

Abstract

Time series forecasting is crucial for various applications, such as weather forecasting, power load forecasting, and financial analysis. In recent studies, MLP-mixer models for time series forecasting have been shown as a promising alternative to transformer-based models. However, the performance of these models is still yet to reach its potential. In this paper, we propose Wavelet Patch Mixer (WPMixer), a novel MLP-based model, for long-term time series forecasting, which leverages the benefits of patching, multi-resolution wavelet decomposition, and mixing. Our model is based on three key components: (i) multi-resolution wavelet decomposition, (ii) patching and embedding, and (iii) MLP mixing. Multi-resolution wavelet decomposition efficiently extracts information in both the frequency and time domains. Patching allows the model to capture an extended history with a look-back window and enhances capturing local information while MLP mixing incorporates global information. Our model significantly outperforms state-of-the-art MLP-based and transformer-based models for long-term time series forecasting in a computationally efficient way, demonstrating its efficacy and potential for practical applications.

Introduction

Typically, time series data volume accumulates to vast amounts in various applications due to recording observations and events over long time horizons. The study of predicting time series data has been essential because of its extensive use in various domains such as finance, weather forecasting, and energy consumption prediction.

While research in time-series forecasting, for a long time, relied on traditional statistical methods such as ARIMA (Ariyo, Adewumi, and Ayo 2014), HMM (Hassan and Nath 2005), and SSM (Durbin and Koopman 2012), with the increasing availability of large datasets and high computational power, deep learning methods gained prevalence due to their superior performance in complex tasks. Specifically, RNN and CNN-based models like DeepAR (Salinas et al. 2020), and SCINet (Liu et al. 2022a), as well as transformer-based time series forecasting models, have become popular over time.

Transformer models for time series forecasting, such as Informer (Zhou et al. 2021), Autoformer (Wu et al. 2021), Fedformer (Zhou et al. 2022b), and Crossformer (Zhang and Yan 2023) have become popular thanks to their improved capability of learning long-term dependencies. However, recently, questions have arisen about the performance of the transformer variants in time series forecasting. The study (Zeng et al. 2023) demonstrated that a simple linear model can outperform or perform similarly with the state-of-the-art transformers on the popular benchmark datasets for time series forecasting.

Recently, MLP-based models have outperformed transformer variants in this domain. TimeMixer (Wang et al. 2024) and TSMixer (Chen et al. 2023) showed excellent prospects in multivariate time series forecasting. TSMixer, an MLP-mixer-based variant, mixes data in the time and channel domain but is computationally expensive for long-term forecasting due to a longer look-back window. TimeMixer, which achieves the state-of-the-art results on most benchmark datasets, decomposes a multi-scaled time series into seasonal and trend series using the moving average method and then employs the mixing among the mult-scaled data. However, due to complex seasonality patterns, decomposing a signal into seasonal and trend data is inadequate, and mixing among the multi-scaled data can cause information loss (Hyndman et al. 2011). Additionally, real-world time series data can have abrupt spikes and dips, which is difficult to explain using multi-scaled moving average-based decomposition techniques. Furthermore, capturing the information only in the time domain is not sufficient due to the complex nature of the time series data. SWformer, a variant of Sepformer (Fan et al. 2022), extracts information in the time and frequency domain utilizing wavelet transform-based decomposition. However, a multi-level wavelet transform is required to achieve its full potential.

To address these challenges, we propose a novel MLP-mixer-based model, called Wavelet Patch Mixer (WPMixer). What sets our model apart is its ability to capture intricate information in both the time and frequency domains, achieved through the use of multi-level wavelet decomposition. WPMixer decomposes the time series into multiple approximation and detail coefficient series using the multi-level wavelet transform. Distinct resolution branches handle each coefficient series, preventing information loss from mixing among multiple coefficient series. We utilize patch-

ing to capture local information and reduce the computational cost. We also employ patch mixer followed by embedding mixer to capture global information. Our contributions can be summarized as follows:

- We propose a novel model consisting of three core parts. Multi-level wavelet decomposition enables utilizing time and frequency domain properties due to spikes and dips, which cannot be captured by moving average-based decomposition methods in the time domain. Patching and mixing, on the other hand, capture local and global information, respectively.
- We analyze each decomposed series using a distinct resolution branch. This approach ensures that information from each resolution is maintained separately, thereby minimizing potential information loss.
- We enhance the performance of the patch mixer by applying an embedding mixer after each patch mixer.
- Our model, WPMixer, efficiently achieves state-of-the-art performance in long-term forecasting on several benchmark datasets.

Related Work

Time series forecasting refers to predicting a sequence of values in a time series based on a past sequence. Research on time series forecasting considers both long-term and short-term forecasting tasks.

Transformer-based models have recently shown remarkable performance in long-term forecasting. Informer (Zhou et al. 2021) applies prob-sparse attention with distill operation. Autoformer (Wu et al. 2021) improves Informer by applying decomposition in the transformer architecture. They decompose time series into seasonal and trend patterns with auto-correlation mechanisms based on time series periodicity. Sepformer (Fan et al. 2022) and FEDformer (Zhou et al. 2022b) are other transformer models which use decomposition techniques for long-term time series forecasting. Sepformer uses a single-level wavelet decomposition, in which wavelet coefficients are processed by a transformer. FEDformer enhances the time domain features using Fourier and wavelet transforms. In addition to the enhancement method, they also utilize separate attention mechanisms for Fourier and wavelet decomposed data. The Crossformer (Zhang and Yan 2023) model employes a dual-stage attention mechanism to capture dependencies across time and variables. In (Liu et al. 2022b), a non-stationary transformer is proposed with de-stationary attention to address the over-stationarization problem. In the framework of PatchTST (Nie et al. 2023), a conventional transformer augmented with patching is introduced to address the challenge of minimizing computational complexity while effectively capturing local semantic information. iTransformer (Liu et al. 2024), an exclusively encoder-based transformer architecture, adopts a strategy of tokenizing each variate series individually rather than processing multivariate data at a single time step. This approach facilitates the computation of mutual attention across the multivariate series.

FiLM (Zhou et al. 2022a) modifies the time series by transforming it into a Legendre polynomial space, thereby preserving the memory of long-term historical data. This method employs a frequency-enhanced operation akin to that used by FEDFormer (Zhou et al. 2022b) to accomplish the enhancement of the time series data. MICN (Wang et al. 2023) employs multiscale hybrid decomposition to analyze seasonal and trend components. Forecasting seasonal series is conducted using a convolutional neural network (CNN) model, which implements a convolutional kernel in the time domain. Trend prediction is achieved through a regression-based approach. TimesNet (Wu et al. 2023) utilizes the Fast Fourier transform to derive multiple periods for transforming time series data, thereby elucidating inter-period and intra-period variations within the series. In (Zeng et al. 2023), authors presents a group of linear models to demonstrate the effectiveness of simple linear models against the transformer-based model.

Recently, MLP-Mixer models have also been shown to provide effective solutions for time series forecasting despite being initially proposed for vision-based tasks (Tolstikhin et al. 2021). This potential is further demonstrated in TSMixer (Chen et al. 2023) and TimeMixer (Wang et al. 2024), where the mixer model is shown to outperform the transformer-based methods on the popular benchmark datasets. TSMixer has the same architecture as the original MLP-Mixer (Tolstikhin et al. 2021), but instead of mixing in the patch and channel domain, it mixes data in the time and channel domain directly. TimeMixer obtains a multi-scaled time series by applying down-sampling, then decomposes the multi-scaled time series into seasonal and trend series and mixes the data.

In our proposed WPMixer model, we improve the performance of the MLP-mixer-based models by employing multi-level wavelet decomposition with patching and mixing.

Proposed Method

Given a multivariate time series $\boldsymbol{X}_L = \{\boldsymbol{x}_{t-L+1}, \ldots, \boldsymbol{x}_{t-1}, \boldsymbol{x}_t\}$, with a look-back window L, at time step t, we aim to forecast the subsequent T data points $\boldsymbol{X}_T = \{\boldsymbol{x}_{t+1}, \boldsymbol{x}_{t+2}, \ldots, \boldsymbol{x}_{t+T}\}$, where $\boldsymbol{x}_t \in \mathbb{R}^{1\times C}$ denotes a multivariate data point at time t, C is the number of the variates, and T is the prediction length.

Model Architecture

The architecture of the proposed model is illustrated in Figure 1. Our approach begins with decomposing the normalized time series data into approximation and detail coefficient series through multi-level wavelet decomposition. This multi-level decomposition facilitates feature extraction from the time series data at various resolutions, where each resolution represents a distinct frequency level. As we progress to higher decomposition levels, the frequency range of the approximation coefficients becomes narrower. At the same time, we get multiple detail coefficient series that represent detailed information at various frequency levels. However, higher-level coefficient series may not always yield relevant information for forecasting tasks. Additionally, different wavelets offer varying trade-offs between time and frequency localization, making the selection of an optimal de-

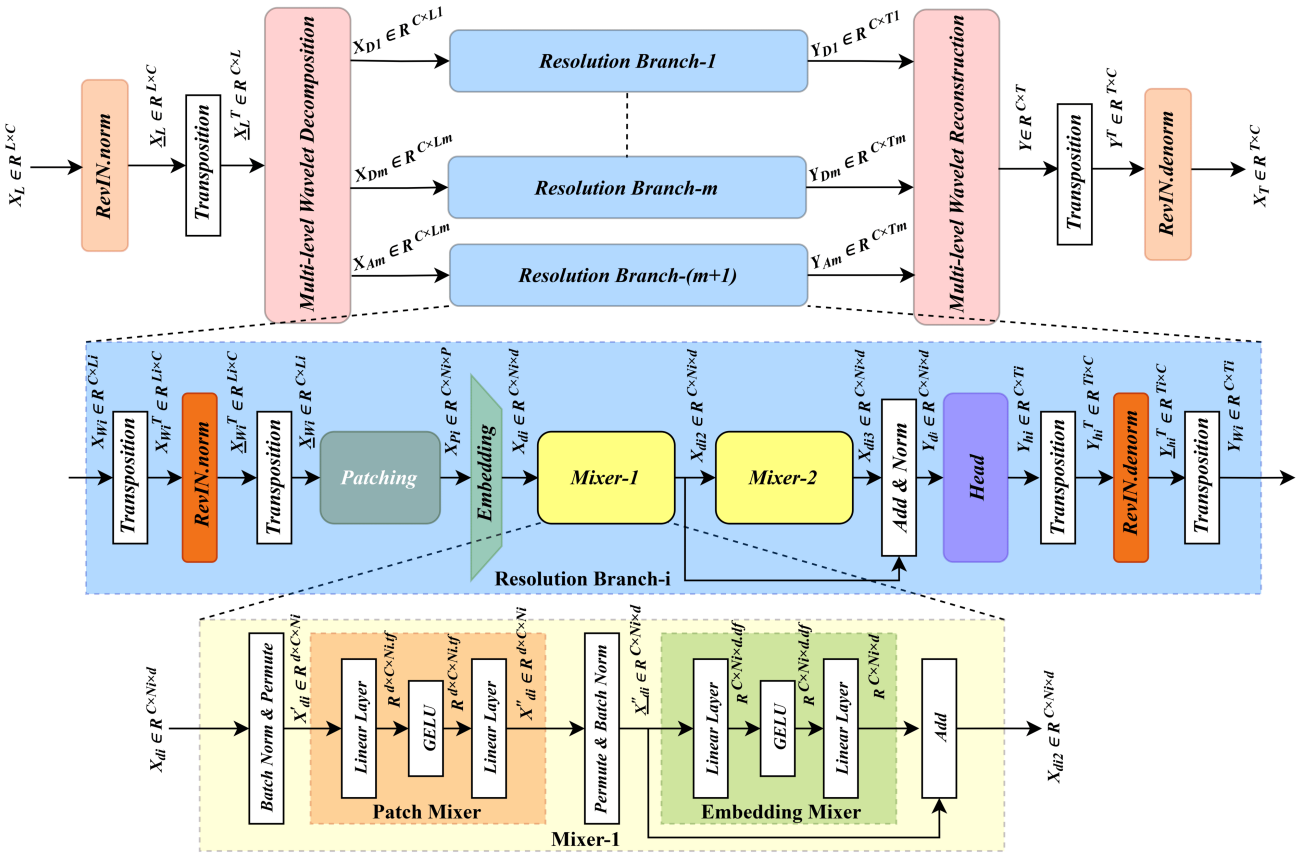


Figure 1: WPMixer with m levels of wavelet decomposition. $\boldsymbol{X}_{A_i}$ and $\boldsymbol{X}_{D_i}$ are the approximation and detail coefficient series corresponding to the input time series $\boldsymbol{X}_L$. $\boldsymbol{Y}_{A_i}$ and $\boldsymbol{Y}_{D_i}$ are the predicted approximation and detail coefficient series corresponding to the predicted time series $\boldsymbol{X}_T$. To simplify notation, $\boldsymbol{X}_{W_i}$ denotes either $\boldsymbol{X}_{A_i}$ or $\boldsymbol{X}_{D_i}$. Code is available at https://github.com/Secure-and-Intelligent-Systems-Lab

composition level and wavelet type a crucial aspect of the optimization process.

Our model processes each wavelet coefficient series through a distinct resolution branch, which prevents the intermixing of information across different frequency scales. Each resolution branch comprises an instance normalization module, a patch and embedding module, several mixer modules, a head module, and an instance denormalization module. The patch and embedding module transforms the normalized wavelet coefficient series into a series of patches. The patch mixer modules then aggregate the local information contained within these patches into a global information context. In the mixer module, which is a fusion of a patch mixer and an embedding mixer, the embedding mixer captures the global information in a higher dimensional space. The head module subsequently forecasts the wavelet coefficient series, providing the information needed for predicting the time series. A denormalization layer is employed to reintegrate the stationary information into the predicted wavelet coefficient series. Finally, the multi-level wavelet reconstruction module reconstructs the predicted time series by utilizing the predicted approximation and detail wavelet coefficient series. In the following subsections, we describe the key modules of our model.

Instance Normalization: One of the main challenges for time series forecasting is to deal with the time-varying mean and variation. To overcome this challenge, Reversible Instance Normalization (RevIN) with learnable affine transform has been proposed in (Kim et al. 2021). We initially employ RevIN normalization and denormalization directly in the time series data before decomposition and after reconstruction, respectively. We also employ RevIN normalization and denormalization in the wavelet coefficient series. The positions of the RevIN normalization and denormalization layers are shown in Fig 1.

Decomposition: We utilize the multi-level discrete wavelet transform to decompose the time series data. This transformation involves an iterative decomposition process utilizing high-pass and low-pass filters to extract wavelet coefficients at multiple levels (Mallat 1989). The coefficients of the filters depend on the type of wavelet. The output of the high-pass filter refers to detailed information, called detail coefficients, whereas the output of the low-pass filter refers to low-frequency information, called approximation coefficients. At each level, the approximation coefficients from the preceding level is split into new approximation and detail coefficients, allowing for a deeper data analysis. We

adapt the implementation of the multi-level discrete wavelet transform from (Cotter 2019) to work with PyTorch mixed precision analysis.

The decomposition module disintegrates the normalized time series $\underline{\boldsymbol{X}}_L^T \in \mathbb{R}^{C\times L}$ into approximation and detail coefficient series:

$$[\boldsymbol{X}_{A_m}, \boldsymbol{X}_{D_m}, \boldsymbol{X}_{D_{m-1}}, \ldots, \boldsymbol{X}_{D_1}] = Decomp(\underline{\boldsymbol{X}}_L^T, \psi, m). \tag{1}$$

In this context, m denotes the decomposition level, ψ denotes the wavelet type, $\boldsymbol{X}_{A_i} \in \mathbb{R}^{C\times L_i}$ and $\boldsymbol{X}_{D_i} \in \mathbb{R}^{C\times L_i}$ represent the approximation and detail coefficient series at the i-th level of decomposition, respectively. Here, L_i indicates the number of wavelet coefficients in the coefficient series at the i-th decomposition level. To avoid information redundancy, we retain only the approximation coefficient series from the final level m while discarding those from levels 1 through $(m-1)$, as they are further decomposed into new approximation and detail coefficient series. However, we include the detail coefficient series from all levels in our analysis. In our experiments, we optimized the wavelet type by considering the Daubechies, Symlets, Coiflets, and Biorthogonal wavelet families.

Each series of wavelet coefficient is processed through a distinct resolution branch within the model, encompassing a RevIN normalization module, a patching and embedding module, multiple mixer modules, a head module, and a RevIN denormalization module. The total number of multivariate coefficient series or resolution branches in the model is given by $(m+1)$ due to the m detail and 1 approximation coefficient series.

To simplify the notation, we will refer both the approximation coefficient series $\boldsymbol{X}_{A_i}$ and the detail coefficient series $\boldsymbol{X}_{D_i}$ with $\boldsymbol{X}_{W_i} \in \mathbb{R}^{C\times L_i}$ in the following steps.

Patching and Embedding Module: To capture the local information efficiently, we adopt patching and embedding techniques from (Nie et al. 2023). Each normalized univariate wavelet coefficient series $\underline{\boldsymbol{X}}_{W_i}^{(j)} \in \mathbb{R}^{1\times L_i},\ j = 1, \ldots, C$, is divided into overlapping patches of length P. The non-overlapping portion is denoted as stride S. Before patching, $\underline{\boldsymbol{X}}_{W_i}^{(j)}$ is padded with S number of repeated last values of the sequence $\underline{\boldsymbol{X}}_{W_i}^{(j)}$. So, each univariate wavelet coefficient series $\underline{\boldsymbol{X}}_{W_i}^{(j)}$ is converted to $\boldsymbol{X}_{P_i}^{(j)} \in \mathbb{R}^{1\times N_i\times P}$, where $N_i = \frac{(L_i - P)}{S} + 2$ is the number of patches.

The multivariate output of the patching block,

$$\boldsymbol{X}_{P_i} = Patch(\underline{\boldsymbol{X}}_{W_i}) \in \mathbb{R}^{C\times N_i\times P} \tag{2}$$

is passed through a linear embedding layer to encode into d dimensions. This embedding layer is shareable across all variates of $\boldsymbol{X}_{P_i}$, i.e.,

$$\boldsymbol{X}_{d_i} = Embedding(\boldsymbol{X}_{P_i}) \in \mathbb{R}^{C\times N_i\times d}. \tag{3}$$

Mixer Module: The Mixer module consists of two primary components, the Patch Mixer and a subsequent Embedding Mixer. The Patch Mixer functions similarly to the token-mixing MLP as outlined in (Tolstikhin et al. 2021).

Before intermixing information across the patch dimension, 2D-Batch normalization followed by dimension permutation operation is applied on $\boldsymbol{X}_{d_i} \in \mathbb{R}^{C\times N_i\times d}$. Within the patch mixer, two linear layers are employed alongside the GELU activation function. The first layer expands the dimensionality with factor t_f while the subsequent layer restores it to its original dimension. The operations in the patch mixer can be summarized as,

$$\boldsymbol{X}_{d_i}^{'} = \mathcal{P}(BN(\boldsymbol{X}_{d_i})) \in \mathbb{R}^{d\times C\times N_i} \tag{4}$$

$$\boldsymbol{X}_{d_i}^{''} = \mathcal{L}_2(\mathcal{G}(\mathcal{L}_1(\boldsymbol{X}_{d_i}^{'}))) \in \mathbb{R}^{d\times C\times N_i} \tag{5}$$

where $BN(.)$ represents the 2D-Batch normalization, $\mathcal{P}(.)$ represents dimension permutation, $\mathcal{G}(.)$ represents GELU activation, $\mathcal{L}_1 : \mathbb{R}^{d\times C\times N_i} \rightarrow \mathbb{R}^{d\times C\times N_i.t_f}$ represents layer-1 and $\mathcal{L}_2 : \mathbb{R}^{d\times C\times N_i.t_f} \rightarrow \mathbb{R}^{d\times C\times N_i}$ represents layer-2 in the patch mixer MLP.

Prior to processing in the Embedding Mixer, $\boldsymbol{X}_{d_i}^{''}$ is subjected to dimension permutation and 2D-Batch normalization. In the Embedding Mixer, $\underline{\boldsymbol{X}_{d_i}^{''}}$ traverses two linear layers incorporating GELU activation similarly to the Patch Mixer. However, the initial layer increases the embedding dimensionality d with factor d_f, while the subsequent layer restores it to its original dimension. Different than Patch Mixer, a residual connection is also included with the MLP. The operations in the Embedding Mixer can be summarized as,

$$\underline{\boldsymbol{X}_{d_i}^{''}} = BN(\mathcal{P}(\boldsymbol{X}_{d_i}^{''})) \in \mathbb{R}^{C\times N_i\times d} \tag{6}$$

$$\boldsymbol{X}_{d_{i2}} = \underline{\boldsymbol{X}_{d_i}^{''}} + \mathcal{L}^{'}{}_2(\mathcal{G}(\mathcal{L}^{'}{}_1(\underline{\boldsymbol{X}_{d_i}^{''}}))) \in \mathbb{R}^{C\times N_i\times d}, \tag{7}$$

where $\mathcal{L}^{'}{}_1 : \mathbb{R}^{C\times N_i\times d} \rightarrow \mathbb{R}^{C\times N_i\times d.d_f}$ represents layer-1 and $\mathcal{L}^{'}{}_2 : \mathbb{R}^{C\times N_i\times d.d_f} \rightarrow \mathbb{R}^{C\times N_i\times d}$ represents layer-2. Two sequential Mixer modules are employed in our model, where the second Mixer module has a residual connection followed by 2D-Batch normalization.

Head Module: The Head module comprises a flatten and a linear projection layers. The flatten layer flattens the last two dimensions of the input $Y_{d_i} \in \mathbb{R}^{C\times N_i\times d}$.

$$\boldsymbol{Y}_{f_i} = Flatten(\boldsymbol{Y}_{d_i}) \in \mathbb{R}^{C\times N_i.d}, \tag{8}$$

and the linear layer transforms $\boldsymbol{Y}_{f_i}$ to

$$\boldsymbol{Y}_{h_i} = Linear(\boldsymbol{Y}_{f_i}) \in \mathbb{R}^{C\times T_i}, \tag{9}$$

where T_i is the prediction length of the wavelet coefficient series. To determine the value of T_i, an auxiliary time series of equivalent length to the predicted series $\boldsymbol{X}_T$ undergoes the decomposition module while initializing the model. T_i is set as the length of the auxiliary decomposed wavelet coefficient series.

Reconstruction: The Reconstruction module can be described as,

$$\boldsymbol{Y} = Reconstruction_{\psi}(\boldsymbol{Y}_{A_m}, \boldsymbol{Y}_{D_m}, \boldsymbol{Y}_{D_{m-1}}, \ldots, \boldsymbol{Y}_{D_1}); \tag{10}$$

where $\boldsymbol{Y}_{A_i} \in \mathbb{R}^{C\times T_i}$ and $\boldsymbol{Y}_{D_i} \in \mathbb{R}^{C\times T_i}$ are the predicted approximation and detail wavelet coefficient series. $\boldsymbol{Y} \in \mathbb{R}^{C\times T}$ is the reconstructed time series, which is transformed by instance denormalization to obtain the final prediction $\boldsymbol{X}_T \in \mathbb{R}^{T\times C}$.

Training: $SmoothL1Loss$ is employed to train our model with the default threshold value. Separate dropout values are used for the Embedding and Mixer modules. We used Optuna (Akiba et al. 2019) with the default setting of Tree-structured Parzen Estimator (TPE) for optimizing the hyperparameters. The optimized hyperparameter values are shown in Table 7 in (Murad, Aktukmak, and Yilmaz 2024).

Differences with the Existing Models

TimeMixer leverages moving average-based seasonal and trend decomposition of multi-scaled time series data and integrates data across multiple scales. WPMixer, on the other hand, employs multi-level wavelet transform-based decomposition, processing each coefficient series individually through a resolution branch. TSMixer incorporates time mixing and channel mixing while WPMixer employs patch mixing followed by embedding mixing. Both TimeMixer and TSMixer handle solely time-domain data, whereas WPMixer extracts features from both the time and frequency domains. Fedformer enhances time series using multi-wavelet transform, frequently converting data between the time and frequency domains. SWformer uses single-level wavelet transform for time series decomposition. However, WPMixer utilizes multi-level wavelet transform, which is computationally less expensive than multi-wavelet transform and more effective than single-level wavelet transform (Zhang and Zhang 2019). Additionally, WPMixer performs time series decomposition at the beginning of the model and reconstructs the series from the predicted coefficient series at the end, avoiding multiple conversions between the time and frequency domains.

Experiments

We extensively evaluate the long-term forecasting performance of WPMixer on 7 popular datasets: ETTh1, ETTh2, ETTm1, ETTm2, Weather, Electricity, and Traffic. The specifications of datasets are given in Table 1.

Baselines: We compare WPMixer with seven recent time series forecasting methods, namely TimeMixer (Wang et al. 2024)), TSMixer (Chen et al. 2023), TimesNet (Wu et al. 2023), FiLM (Zhou et al. 2022a), DLinear (Zeng et al. 2023), PatchTST (Nie et al. 2023), and Crossformer (Zhang and Yan 2023). TimeMixer and TSMixer, which can be considered as the state-of-the-art models based on their performances on the benchmark datasets, derive their architectures from the MLP-Mixer model while PatchTST and Crossformer utilize transformer architectures.

Dataset	Variates	Dataset Size	Freq.
ETTh1, ETTh2	7	(8545, 2881, 2881)	Hourly
ETTm1, ETTm2	7	(34465, 11521, 11521)	15 min
Weather	21	(36792, 5271, 10540)	10 min
Electricity	321	(18317, 2633, 5261)	Hourly
Traffic	862	(12185, 1757, 3509)	Hourly

Table 1: Specifications of the datasets. Dataset size refers to the training, validation, and testing dataset sizes.

Setup: Following the practice in Informer, Autoformer, PatchTST, TSMixer, and TimeMixer, all datasets were normalized to a zero mean and unit standard deviation. The normalized datasets served as the basis for ground truth in our evaluations. In long-term forecasting, the lengths of predictions were set at 96, 192, 336, and 720, in alignment with prior studies. During the training phase, SmoothL1Loss was employed, whereas Mean Squared Error (MSE) and Mean Absolute Error (MAE) were utilized for evaluation purposes. Experiments with the ETT and Weather datasets were performed on a single NVIDIA GeForce RTX 4090 GPU while the experiments with the Electricity and Traffic datasets were carried out using two NVIDIA A100 GPUs.

Multivariate Long-Term Forecasting Results

In long-term multivariate time series forecasting, existing studies employed distinct look-back window lengths to optimize performance. For a comprehensive comparison, we present our results under two experimental setups following TimeMixer (Wang et al. 2024).

In the first setup, we calibrated the look-back window length alongside other hyperparameters to enhance forecasting accuracy. We determined the optimal look-back window lengths for each dataset, exploring values of 96, 192, 336, 512, 1024, and 1200. The comprehensive results under this setup are presented in Table 2 while the optimized hyperparameter values and run information are given in Table 7 in (Murad, Aktukmak, and Yilmaz 2024). The performance of other models listed in Table 2 are also their optimized results (Wang et al. 2024). Our analysis revealed that our model's performance is notably superior compared to its counterparts. Specifically, our model decreased MSE on average across the ETTh1, ETTh2, ETTm1, and ETTm2 datasets by 7.8%, 2.2%, 3.4%, and 3.9%, respectively. Similarly, MAE was reduced by 3.3%, 6.4%, 0.5%, and 2.5%, respectively, for these datasets. On the Weather and Traffic datasets, our model demonstrated lower MSE and MAE in average prediction relative to the state-of-the-art TimeMixer model. Moreover, on the Electricity dataset, our model achieved the highest performance following the TimeMixer model.

In the second setup, we followed the unified setting of TimeMixer for all the datasets. The detailed results are presented in Table 9 in (Murad, Aktukmak, and Yilmaz 2024). We achieved lower MSE and MAE scores on average on the ETT and Electricity datasets compared to the TimeMixer model.

Computational Efficiency and Robustness

We evaluate WPMixer's computational cost in terms of the number of giga floating point operations (GFLOPs), a hardware-independent metric. We compute the GFLOPs for WPMixer and TimeMixer using the unified setting outlined by (Wang et al. 2024) with embedding dimension $d = 16$ for the ETTh1 dataset. The comparison is presented in Table 3. WPMixer consistently requires less than one tenth GFLOPs across all prediction lengths compared to TimeMixer.

We also evaluate our model with three different random seeds by computing the mean and standard deviation for MSE and MAE. Results are averaged over the prediction

Models		WPMixer (Ours)		TimeMixer* 2024		PatchTST 2023		TSMixer 2023		TimesNet 2023		Crossformer* 2023		FiLM* 2022a		Dlinear* 2023	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.347**	**0.383**	0.361	0.390	0.370	0.400	0.361	0.392	0.384	0.402	0.418	0.438	0.422	0.432	0.375	0.399
	192	**0.381**	**0.408**	0.409	0.414	0.413	0.429	0.404	0.418	0.436	0.429	0.539	0.517	0.462	0.458	0.405	0.416
	336	**0.382**	**0.412**	0.430	0.429	0.422	0.440	0.420	0.431	0.491	0.469	0.709	0.638	0.501	0.483	0.439	0.443
	720	**0.405**	**0.432**	0.445	0.460	0.447	0.468	0.463	0.472	0.521	0.500	0.733	0.636	0.544	0.526	0.472	0.490
	Avg	**0.379**	**0.409**	0.411	0.423	0.413	0.434	0.412	0.428	0.458	0.450	0.600	0.557	0.482	0.475	0.423	0.437
ETTh2	96	**0.253**	**0.328**	0.271	0.330	0.274	0.337	0.274	0.341	0.340	0.374	0.425	0.463	0.323	0.370	0.289	0.353
	192	**0.303**	**0.364**	0.317	0.402	0.341	0.382	0.339	0.385	0.402	0.414	0.473	0.500	0.391	0.415	0.383	0.418
	336	**0.305**	**0.371**	0.332	0.396	0.329	0.384	0.361	0.406	0.452	0.452	0.581	0.562	0.415	0.440	0.448	0.465
	720	0.373	0.417	**0.342**	**0.408**	0.379	0.422	0.445	0.470	0.462	0.468	0.775	0.665	0.441	0.459	0.605	0.551
	Avg	**0.309**	**0.370**	0.316	0.384	0.331	0.381	0.355	0.401	0.414	0.427	0.564	0.548	0.393	0.421	0.431	0.447
ETTm1	96	**0.275**	**0.333**	0.291	0.340	0.293	0.346	0.285	0.339	0.338	0.375	0.361	0.403	0.302	0.345	0.299	0.343
	192	**0.319**	**0.362**	0.327	0.365	0.333	0.370	0.327	0.365	0.374	0.387	0.387	0.422	0.338	0.368	0.335	0.365
	336	**0.347**	0.384	0.360	**0.381**	0.369	0.392	0.356	0.382	0.410	0.411	0.605	0.572	0.373	0.388	0.369	0.386
	720	**0.403**	**0.414**	0.415	0.417	0.416	0.420	0.419	**0.414**	0.478	0.450	0.703	0.645	0.420	0.420	0.425	0.421
	Avg	**0.336**	**0.373**	0.348	0.375	0.353	0.382	0.347	0.375	0.400	0.406	0.514	0.510	0.358	0.380	0.357	0.379
ETTm2	96	**0.159**	**0.246**	0.164	0.254	0.166	0.256	0.163	0.252	0.187	0.267	0.275	0.358	0.165	0.256	0.167	0.260
	192	**0.214**	**0.286**	0.223	0.295	0.223	0.296	0.216	0.290	0.249	0.309	0.345	0.400	0.222	0.296	0.224	0.303
	336	**0.266**	**0.322**	0.279	0.330	0.274	0.329	0.268	0.324	0.321	0.351	0.657	0.528	0.277	0.333	0.281	0.342
	720	**0.344**	**0.374**	0.359	0.383	0.362	0.385	0.420	0.422	0.408	0.403	1.208	0.753	0.371	0.389	0.397	0.421
	Avg	**0.246**	**0.307**	.256	0.315	0.256	0.317	0.267	0.322	0.291	0.333	0.621	0.510	0.259	0.319	0.267	0.332
Weather	96	**0.141**	**0.188**	0.147	0.197	0.149	0.198	0.145	0.198	0.172	0.220	0.232	0.302	0.199	0.262	0.176	0.237
	192	**0.185**	**0.229**	0.189	0.239	0.194	0.241	0.191	0.242	0.219	0.261	0.371	0.410	0.228	0.288	0.220	0.282
	336	**0.236**	**0.271**	0.241	0.280	0.245	0.282	0.242	0.280	0.280	0.306	0.495	0.515	0.267	0.323	0.265	0.319
	720	**0.307**	**0.321**	0.310	0.330	0.314	0.334	0.320	0.336	0.365	0.359	0.526	0.542	0.319	0.361	0.323	0.362
	Avg	**0.217**	**0.252**	0.222	0.262	0.226	0.264	0.225	0.264	0.259	0.287	0.406	0.442	0.253	0.309	0.246	0.300
Electricity	96	**0.128**	**0.222**	0.129	0.224	0.129	**0.222**	0.131	0.229	0.168	0.272	0.150	0.251	0.154	0.267	0.140	0.237
	192	0.145	0.237	**0.140**	**0.220**	0.147	0.240	0.151	0.246	0.184	0.289	0.161	0.260	0.164	0.258	0.153	0.249
	336	**0.161**	0.256	**0.161**	**0.255**	0.163	0.259	**0.161**	0.261	0.198	0.300	0.182	0.281	0.188	0.283	0.169	0.267
	720	0.196	**0.287**	**0.194**	**0.287**	0.197	0.290	0.197	0.293	0.220	0.320	0.251	0.339	0.236	0.332	0.203	0.301
	Avg	0.158	0.251	**0.156**	**0.246**	0.159	0.253	0.160	0.257	0.192	0.295	0.186	0.283	0.186	0.285	0.166	0.264
Traffic	96	**0.354**	**0.246**	0.360	0.249	0.360	0.249	0.376	0.264	0.593	0.321	0.514	0.267	0.416	0.294	0.410	0.282
	192	**0.371**	0.253	0.375	**0.250**	0.379	0.256	0.397	0.277	0.617	0.336	0.549	0.252	0.408	0.288	0.423	0.287
	336	0.387	0.267	**0.385**	0.270	0.392	**0.264**	0.413	0.290	0.629	0.336	0.530	0.300	0.425	0.298	0.436	0.296
	720	0.431	0.289	**0.430**	**0.281**	0.432	0.286	0.444	0.306	0.640	0.350	0.573	0.313	0.520	0.353	0.466	0.315
	Avg	**0.386**	0.264	0.387	**0.262**	0.391	0.264	0.408	0.284	0.620	0.336	0.542	0.283	0.442	0.308	0.434	0.295
	1st Count:	29	26	7	9	0	2	1	1	0	0	0	0	0	0	0	0

Table 2: Multivariate long-term forecasting results. Four commonly used prediction lengths (96,192,336,720) from the literature are considered for each dataset. The length of the look-back window is a hyperparameter. The results of the models marked with $*$ are taken from (Wang et al. 2024); other results are taken from the corresponding papers.

		WPMixer			TimeMixer		
	T	MSE	MAE	GFLOPs	MSE	MAE	GFLOPs
ETTh1	96	0.370	0.390	0.210	0.375	0.400	2.774
	192	0.424	0.420	0.226	0.429	0.421	3.281
	336	0.462	0.433	0.211	0.484	0.458	4.040
	720	0.455	0.449	0.481	0.498	0.482	6.066

Table 3: WPMixer is ten folds more efficient for $d = 16$.

lengths of 96, 192, 336, and 720. As shown in Table 4, our model exhibits a lower standard deviation than TimeMixer in all cases, highlighting the robustness of our approach.

Ablation Study

WPMixer Modules: We conducted an extensive ablation study to evaluate the individual contribution of each module within the proposed model using the ETT datasets. This analysis consists of fourteen distinct cases, each exploring a different combination of the modules. Case-I represents the foundational architecture of WPMixer. The details of the other cases are delineated in Table 5. For each case, we performed a thorough search of optimum hyperparameters utilizing Optuna. The results in Table 5 demonstrate the importance of all proposed modules.

Effect of Multiple Levels of Decomposition: We assessed the impact of multi-level decomposition by varying m from 1 to 5. The other parameters are kept fixed for all m as follows, look-back window 512, initial learning rate

	WPMixer		TimeMixer	
	MSE	MAE	MSE	MAE
(1)	0.422 ± 0.001	0.423 ± 0.001	0.447 ± 0.002	0.440 ± 0.005
(2)	0.355 ± 0.003	0.387 ± 0.001	0.364 ± 0.008	0.395 ± 0.010
(3)	0.376 ± 0.002	0.388 ± 0.001	0.381 ± 0.003	0.395 ± 0.006
(4)	0.271 ± 0.001	0.317 ± 0.001	0.275 ± 0.001	0.323 ± 0.003
(5)	0.243 ± 0.001	0.269 ± 0.000	0.240 ± 0.010	0.271 ± 0.009
(6)	0.177 ± 0.000	0.267 ± 0.000	0.182 ± 0.017	0.272 ± 0.006
(7)	0.489 ± 0.005	0.297 ± 0.001	0.484 ± 0.015	0.297 ± 0.013

Table 4: Model robustness under the unified setting, including similar look-back window length, batch size, and epochs for all models. (1), (2), (3), (4), (5), (6), and (7) refer to ETTh1, ETTh2, ETTm1, ETTm2, Weather, Electricity, and Traffic datasets, respectively.

	Modules						ETTh1	ETTh2	ETTm1	ETTm2
Case	D	P	E	P_x	E_x	H	MSE	MSE	MSE	MSE
I	✓	✓	✓	✓	✓	✓	0.379	**0.308**	**0.336**	**0.245**
II	×	✓	✓	✓	✓	✓	0.388	0.311	0.339	0.247
III	✓	×	×	✓	✓	✓	0.384	0.316	0.339	0.250
IV	×	×	×	✓	✓	✓	0.392	0.325	0.345	0.249
V	✓	✓	×	✓	✓	✓	0.378	0.314	0.339	0.247
VI	×	✓	×	✓	✓	✓	0.390	0.320	0.343	0.248
VII	✓	✓	✓	×	×	✓	0.394	0.311	0.353	0.252
VIII	×	✓	✓	×	×	✓	0.399	0.312	0.354	0.252
IX	✓	×	×	×	×	✓	0.400	0.315	0.356	0.251
X	×	×	×	×	×	✓	0.403	0.315	0.355	0.252
XI	✓	✓	×	×	×	✓	0.400	0.312	0.355	0.251
XII	×	✓	×	×	×	✓	0.403	0.314	0.355	0.252
XIII	✓	✓	✓	×	✓	✓	**0.377**	0.314	0.339	0.247
XIV	×	✓	✓	×	✓	✓	0.392	0.314	0.342	0.248

Table 5: Contribution of each module in WPMixer. D, P, E, P_x, E_x, and H refer to the decomposition, patch, embedding, patch mixer, embedding mixer, and head modules, respectively. Look-back window is set to 512. Results are averaged over the prediction lengths 96, 192, 336, and 720.

0.001, wavelet type Daubechies 5, batch size 128, epochs 10, $d = 256$, $t_f = 7$, $d_f = 7$, patch size 16, and stride 8. MSE performances for prediction lengths of 336 and 720 on the ETTh datasets are presented in Figure 2. The results indicate that the optimal level m depends on the prediction length and dataset. Consequently, we treated m as a hyperparameter in our model and performed a search to identify its optimal value for every experiment.

SmoothL1 vs MSE Loss: In our experiments, we utilized the $SmoothL1$ loss as the primary loss function instead of the traditional MSE loss. We conducted an ablation study using the ETTh2 and ETTm2 datasets, employing an exhaustive search across the hyperparameter space. Detailed findings are presented in Table 6. Analysis of the results from Table 6 demonstrates that the adoption of the $SmoothL1$ loss improves the performance of our model.

Look-Back Window: We also evaluated the impact of look-back window size on the forecasting performance using the ETTh datasets, as illustrated in Figure 3. While in general the MSE value is reduced with increasing look-back

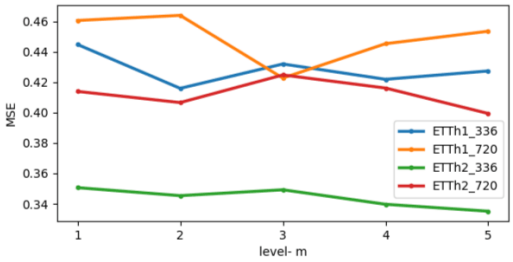
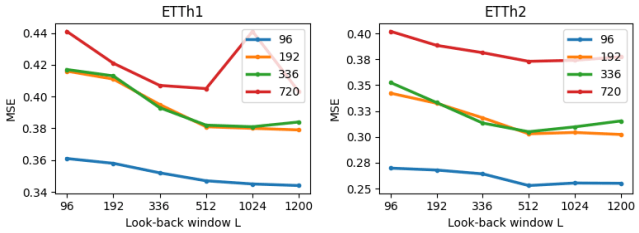


Figure 2: WPMixer performance with the varying level of the decomposition m.



(a) ETTh1 (b) ETTh2

Figure 3: Performance of the model with increasing look-back window length L.

	ETTm2				ETTh2			
	$MSEloss$		$SmoothL1$		$MSEloss$		$SmoothL1$	
T	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
96	0.165	0.257	0.159	0.246	0.251	0.327	0.253	0.328
192	0.219	0.291	0.214	0.286	0.308	0.365	0.303	0.364
336	0.271	0.327	0.266	0.322	0.306	0.373	0.305	0.371
720	0.349	0.384	0.344	0.374	0.374	0.419	0.373	0.417

Table 6: $SmoothL1$ loss vs. MSE loss for training.

window length, after a certain length, the model's performance stops improving or even degrades in some cases such as the prediction length of 336.

Conclusion

In this study, we introduced the Wavelet Patch Mixer (WPMixer), a computationally efficient long-term time series forecasting model. Our model utilizes multi-level wavelet decomposition to capture multi-resolution information in both the time and frequency domains. By incorporating patching for local information and a patch mixer for global information, we enhanced the model's capability to handle complex characteristics and abrupt spikes and dips in real-world data. The addition of an embedding mixer after each patch mixer further improved the model's forecasting performance. Our experimental results demonstrated that WPMixer achieves state-of-the-art performance efficiently in various long-term forecasting tasks. Through comprehensive experiments, we analyzed the model performance, computational cost, robustness to random initializations, effects of decomposition level, loss function, and look-back window size.

Acknowledgements

This work was supported by the U.S. National Institute of Food and Agriculture under Grant 2023-67019-38829.

References

Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; and Koyama, M. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2623–2631.

Ariyo, A. A.; Adewumi, A. O.; and Ayo, C. K. 2014. Stock price prediction using the ARIMA model. In *2014 UKSim-AMSS 16th international conference on computer modelling and simulation*, 106–112. IEEE.

Chen, S.-A.; Li, C.-L.; Arik, S. O.; Yoder, N. C.; and Pfister, T. 2023. TSMixer: An All-MLP Architecture for Time Series Forecast-ing. *Transactions on Machine Learning Research*.

Cotter, F. 2019. *Uses of Complex Wavelets in Deep Convolutional Neural Networks*. Ph.D. thesis, Apollo - University of Cambridge Repository.

Durbin, J.; and Koopman, S. J. 2012. *Time series analysis by state space methods*, volume 38. OUP Oxford.

Fan, J.; Wang, Z.; Sun, D.; and Wu, H. 2022. Sepformer-based models: More efficient models for long sequence time-series forecasting. *IEEE Transactions on Emerging Topics in Computing*.

Hassan, M. R.; and Nath, B. 2005. Stock market forecasting using hidden Markov model: a new approach. In *5th international conference on intelligent systems design and applications (ISDA'05)*, 192–196. IEEE.

Hyndman, R. J.; Ahmed, R. A.; Athanasopoulos, G.; and Shang, H. L. 2011. Optimal combination forecasts for hierarchical time series. *Computational statistics & data analysis*, 55(9): 2579–2589.

Kim, T.; Kim, J.; Tae, Y.; Park, C.; Choi, J.-H.; and Choo, J. 2021. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*.

Liu, M.; Zeng, A.; Chen, M.; Xu, Z.; Lai, Q.; Ma, L.; and Xu, Q. 2022a. Scinet: Time series modeling and forecasting with sample convolution and interaction. *Advances in Neural Information Processing Systems*, 35: 5816–5828.

Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; and Long, M. 2024. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. In *The Twelfth International Conference on Learning Representations*.

Liu, Y.; Wu, H.; Wang, J.; and Long, M. 2022b. Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in Neural Information Processing Systems*, 35: 9881–9893.

Mallat, S. G. 1989. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, 11(7): 674–693.

Murad, M. M. N.; Aktukmak, M.; and Yilmaz, Y. 2024. WPMixer: Efficient Multi-Resolution Mixing for Long-Term Time Series Forecasting. *arXiv preprint arXiv:2412.17176*.

Nie, Y.; Nguyen, N. H.; Sinthong, P.; and Kalagnanam, J. 2023. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *The Eleventh International Conference on Learning Representations*.

Salinas, D.; Flunkert, V.; Gasthaus, J.; and Januschowski, T. 2020. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International journal of forecasting*, 36(3): 1181–1191.

Tolstikhin, I. O.; Houlsby, N.; Kolesnikov, A.; Beyer, L.; Zhai, X.; Unterthiner, T.; Yung, J.; Steiner, A.; Keysers, D.; Uszkoreit, J.; et al. 2021. Mlp-mixer: An all-mlp architecture for vision. *Advances in neural information processing systems*, 34: 24261–24272.

Wang, H.; Peng, J.; Huang, F.; Wang, J.; Chen, J.; and 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, S.; Wu, H.; Shi, X.; Hu, T.; Luo, H.; Ma, L.; Zhang, J. Y.; and ZHOU, J. 2024. TimeMixer: Decomposable Multiscale Mixing for Time Series Forecasting. In *The Twelfth International Conference on Learning Representations*.

Wu, H.; Hu, T.; Liu, Y.; Zhou, H.; Wang, J.; and Long, M. 2023. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *The Eleventh International Conference on Learning Representations*.

Wu, H.; Xu, J.; Wang, J.; and Long, M. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34: 22419–22430.

Zeng, A.; Chen, M.; Zhang, L.; and Xu, Q. 2023. Are Transformers Effective for Time Series Forecasting? *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(9): 11121–11128.

Zhang, D.; and Zhang, D. 2019. Wavelet transform. *Fundamentals of image data mining: Analysis, Features, Classification and Retrieval*, 35–44.

Zhang, Y.; and Yan, J. 2023. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*.

Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; and Zhang, W. 2021. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12): 11106–11115.

Zhou, T.; Ma, Z.; Wen, Q.; Sun, L.; Yao, T.; Yin, W.; Jin, R.; et al. 2022a. Film: Frequency improved legendre memory model for long-term time series forecasting. *Advances in neural information processing systems*, 35: 12677–12690.

Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; and Jin, R. 2022b. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, 27268–27286. PMLR.