HAR-former:Hybrid Transformer with an Adaptive Time-Frequency Representation Matrix for Long-Term Series Forecasting

Kenghao Zheng
Shenzhen Technology University

Zi Long*
Shenzhen Technology University

Shuxin Wang
Valueonline Inc.

Abstract

Time series forecasting is crucial across various fields such as economics, energy, transportation planning, and weather prediction. Nevertheless, accurately modeling real-world systems is challenging due to their inherent complexity and non-stationarity. Traditional methods, which often depend on high-dimensional embeddings, can obscure multivariate relationships and struggle with performance limitations, especially when handling complex temporal patterns. To address these issues, we propose **HAR-former**, a **H**ybrid Transformer with an **A**daptive Time-Frequency **R**epresentation Matrix, which combines the strengths of Multi-Layer Perceptrons (MLPs) and Transformers to process trend and seasonal components, respectively. The HAR-former leverages a novel adaptive time-frequency representation matrix to bridge the gap between the time and frequency domains, allowing the model to capture both long-range dependencies and localized patterns. Extensive experimental evaluation on eight real-world benchmark datasets demonstrates that HAR-former outperforms existing state-of-the-art (SOTA) methods, establishing it as a robust solution for complex time series forecasting tasks.

1 INTRODUCTION

Time series forecasting has a wide range of applications in economics, energy, transportation planning, and weather forecasting (Kaastra and Boyd, 1996). The inherent complexity and non-stationarity of real-world systems present significant challenges in time series analysis (Qin et al., 2017). Observed sequences often exhibit intricate temporal patterns that require sophisticated modeling techniques to capture their dynamics effectively. A single time point is frequently inadequate for conveying sufficient semantic information, making it essential to consider a broader context when analyzing such data (Liu et al., 2023; Li et al., 2023; Talukder et al., 2024). Traditional approaches often rely on embedding techniques—value embeddings, position embeddings, and time embeddings—that map or convolve information into high-dimensional feature spaces (Liu et al., 2023). This process can obscure multivariate correlations, leading to attention matrices that closely resemble the identity matrix after just the first epoch of training (Zhai et al., 2023; Dong et al., 2021), with minimal subsequent changes. This phenomenon is particularly pronounced in models employing softmax functions, which amplify differences in matrix values, thereby undermining the capacity of simple Transformer models to capture fundamental sequential characteristics and describe complex multivariate relationships (Zhai et al., 2023). Consequently, these limitations hinder the models' processing and generalization capabilities across diverse time series datasets.

To address the challenge of information masking due to embedding processes, it is crucial to enhance the representational power of models through the incorporation of time-frequency domain information. However, the alignment of temporal and frequency data poses significant difficulties, as the Time-Frequency non-alignment principle often results in weak correlations between these domains (Eldele et al., 2024; Wang et al., 2024b). The problem of domain adaptation (DA) further complicates matters, as models that are robust to domain shifts must learn highly generalizable features (He et al., 2023; Lu et al., 2024). Yet, neural networks trained under conventional practices tend to exploit spurious correlations arising from non-causal data artifacts (Geirhos et al., 2020; DeGrave et al., 2021), which impedes their transferability across different domains. This issue underscores the reality

Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258.

that latent representations may not generalize well to test datasets derived from subtly distinct underlying distributions.

The trend sequence is relatively stable, the multivariate relationship is simple, and the embedded label noise has little effect on the global representation of the aggregated sequence (Lim et al., 2021). While linear models can perform admirably on small, clean datasets, their efficacy diminishes when confronted with complex and noisy time series data. Research has shown that hybrid models, which combine different architectures, can effectively leverage the strengths of each component to improve predictive accuracy and robustness (Hajirahimi and Khashei, 2023). Specifically, while Transformer models excel at capturing long-range dependencies due to their self-attention mechanisms, Multi-Layer Perceptrons (MLPs) can effectively model local patterns and interactions in the data (Mahmoud and Mohammed, 2021; Toner and Darlow, 2024; Chen et al., 2023). This complementary nature suggests that a hybrid approach could provide a more comprehensive framework for time series analysis.

Our innovation, **HAR-former**, a **H**ybrid Transformer with an **A**daptive Time-Frequency **R**epresentation Matrix, strategically utilizes MLP and Transformer architectures to process trend and seasonal sequences, respectively. Within the Transformer framework, we introduce an adaptive time-frequency representation matrix designed to explore latent features that bridge the time and frequency domains. This enhancement enables the attention mechanism to focus on both time-domain and frequency-domain, significantly improving the model's representational capacity for intricate time series data. By leveraging the strengths of both MLPs and Transformers, HAR-former not only enhances the model's ability to capture diverse temporal patterns but also mitigates the impact of noise and irrelevant information. This hybrid approach positions HAR-former as a robust solution for analyzing complex time series, ultimately leading to improved predictive performance across various applications. Furthermore, it establishes HAR-former as the current state-of-the-art (SOTA) in performance. Our contributions lie in three key aspects:

- Given the non-stationarity of time series data and the differing performance of models on distinct components, the MLP model is employed for the trend component, while the Transformer model is utilized for the seasonal component. The final prediction is reconstructed from these two components.

- This paper introduces a custom Adaptive Representation Matrix (AR-Matrix) for mapping time-frequency learning. This matrix adaptively maps time-frequency domain information, effectively capturing the global characteristics of the time series.

- Extensive experimental work on eight real-world benchmark datasets demonstrates that our model outperforms existing state-of-the-art (SOTA) methods in predictive performance.

2 RELATED WORK

Recent advancements in time series forecasting primarily aim to simplify the modeling of temporal dependencies, yet they often overlook the essential cross-dimensional dependencies crucial for effective multivariate time series (MTS) forecasting.

Most existing approaches rely on pointwise attention mechanisms, which fail to capture the significance of patch-based representations. For example, LogTrans (Li et al., 2019) introduces LogSparse attention, eliminating pointwise dot products between keys and queries, yet its value representations still depend on individual time steps. Informer (Zhou et al., 2021) proposes ProbSparse self-attention utilizing KL divergence estimation, while Autoformer (Wu et al., 2021) employs autocorrelation for patch-level integration. FEDformer (Zhou et al., 2022) enhances the Transformer with a sparse frequency domain representation, but these designs may not effectively capture the vital semantic information inherent in patches. Similarly, Triformer (Cirstea et al., 2022) introduces patch attention to reduce complexity by using pseudo timestamps as queries; however, it does not treat patches as distinct input units nor adequately address their semantic importance.

In contrast, more recent models have aimed to capture both cross-time and cross-dimensional dependencies. Pyraformer (Liu et al., 2021) utilizes a pyramid attention mechanism, while Crossformer (Zhang and Yan, 2023) integrates a cross-scale embedding layer with Long Short Distance Attention (LSDA). Although these models attempt to address inter-dimensional dependencies, their computational complexity does not correspond to performance improvements and may introduce noise. In contrast, PatchTST (Nie et al., 2022), which employs a block strategy to extract local semantics while ensuring channel independence, demonstrates superior competitiveness. Additionally, TimesNet (Wu et al., 2022) applies Fourier transforms to decompose time series into components, and iTransformer (Liu et al., 2023) adapts Transformer inputs for improved time-series modeling.

However, these models may not fully leverage the rich contextual information available across different dimensions. Overall, while these models offer various innovations, they often fall short of addressing the complex relationships inherent in multivariate time series data, highlighting the necessity for a more integrated approach that effectively captures both temporal and cross-dimensional dependencies. Furthermore, mixing has emerged as an effective strategy for information integration, widely used in computer vision and natural language processing.

For instance, MLP-Mixer (Tolstikhin et al., 2021; Ekambaram et al., 2023; Wang et al., 2024a) introduces a two-stage mixing structure for image recognition, sequentially combining channel and patch information through linear layers. FNet (Lee-Thorp et al., 2021) replaces the attention layers in Transformers with a simple Fourier Transform, facilitating efficient token mixing within sentences. This paper further investigates the mixing structure within the context of time series forecasting. Rough Transformer (Moreno-Pino et al., 2025) introduces a novel extension of the Transformer architecture by incorporating rough path theory, inspired by the principles of Neural ODE models (Chen et al., 2018). Similarly, Structured State Space Models (Gu and Dao, 2023) leverage the application of Ordinary Differential Equations to address the challenges of modeling complex, high-dimensional time series data. The intricate nonlinear relationships in time series, which often exhibit chaotic dynamics, suggest that SSM is particularly adept at capturing long-term dependencies and modeling the dynamic evolution of systems over time. Orvieto and his research team propose a robust recursive framework for long-sequence modeling through an enhanced recurrent neural network, where the hidden state update mechanism closely parallels the state transition equations found in SSM. Additionally, linear recurrence can be conceptualized as a form of token mixing, a mechanism that enables the effective integration of sequence data. The Mamba model further advances the modeling of long sequences by introducing selective state spaces, which significantly improve both modeling efficiency and performance. Similarly, the S4 model optimizes the design of state spaces, thereby reducing the computational complexity associated with long-sequence modeling. These developments align with the objectives of HAR-former, which integrates time-domain and frequency-domain information to capture intricate dynamic patterns while maintaining computational efficiency.

Although the core frameworks of SSM and HAR-former differ—SSM primarily relies on hidden states to model the dynamic evolution of sequences, while HAR-former enhances its representational power by incorporating both time-domain and frequency-domain features—both methods share a common goal: optimizing the modeling of time series data. Specifically, both approaches excel in capturing long-term dependencies and cross-dimensional interactions, offering complementary strengths for time series forecasting tasks.This paper further investigates the mixing structure within the context of time series forecasting.

3 METHODOLOGY

In multivariate time series forecasting, given historical observations $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\} \in \mathbb{R}^{T \times N}$, where T represents time steps and N denotes variates, the goal is to predict the next S time steps $\boldsymbol{Y} = \{\boldsymbol{x}_{T+1}, \ldots, \boldsymbol{x}_{T+S}\} \in \mathbb{R}^{S \times N}$. Figure 1 presents the architecture of the proposed HAR-former model. The HAR-former begins by applying a sequence decomposition method, which divides the input time series into trend and seasonal components. The trend component is processed through a Multi-Layer Perceptron (MLP), while the seasonal component is addressed using Transformer-based attention mechanisms, specifically the AR-Matrix, to capture multiscale temporal dependencies. The final prediction is obtained by merging the trend and seasonal predictions.

3.1 Decomposition And Reconstruction

To effectively capture the intricate temporal patterns in long-term forecasting, this model utilizes a sequence decomposition method that separates the time series into trend and seasonal components (Zhou et al., 2022). Each component is forecasted independently, and the results are aggregated at the end.

As illustrated in Figure 1, the model utilizes a sequence decomposition block to incrementally extract long-term trends from intermediate hidden variables during forecasting. By applying averaging filters of varying dimensions, distinct trend patterns are captured and combined into a unified trend component with adaptive weights. The seasonal component is then obtained by subtracting the trend from the original time series.

The trend component, denoted as X_{trend}, is calculated using the softmax operation σ, data-dependent weights $w(x)$, and averaging filters $f(x)$ (Babu and Reddy, 2014):

$$X_{\text{trend}} = \sigma(w(x)) \cdot f(x)$$

The seasonal component is then determined by:

$$X_{\text{seasonal}} = X - X_{\text{trend}}$$

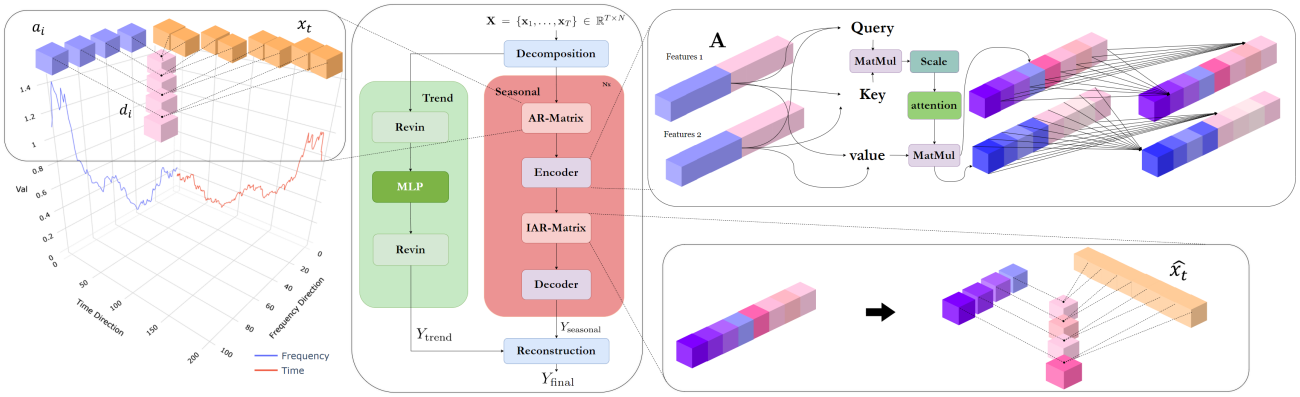

Figure 1: Illustration of HAR-former Architecture: Visual representation of the decomposition, feature extraction, and reconstruction process in the AR-Matrix. The purple and pink blocks represent the concatenated approximation and detail coefficients.

This methodology ensures that both components are accurately modeled, leveraging the strengths of MLP and Transformer architectures to enhance prediction accuracy and efficiency. A three-layer MLP is employed for forecasting the trend component. Reversible Instance Normalization (RevIN) (Kim et al., 2021) is integrated to effectively reduce non-stationary information, a key characteristic of trends:

$$Y_{\text{trend}} = \text{RevIN}(\text{MLP}(\text{RevIN}(X_{\text{trend}})))$$

Once the predictions for the trend and seasonal components are obtained, they are merged to produce the final prediction. This merging process sums the predicted components to reconstruct the original time series, allowing the model to integrate long-term trend changes with short-term seasonal fluctuations:

$$Y_{\text{final}} = Y_{\text{trend}} + Y_{\text{seasonal}}$$

This strategy effectively considers both linear trends and periodic variations, improving the accuracy of predictions and the model's adaptability to different time series features. By combining these components, the model is equipped to tackle complex forecasting tasks, leveraging the strengths of both MLP and Transformer architectures, ultimately enhancing overall predictive performance. This approach of decomposition and merging provides a robust framework for time series analysis.

3.2 Adaptive Representation Matrix

The AR-Matrix employs the Haar wavelet transform to seamlessly integrate time-domain and frequency-domain information. As illustrated in Figure 1, the time series data $x_t \in \mathbb{R}^N$ undergoes wavelet decomposition, resulting in approximation coefficients a_i and detail coefficients d_i across various scales. In the left section of the diagram, this decomposition is represented by the blue and orange 3D cubes: the approximation coefficients a_i capture the low-frequency components of the signal, while the detail coefficients d_i represent the high-frequency variations. The frequency-domain component, represented by a_i, is plotted along the frequency axis, while the time-domain component, represented by d_i, is aligned along the time axis.

Upon applying the Haar wavelet transform, the original sequence is partitioned into segments of equal length, 2^k. Each segment produces an approximation coefficient a_i and a corresponding detail coefficient d_i, visually represented by the pink intermediate cubes between the frequency (blue) and time (orange) segments in the diagram. The decomposition spans the range $1 \leq i \leq \frac{N}{2^k}$, and at $k = 1$, the sequence is halved, significantly reducing memory overhead, which is critical for processing large datasets.

The relationship between the original sequence $g(x)$, its approximation coefficients, and detail coefficients is given by the following mathematical expression:

$$g(x) = \sum_{k \in \mathbb{Z}} a_k \phi(x - k) + \sum_{l=0}^{L} \sum_{k \in \mathbb{Z}} d_k^l \psi(2^l x - k)$$

In this equation, a_k represents the approximation coefficients at different scales, while d_k^l corresponds to the detail coefficients at various levels l. The functions $\phi(x)$ and $\psi(x)$ are the scaling and wavelet functions, respectively, capturing the low- and high-frequency information of the signal. The summation over l accumulates the detail coefficients across different scales, providing a complete reconstruction of the original sequence $g(x)$ by combining both approximation and de-

tail components. Here, $\mathbb{Z}$ represents the set of integers, and L denotes the maximum level of decomposition.

Once the approximation and detail coefficients are obtained, they are concatenated into a matrix, visually represented as the combined purple cubes on the right side of the diagram. This matrix is denoted as:

$$\mathbf{A} = [a_i, d_i]_{i=1}^{\frac{N}{2^k}}$$

To establish a solid theoretical basis for processing these coefficients, we introduce the DeepSets (Zaheer et al., 2017) framework. Traditionally, DeepSets use a permutation-invariant function f to aggregate features via summation, often implemented with neural networks. In our approach, we replace this neural network component with an attention mechanism to dynamically capture dependencies among the coefficient pairs.

The right portion of the diagram shows how these coefficients are processed through an attention mechanism, where the representations are input into the query, key, and value matrices $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ for multi-head attention. Specifically, we compute: $\mathbf{Q} = \mathbf{W}_q\mathbf{A}$, $\mathbf{K} = \mathbf{W}_k\mathbf{A}$, $\mathbf{V} = \mathbf{W}_v\mathbf{A}$,where $\mathbf{W}_q$, $\mathbf{W}_k$, and $\mathbf{W}_v$ are learned weight matrices. The attention weights are then calculated as:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{p_k}}\right)\mathbf{V},$$

With p_k representing the dimension of the key vectors. This approach replaces the neural network in DeepSets, allowing the model to weigh the importance of each coefficient pair (a_i, d_i) based on their interrelationships.

Using attention instead of a neural network enables the model to effectively integrate time-domain trends with high-frequency fluctuations. The approximation coefficients a_i capture smooth, long-term patterns, while the detail coefficients d_i reflect rapid changes and high-frequency variations. This duality enriches the input, allowing the model to extract latent relationships between the frequency-domain and time-domain components.

The transition from smaller to larger purple and pink blocks in the diagram illustrates how the attention mechanism constructs a richer representation by capturing complex interactions across these domains, enhancing the model's ability to learn intricate time series dynamics.

In the final stage, the collaborative integration of approximation and detail coefficients, facilitated by the attention mechanism, ensures that the fusion of temporal features is both sophisticated and mathematically sound, leading to improved performance in modeling complex time series data.

Finally, the reconstructed sequence $\hat{x}_t$ is derived through the inverse Haar wavelet transform (IWHT), as demonstrated in the lower-right section of the diagram. The inverse transformation seamlessly reassembles the original sequence $\hat{x}_t$ by combining both approximation and detail coefficients:

$$\hat{x}_t = \text{IWHT}(\mathbf{A}) = \sum_{i=1}^{\frac{N}{2^k}} a_i\phi(x-i) + \sum_{l=0}^{L}\sum_{i=1}^{\frac{N}{2^k}} d_i\psi(2^l x - i)$$

This approach, visualized through the merging of the purple and pink blocks into $\hat{x}_t$, equips the AR-Matrix with robust representational capabilities, enabling it to effectively handle intricate time series forecasting tasks by merging time-domain and frequency-domain information. Consequently, the seasonal component of the model can be expressed as: $Y_{\text{seasonal}} = \hat{x}_t$, providing a clear delineation of the seasonal effects captured by the model. As shown in the red section of the diagram, the encoder and decoder play key roles in processing time-domain and frequency-domain information.

The encoder extracts multi-level features from the concatenated matrix $\mathbf{A}$, leveraging multi-head attention to capture complex dependencies across both time steps and frequency components. This process is illustrated by the connections between the pink and purple blocks and the attention mechanism. The decoder, located in the lower red section, takes the high-dimensional embeddings from the encoder and employs reverse attention to reconstruct the original data, effectively integrating time and frequency information. Working together, the encoder and decoder empower the model to learn both long-term trends and fine-grained variations, significantly enhancing its forecasting accuracy and overall performance.

4 EXPERIMENTS

This model provides a comprehensive evaluation of the proposed HAR-former in various time series forecasting applications, validates the generality of the proposed framework, and offers further insights into the effectiveness of applying the Transformer component to different dimensions of time series.

Dataset The model was tested on real-world datasets, as summarized in Table 1, following the methodology of iTransformer (Liu et al., 2023). The datasets include: (1) ETTh (Electricity Transformer Temperature-hourly), which contains data collected from power Transformers every 15 minutes from July 2016 to July 2018, including load and oil temperatures;

Table 1: Summary of Dataset

Dataset	Variate	Frequency	Timesteps
ETTm1	7	5 min	69,680
ETTh1	7	Hourly	17,420
Weather	21	10 min	52,696
Solar-Energy	137	10 min	52,560
Electricity	321	Hourly	26,304
Exchange	8	Daily	7,588
Traffic	862	Hourly	17,544

(2) the Weather dataset, recorded every 10 minutes throughout 2020, containing 21 meteorological indicators such as temperature and humidity; (3) the ECL (Electricity Consuming Load) dataset, which contains the hourly electricity consumption of 321 customers from 2012 to 2014; (4) the Solar-Energy dataset; and (5) the Exchange dataset, which records the daily exchange rates of 8 different countries from 1990 to 2016; and (6) Traffic is a collection of hourly data from California Department of Transportation, which describes the road occupancy rates measured by different sensors on San Francisco Bay area freeways. The model follows the standard protocol of dividing all datasets into training, validation, and test sets in chronological order. For the ETT dataset, the split ratio is 6:2:2, while for the other datasets, the split ratio is 7:1:2.

Baseline Five state-of-the-art long-term sequence prediction models were selected as benchmarks, Transformer-based methods: iTransformer (Liu et al., 2023), Autoformer (Wu et al., 2021), FEDformer (Zhou et al., 2022), PatchTST (Nie et al., 2022); and TCN-based methods: TimesNet (Wu et al., 2022).

Setup All experiments were implemented using PyTorch and performed on a single NVIDIA A800 80GB GPU. The Adam optimizer was used with an initial learning rate of 0.001, and the model was optimized for L1Loss with a batch size of 16. The training process is early stopped within 10 epochs. HAR-former contains 2 encoder layers and 1 decoder layer. The data reproduced in this paper for comparing the baseline models were partly taken from the original papers and partly from the data provided by iTransformer. All experimental results are averaged over five runs with different random seeds.

4.1 Main Results

Table 2 presents the combined prediction results for prediction lengths $S \in \{96, 192, 336, 720\}$ with a fixed lookback length of $T = 96$. The results are averaged across all prediction lengths, with "Avg" denoting the average of these subsets. The best results are highlighted in **black**, while the second-best results are underlined. A lower Mean Squared Error (MSE) or Mean Absolute Error (MAE) indicates a smaller gap between the predicted results and the true values, reflecting improved prediction performance. Specifically, the lower the MSE/MAE (Chicco et al., 2021), the smaller the difference between the predicted and actual values, leading to greater accuracy.

Compared to other predictors, HAR-former excels at predicting high-dimensional time series. PatchTST, previously considered state-of-the-art, often struggles in many cases, potentially due to significant fluctuations in the dataset that cause its patching mechanism to lose focus on specific local areas and struggle with rapid changes. In contrast, HAR-former aggregates information from both the time and frequency domains into a sequence representation, making it better equipped to handle such fluctuations.

It is also notable that iTransformer, which explicitly captures multivariate correlations, does not perform as well as HAR-former. The interaction of time-calibrated patches from different multivariate variables is likely to introduce unwanted noise into the forecasts. By capitalizing on the intrinsic properties of time series, the native Transformer component is able to effectively model temporal data and multivariate correlations. The proposed AR-Matrix further enhances this capability, effectively addressing real-world time series forecasting challenges.

4.2 Model Analysis

Increasing Lookback Length Previous studies have demonstrated that the predictive performance of a system does not necessarily improve with an increase in the length of the Transformer lookback (Nie et al., 2022). This may be attributed to the distraction caused by the growing input size. Nevertheless, linear models typically exhibit the expected performance enhancements, which are theoretically substantiated by statistical methodologies and allow for the incorporation of a larger quantity of historical data.

The performance of various Transformer models and HAR-former is evaluated in Figure 2 following an increase in the lookback length. The results validate the effectiveness of combining the AR-Matrix with frequency domain information, demonstrating that HAR-former can benefit from an extended lookback window. Compared to other Transformer models, the proposed approach in this paper has a stronger characterization capability and achieves better performance even when dealing with more complex time series, leading to more

Table 2: Lists the combined prediction results with prediction length $S \in \{96, 192, 336, 720\}$ and fixed lookback length T = 96.

Models		**Ours**		iTransformer		PatchTST		TimesNet		FEDformer		Autoformer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Exchange	96	**0.081**	**0.199**	0.086	0.206	0.088	0.205	0.107	0.234	0.148	0.278	0.197	0.323
	192	**0.169**	**0.292**	0.177	0.299	0.176	0.299	0.226	0.344	0.271	0.380	0.300	0.369
	336	0.317	0.406	0.331	0.417	**0.301**	**0.397**	0.367	0.448	0.460	0.500	0.509	0.524
	720	**0.819**	**0.680**	0.847	0.691	0.901	0.714	0.964	0.746	1.195	0.841	1.447	0.941
ECL	96	**0.144**	**0.237**	0.148	0.240	0.195	0.285	0.169	0.273	0.193	0.308	0.201	0.317
	192	**0.162**	**0.246**	**0.162**	0.253	0.199	0.289	0.182	0.286	0.201	0.315	0.222	0.334
	336	**0.178**	**0.263**	**0.178**	0.269	0.215	0.305	0.200	0.304	0.214	0.329	0.231	0.338
	720	**0.201**	**0.293**	0.225	0.317	0.256	0.337	0.222	0.321	0.246	0.355	0.254	0.361
Weather	96	**0.162**	**0.196**	0.174	0.214	0.177	0.218	0.172	0.220	0.217	0.296	0.266	0.336
	192	**0.208**	**0.240**	0.221	0.254	0.225	0.259	0.219	0.261	0.276	0.336	0.307	0.367
	336	**0.263**	**0.281**	0.278	0.296	0.278	0.297	0.280	0.306	0.339	0.380	0.359	0.395
	720	**0.339**	**0.332**	0.358	0.349	0.354	0.348	0.365	0.359	0.403	0.428	0.419	0.428
ETTh1	96	**0.329**	**0.373**	0.386	0.405	0.414	0.419	0.384	0.402	0.376	0.419	0.449	0.459
	192	**0.358**	**0.392**	0.441	0.436	0.460	0.445	0.436	0.429	0.420	0.448	0.500	0.482
	336	**0.428**	**0.441**	0.487	0.458	0.501	0.466	0.491	0.469	0.459	0.465	0.521	0.496
	720	**0.462**	**0.457**	0.503	0.491	0.500	0.488	0.521	0.500	0.506	0.507	0.514	0.512
ETTh2	96	**0.190**	**0.275**	0.297	0.349	0.302	0.347	0.340	0.374	0.358	0.397	0.346	0.388
	192	**0.231**	**0.306**	0.380	0.400	0.387	0.401	0.402	0.414	0.429	0.439	0.456	0.452
	336	**0.274**	**0.347**	0.428	0.432	0.426	0.433	0.452	0.452	0.496	0.487	0.482	0.486
	720	**0.418**	**0.433**	0.427	0.445	0.431	0.446	0.462	0.468	0.463	0.474	0.515	0.511
ETTm1	96	**0.321**	**0.343**	0.334	0.368	0.329	0.367	0.338	0.375	0.379	0.419	0.505	0.475
	192	0.372	**0.370**	0.377	0.391	**0.367**	0.385	0.374	0.387	0.426	0.441	0.553	0.496
	336	**0.399**	**0.391**	0.426	0.420	**0.399**	0.409	0.410	0.411	0.445	0.459	0.621	0.537
	720	0.462	**0.429**	0.491	0.459	**0.458**	0.441	0.478	0.450	0.543	0.490	0.671	0.561
ETTm2	96	**0.174**	**0.251**	0.180	0.264	0.175	0.259	0.187	0.267	0.203	0.287	0.255	0.475
	192	**0.241**	**0.295**	0.250	0.309	**0.241**	0.302	0.249	0.309	0.269	0.328	0.281	0.340
	336	**0.299**	**0.333**	0.311	0.348	0.305	0.343	0.321	0.351	0.325	0.366	0.339	0.372
	720	**0.395**	**0.390**	0.412	0.407	0.402	0.400	0.408	0.403	0.421	0.415	0.433	0.432
Solar	96	**0.199**	0.239	0.203	**0.237**	0.234	0.286	0.250	0.292	0.242	0.342	0.884	0.711
	192	**0.232**	**0.261**	0.233	**0.261**	0.267	0.310	0.296	0.318	0.285	0.380	0.834	0.692
	336	0.250	0.275	**0.248**	**0.273**	0.290	0.315	0.319	0.330	0.282	0.376	0.941	0.723
	720	0.254	0.278	**0.249**	**0.275**	0.289	0.317	0.338	0.337	0.357	0.427	0.882	0.717
Traffic	96	0.453	**0.267**	**0.395**	0.268	0.544	0.359	0.593	0.321	0.587	0.366	0.613	0.388
	192	0.464	**0.273**	**0.417**	0.276	0.540	0.354	0.617	0.336	0.604	0.373	0.616	0.382
	336	0.480	**0.279**	**0.433**	0.283	0.551	0.358	0.629	0.336	0.621	0.383	0.622	0.337
	720	0.531	**0.298**	**0.467**	0.302	0.589	0.328	0.640	0.350	0.626	0.382	0.660	0.408

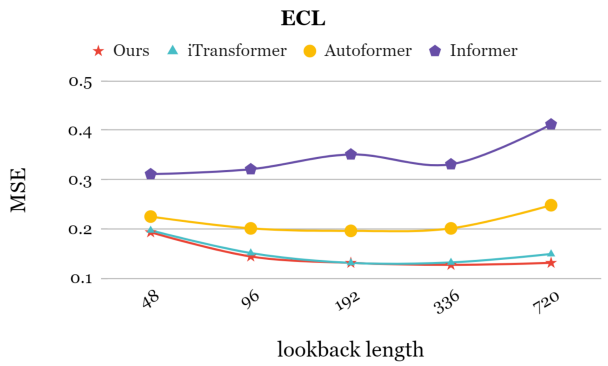

Figure 2: Forecasting performance with the lookback length $T \in 48, 96, 192, 336, 720$ and fixed prediction length $S = 96$.

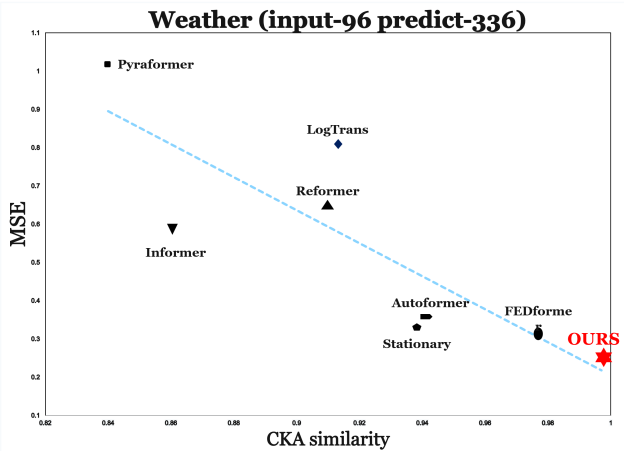

Figure 3: Analysis of series representations and multivariate correlations, MSE and CKA similarity of representations comparison between Transformers and HAR-former.

accurate predictions.

Analysis of Series Representations By assigning the responsibility of modeling multivariate correlations to the attention mechanism, the learned representations become more interpretable. Figure 3 illustrates a visual example of a weather sequence, which shows discernible correlations in both the past and future windows. There is a high degree of similarity between the correlations in the learned embeddings and those in the original input sequences at the attention level. This verifies that the AR-Matrix provides interpretable and relevant attention, and that the process of encoding the past and decoding the future in the feed-forward network is effectively carried out within the series representation.

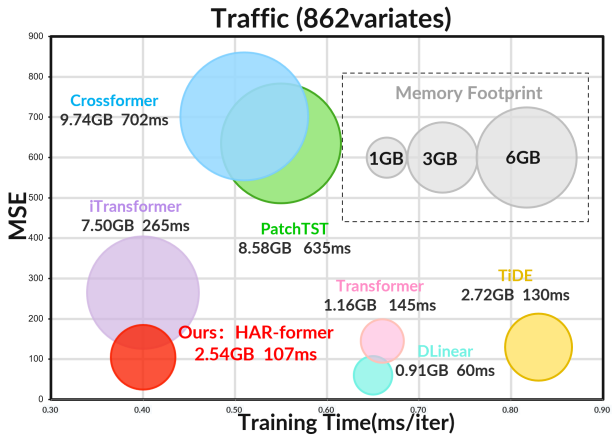

Figure 4: Model efficiency comparison under input-96-predict-96 of Traffic.

Performance Improvement We evaluated HAR-former and other models after integrating the AR-Matrix, as shown in Table 3. The results indicate consistent performance improvements across various models. For instance, on the Exchange dataset, the AR-Matrix improved MSE by **11.31%** and MAE by **13.41%**, while on the ETT-h2 dataset, MSE improved by **18.49%** and MAE by **14.03%**. Additionally, integrating the AR-Matrix into Reformer, Informer, and Autoformer led to notable MSE reductions, with improvements of up to **54.74%** on Reformer and **64.64%** on Informer. These enhancements demonstrate that the AR-Matrix effectively boosts model accuracy and reduces errors across different architectures.

Efficient Training Strategy Figure 4 shows the relative efficiency of different models when processing a traffic dataset comprising 862 variables and 96 backtracking time steps. To ensure the validity of the evaluation, the parameter configuration of each model was kept constant (Liu et al., 2023). HAR-former exhibits a notable advantage in memory consumption compared to other Transformers when handling this large dataset. Additionally, HAR-former demonstrates superior training speed. As a hybrid model, HAR-former readily outperforms traditional Transformers in terms of efficiency. Moreover, its efficiency is noteworthy even when compared to lightweight MLP models such as DLinear (Zeng et al., 2023) and TiDE (Das et al., 2023), due to its exemplary performance and effective utilization of multivariate correlations, which enhance both performance and efficiency.

Analysis of Series Representations By ascribing the responsibility for modeling multivariate correlations to the attention mechanism, the acquired representations become more interpretable (Kornblith

Table 3: Comparison of models on Exchange, Weather, and ETT-h2 datasets with MAE and MSE

Models	Metric	HAR-former*(Our)		Reformer		Informer		Autoformer	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Exchange	Original	0.389	0.455	1.280	0.932	1.550	0.998	0.613	0.539
	+AR-Matrix	**0.345**	**0.394**	**1.102**	**0.899**	**0.994**	**0.796**	**0.434**	**0.464**
	Promotion	11.31%	13.41%	13.91%	3.54%	35.87%	20.24%	29.2%	13.91%
Weather	Original	0.261	0.279	0.803	0.656	0.634	0.548	0.338	0.382
	+AR-Matrix	**0.243**	**0.263**	**0.523**	**0.507**	**0.373**	**0.425**	**0.298**	**0.331**
	Promotion	6.9%	5.73%	34.87%	22.71%	41.17%	22.45%	11.83%	13.35%
ETTh2	Original	0.438	0.449	6.462	1.910	4.431	1.729	0.450	0.459
	+AR-Matrix	**0.357**	**0.386**	**2.925**	**1.307**	**1.567**	**1.071**	**0.359**	**0.406**
	Promotion	18.49%	14.03%	54.74%	31.57%	64.64%	38.06%	20.22%	11.55%

Table 4: Comparison of HAR-former designs on various datasets .

Design	Seasonal	Trend	ECL		Exchange		Weather		ETTh2	
			MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
HAR-former	Attention	MLP	**0.173**	**0.260**	**0.346**	**0.394**	**0.242**	**0.263**	**0.278**	**0.340**
Replace	Attention	Attention	0.387	0.448	0.434	0.467	0.302	0.335	0.361	0.405
	MLP	Attention	0.342	0.399	2.863	1.499	1.264	0.795	1.755	0.871
	MLP	MLP	0.352	0.398	0.599	0.348	0.245	0.273	0.366	0.393
w/o	Attention	Attention	0.897	0.689	2.818	1.172	1.245	0.778	1.331	0.772
	MLP	MLP	0.193	0.281	0.601	0.402	0.265	0.278	0.365	0.483

et al., 2019). The accompanying figure provides a visual illustration of a weather sequence exhibiting clear correlations in both the past and future windows, as shown in Figure 3. It can be observed that there are numerous similarities between the correlations of the learned representations and those of the original input sequences at the shallow attention level. This demonstrates that the AR-Matrix enables interpretable and relevant attention, and that the process of encoding the past and decoding the future in the feed-forward network is effectively carried out within the series representation.

Ablation Study To verify the effectiveness of the Transformer components, we conducted ablation experiments on the HAR-former model, specifically removing the decomposition blocks for seasonality and trend (w/o decomposition). The results in Table 4 show that while the model performs well without decomposition, including it significantly boosts performance. The AR-Matrix remains highly effective at learning multivariate correlations, but the decomposition further enhances the overall results.

5 CONCLUSION

In light of the distinctive characteristics of multivariate time series, we propose the HAR-former, which employs a model architecture that aligns with the nature of the time series in question. This architecture facilitates the integration of time-domain and frequency-domain information, thereby directing the attention mechanism to a more optimal learning of global dependencies and the relationships among multiple variables. In experimental trials, the HAR-former demonstrated state-of-the-art performance, exhibiting notable versatility in its framework and providing robust analytical support.

[1]

References

Babu, C. N. and Reddy, B. E. (2014). A moving-average filter based hybrid arima–ann model for forecasting time series data. *Applied Soft Computing*, 23:27–38.

Chen, R. T., Rubanova, Y., Bettencourt, J., and Du-

[1]The HARformer model code will be open-sourced at the following link. `https://github.com/kktoucheme/HARformer`.

venaud, D. K. (2018). Neural ordinary differential equations. *Advances in neural information processing systems*, 31.

Chen, S.-A., Li, C.-L., Yoder, N., Arik, S. O., and Pfister, T. (2023). Tsmixer: An all-mlp architecture for time series forecasting. *arXiv preprint arXiv:2303.06053*.

Chicco, D., Warrens, M. J., and Jurman, G. (2021). The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation. *Peerj computer science*, 7:e623.

Cirstea, R.-G., Guo, C., Yang, B., Kieu, T., Dong, X., and Pan, S. (2022). Triformer: Triangular, variable-specific attentions for long sequence multivariate time series forecasting–full version.

Das, A., Kong, W., Leach, A., Mathur, S., Sen, R., and Yu, R. (2023). Long-term forecasting with tide: Time-series dense encoder.

DeGrave, A. J., Janizek, J. D., and Lee, S.-I. (2021). Ai for radiographic covid-19 detection selects shortcuts over signal. *Nature Machine Intelligence*, 3(7):610–619.

Dong, Y., Cordonnier, J.-B., and Loukas, A. (2021). Attention is not all you need: Pure attention loses rank doubly exponentially with depth. In *International Conference on Machine Learning*, pages 2793–2803. PMLR.

Ekambaram, V., Jati, A., Nguyen, N., Sinthong, P., and 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*, pages 459–469.

Eldele, E., Ragab, M., Chen, Z., Wu, M., and Li, X. (2024). Tslanet: Rethinking transformers for time series representation learning. *arXiv preprint arXiv:2404.08472*.

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., and Wichmann, F. A. (2020). Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.

Gu, A. and Dao, T. (2023). Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

Hajirahimi, Z. and Khashei, M. (2023). Hybridization of hybrid structures for time series forecasting: A review. *Artificial Intelligence Review*, 56(2):1201–1261.

He, H., Queen, O., Koker, T., Cuevas, C., Tsiligkaridis, T., and Zitnik, M. (2023). Domain adaptation for time series under feature and label shifts. In *International Conference on Machine Learning*, pages 12746–12774. PMLR.

Kaastra, I. and Boyd, M. (1996). Designing a neural network for forecasting financial and economic time series. *Neurocomputing*, 10(3):215–236.

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*.

Kornblith, S., Norouzi, M., Lee, H., and Hinton, G. (2019). Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMLR.

Lee-Thorp, J., Ainslie, J., Eckstein, I., and Ontanon, S. (2021). Fnet: Mixing tokens with fourier transforms. *arXiv preprint arXiv:2105.03824*.

Li, S., Jin, X., Xuan, Y., Zhou, X., Chen, W., Wang, Y.-X., and Yan, X. (2019). Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. volume 32.

Li, Z., Qi, S., Li, Y., and Xu, Z. (2023). Revisiting long-term time series forecasting: An investigation on linear mapping. *arXiv preprint arXiv:2305.10721*.

Lim, B., Arık, S. Ö., Loeff, N., and Pfister, T. (2021). Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4):1748–1764.

Liu, S., Yu, H., Liao, C., Li, J., Lin, W., Liu, A. X., and Dustdar, S. (2021). Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International conference on learning representations*.

Liu, Y., Hu, T., Zhang, H., Wu, H., Wang, S., Ma, L., and Long, M. (2023). itransformer: Inverted transformers are effective for time series forecasting.

Lu, J., Han, X., Sun, Y., and Yang, S. (2024). Cats: Enhancing multivariate time series forecasting by constructing auxiliary time series as exogenous variables. *arXiv preprint arXiv:2403.01673*.

Mahmoud, A. and Mohammed, A. (2021). A survey on deep learning for time-series forecasting. *Machine learning and big data analytics paradigms: analysis, applications and challenges*, pages 365–392.

Moreno-Pino, F., Arroyo, Á., Waldon, H., Dong, X., and Cartea, Á. (2025). Rough transformers: Lightweight and continuous time series modelling through signature patching. *Advances in Neural Information Processing Systems*, 37:106264–106294.

Nie, Y., Nguyen, N. H., Sinthong, P., and Kalagnanam, J. (2022). A time series is worth 64 words: Long-term forecasting with transformers.

Qin, Y., Song, D., Chen, H., Cheng, W., Jiang, G., and Cottrell, G. (2017). A dual-stage attention-based recurrent neural network for time series prediction.

Talukder, S., Yue, Y., and Gkioxari, G. (2024). Totem: Tokenized time series embeddings for general time series analysis. *arXiv preprint arXiv:2402.16412*.

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.

Toner, W. and Darlow, L. (2024). An analysis of linear time series forecasting models. *arXiv preprint arXiv:2403.14587*.

Wang, S., Wu, H., Shi, X., Hu, T., Luo, H., Ma, L., Zhang, J. Y., and Zhou, J. (2024a). Timemixer: Decomposable multiscale mixing for time series forecasting. *arXiv preprint arXiv:2405.14616*.

Wang, X., Wang, S., Tang, C., Zhu, L., Jiang, B., Tian, Y., and Tang, J. (2024b). Event stream-based visual object tracking: A high-resolution benchmark dataset and a novel baseline. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19248–19257.

Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., and Long, M. (2022). Timesnet: Temporal 2d-variation modeling for general time series analysis. arXivpreprint.

Wu, H., Xu, J., Wang, J., and Long, M. (2021). Autoformer: Decomposition transformers with autocorrelation for long-term series forecasting. volume 34, pages 22419–22430.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., and Smola, A. J. (2017). Deep sets. *Advances in neural information processing systems*, 30.

Zeng, A., Chen, M., Zhang, L., and Xu, Q. (2023). Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 11121–11128.

Zhai, S., Likhomanenko, T., Littwin, E., Busbridge, D., Ramapuram, J., Zhang, Y., Gu, J., and Susskind, J. M. (2023). Stabilizing transformer training by preventing attention entropy collapse. In *International Conference on Machine Learning*, pages 40770–40803. PMLR.

Zhang, Y. and Yan, J. (2023). Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *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. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11106–11115.

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

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [**Yes**/No/Not Applicable]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [**Yes**/No/Not Applicable]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [**Yes**/No/Not Applicable]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [**Yes**/No/Not Applicable]
   (b) Complete proofs of all theoretical results. [Yes/No/**Not Applicable**]
   (c) Clear explanations of any assumptions. [**Yes**/No/Not Applicable]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [**Yes**/No/Not Applicable]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [**Yes**/No/Not Applicable]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [**Yes**/No/Not Applicable]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [**Yes**/No/Not Applicable]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [**Yes**/No/Not Applicable]
   (b) The license information of the assets, if applicable. [**Yes**/No/Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [**Yes**/No/Not Applicable]
   (d) Information about consent from data providers/curators. [Yes/No/**Not Applicable**]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Yes/No/**Not Applicable**]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Yes/No/**Not Applicable**]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Yes/No/**Not Applicable**]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Yes/No/**Not Applicable**]