# Memory-Augmented Short Time Series Forecasting

Si Chen[1], Xinhuan Chen[2(✉)], and Youhuan Li[1]

[1] College of Cyber Science and Technology, Hunan University, Hunan, China
{sichen,liyouhuan}@hnu.edu.cn
[2] Tencent Inc.
chenxinhuancxh@gmail.com

**Abstract.** Time series forecasting is widely applied in many applications such as finance, healthcare, and transportation. Despite its broad use, real-world scenarios often present the challenge of cold start problems. For example, newly built wind farms face challenges in forecasting wind speed due to insufficient historical observation records. Cold start problems make it challenging for the current time series forecasting solutions to perform effectively. Therefore, addressing the issue and achieving high-quality time series predictions with limited data (i.e., short time series data) is crucial. In response to these challenges, we propose a novel **Mem**ory-Augmented **S**hort **T**ime **S**eries **F**orecasting model (**MEMSTSF**). Specifically, we design a novel Transformer architecture with a memory module to not only capture temporal dependencies within the sequence, but also leverage auxiliary information from multiple data sources as memory to assist short time series in learning intricate temporal patterns. Furthermore, the novel Transformer architecture features a built-in time series decomposition module, which captures the global properties of the target time series, enhancing the comprehensiveness and overall predictive performance of MEMSTSF. Extensive experiments on three datasets demonstrate the superiority of our model in forecasting short time series with insufficient historical data.

**Keywords:** Time Series Forecasting · Memory Network · Deep Neural Network.

## 1 Introduction

Time series forecasting has extensive applications in various domains, such as economics, traffic, and weather. However, in many real-world scenarios, the lack of sufficient historical data and stable patterns often leads to the cold start problem and short time series, making it challenging for current time series forecasting models to deliver accurate predictions. Nevertheless, achieving precise forecasts under cold start conditions remains crucial and necessary. For example, in the energy domain, predicting wind speeds for newly established wind farms is essential to maximizing energy production efficiency. In financial services where a lender must provide funds to users and predict their future borrowing behavior,

the limited and unstable data from new user groups often lead to poor predictive performance, negatively impacting user experience and potentially resulting in customer churn. Although waiting for sufficient data records to accumulate can enhance the accuracy of funding predictions, it also increases the fund preparation costs, thereby reducing profitability. Therefore, improving prediction accuracy in situations with insufficient and unstable data is particularly critical, as it can significantly optimize fund utilization and improve overall profitability.

Inspired by the powerful capability of the Transformer network [20] in capturing sequence interactions, numerous Transformer-based models have been proposed for time series modeling, with most of them focusing on long-term time series forecasting and have achieved great success [27,23,29,11]. These methods require sufficient historical data, which is usually hard to obtain. The short time series forecasting problem, which suffers from insufficient historical data and unstable data patterns, remains a significant challenge in time series forecasting. However, there has been little research on addressing this problem. For example, [8] designs a shared-hidden-layer DNN architecture for forecasting newly built wind farm speeds. D3VAE [10] proposes a bidirectional variational auto-encoder for limited and noisy time series forecasting. BHT-ARIMA [16] exploits the low-rank structure of block Hankel tensors and captures the correlations among multiple time series. Despite the progress achieved by these works, they have not focused on some important attributes of time series, such as seasonality and periodicity, resulting in their inability to reason complex temporal patterns. Traditional models [1,18] usually take the idea of time series decomposition to capture the temporal attributes of time series. Neural network methods typically prioritize improving prediction accuracy while overlooking these attributes. Recently, there have been some efforts to integrate these two aspects. NeuralProphet [19] is a hybrid framework that combines deep learning and time series decomposition. LaST [21] presents a disentangled variational inference framework to disassociate seasonal-trend representations in latent space. However, these models were not explicitly designed for short time series forecasting problems and cannot leverage limited data to generate high-quality predictions.

We observe that although predicting short time series is a significant challenge, other data-rich long time series within the same scenario may provide valuable insights, such as the behavior patterns and trends of mature users in the business domain. These data can offer important references for new users, helping to better understand their potential needs and behaviors. To address the aforementioned challenges and leverage valuable information of long time series from the same scenario as the target series, we propose a novel **Mem**ory-Augmented **S**hort **T**ime **S**eries **F**orecasting model (**MEMSTSF**). Specifically, we design a novel memory-augmented Transformer architecture that integrates memory networks and time series decomposition functions. This architecture can not only efficiently capture temporal dependencies within the time series but also provide a comprehensive understanding of complex temporal patterns, such as overall trends and seasonal variations. MEMSTSF can store and retrieve auxiliary long time series data from multiple sources in real time. By intelli-

gently matching short time series with the information in the memory module, it provides more accurate and targeted data references for short time series, improving the model's predictive performance when handling limited data. The contributions of our work are summarized as follows:

- We focus on addressing the challenge of insufficient historical data in short time series forecasting under cold start situations. This innovation enriches the solutions for short time series forecasting and has broad application potential in practical scenarios.
- We propose a novel memory-augmented Transformer model for short time series forecasting. It designs a memory module to mitigate constraints from insufficient historical data while integrating time series decomposition technique within the Transformer architecture, enabling it to comprehensively capture temporal dependencies and global patterns, enhancing forecasting accuracy.
- We conduct extensive experiments to evaluate our model on three datasets. The results show that our model outperforms various baseline methods, demonstrating the effectiveness of MEMSTSF in short time series forecasting.

## 2 Related Work

**Deep Learning for Time Series Forecasting.** With the development of deep learning, compared to traditional models, such as ARIMA [1], Neural models have been proposed and achieve superior performance in multiple applications. Recurrent neural networks RNNs and their variants long-short term memory LSTM [7] were designed especially for tasks involving sequential data. In recent years, due to the impressive performance of Transformers in handling sequential data, there has been a surge of research applying them to time series forecasting. Informer [27] designed a ProbSparse attention mechanism and distilling operation to reduce time complexity and memory usage in vanilla Transformer. FEDformer [29] proposes a frequency-enhanced transformer model and gets linear computational complexity and memory cost. Innovatively, Crossformer [26] utilizes cross-dimension dependency to use the information at different scales for time-series forecasting. Currently, many models focus on long-term forecasting problems that require a significant amount of training data, there is a scarcity of models explicitly tailored for short time series forecasting tasks.

**Decomposition for Time Series Forecasting.** Time series decomposition is a classical time series analysis technique that breaks down complex time series data into components with different time scales and features, helping to understand the data structure and enabling more accurate prediction. Prophet [18] decomposes the time series into multiple components such as trend, seasonality, and holidays, and then fits them using an additive model. Neuralprophet [19] is a successor to Prophet which bridges the gap between traditional time-series models and deep learning methods by decomposing complex time series into six components. Some deep-learning methods tackle forecasting with the assistance of decomposition. Autoformer [23] harnesses the decomposition as an inner block

of deep models, so it can decompose the hidden series throughout the whole forecasting process. LaST [21] utilizes an encoder-decoder architecture and follows variational inference theory to learn disentangled latent representations that describe seasonal and trends of time series.

**Memory Network.** Memory network framework was initially proposed by [22] for knowledge question-answering. It combines inference components with a memory unit that is capable of both reading and writing to remember supporting facts from the past for question answering. Memory networks generally incorporate attention mechanisms, most methods [4,5,17] adopt the attention mechanism to read the memory unit. The flexibility of the memory network architecture allows it to perform well in many application domains, such as question answering (QA) [9], visual question answering (VQA) [24], recommendation system [3]. However, there are currently few works [2,**?**] that integrate memory networks with time series forecasting. Our paper suggests that incorporating the idea of memory network could better address the challenges of short time series forecasting with insufficient historical data.

## 3 Our Proposed Model

In this section, we present the details of our model for solving the short time series forecasting problem which is defined as:

**Definition 1 (Short Time Series Forecasting).** *Given a time series with time-dependent variables* $X = [x_1, x_2, \cdots, x_T] \in R^{T \times D}$*, where* $T$ *is the number of input time steps,* $D$ *represents the variable dimension. Our task is to predict* $Y = [y_1, y_2, \cdots, y_L] \in R^{L \times D}$ *with* $L$ *future timestep values. We denote the problem of predicting a future series of length* $L$ *based on a historical series of length* $T$ *as input-*$T$*-predict-*$L$*, the short time series forecasting setting emphasizes shorter input, i.e. smaller* $T$*.*

The MEMSTSF model introduces an innovative approach by combining a Transformer-based architecture with a memory module to overcome the challenge of limited historical data. This design enables the model to capture complex temporal dependencies, incorporate external information as memory, and enrich short time series with key features. Furthermore, MEMSTSF uniquely integrates a time series decomposition mechanism to extract global properties like seasonality and periodicity, improving the temporal representation of the data. By ingeniously integrating the memory module and the decomposition mechanism, MEMSTSF effectively addresses the challenge of short time series forecasting while capturing the intrinsic features, significantly enhancing its forecasting performance. Figure 1 illustrates the overall framework of MEMSTSF.

### 3.1 Transformer with Memory

To address the challenge of short time series forecasting, we have designed a novel Transformer architecture with a memory module to enhance the input sequence before it enters the Transformer encoder module for encoding.

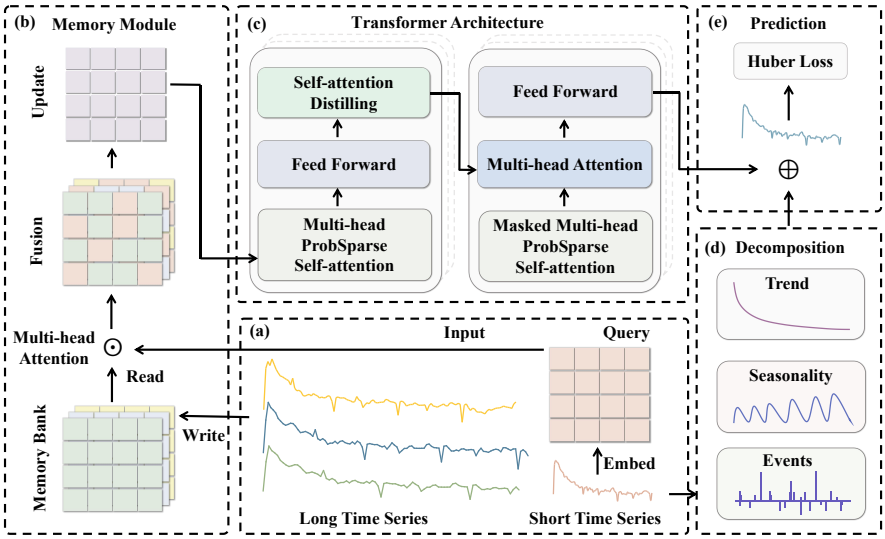


Fig. 1: The overall framework of MEMSTSF: (a) It first encodes the long time series into the memory module; (b) Next, the short time series retrieves key information from the memory module; (c) Then, it uses a Transformer architecture to encode and decode the updated short time series; (d) Afterward, it employs a time series decomposition module to learn global properties of time series. (e) Finally, it integrates the outputs of (c) and (d) to generate predictions.

**Memory Module** Despite the limitations of insufficient historical data and unstable patterns in short time series, long time series from the same scenario may contain valuable information. For instance, a newly established wind farm can leverage wind speed data from older wind farms to optimize forecasting performance. In the business domain, by referencing the behavior data of mature customers, it is possible to predict the behavioral trends of new customers more accurately. Inspired by this, MEMSTSF introduces an innovative design by incorporating a memory module to leverage long time series information, compensating for the limitations of short time series forecasting.

Specifically, for each short time series $X = [x_1, x_2, \cdots, x_T]$ obtained through the sliding window, we randomly select $n$ slices from other long sequences as set $M = [m_1, m_2, \cdots, m_n]$ to store in the memory module, where $m_i \in R^{T_2 \times D}$ represents $i$-th time series in set $M$, $T_2$ represents the length of long time series $m_i$, both long and short time series having the same variable dimension $D$. Next, we embed short time series $X$ and each long time series in set $M$ to obtain the input and memory vector. As an efficient information fusion method, the attention mechanism [20] has accomplished remarkable success in many fields. To fully utilize the rich information in long time series, we utilize the cross-attention method to integrate long and short time series. Specifically, we employ

the Query-Key-Value (QKV) model to process each long time series in memory vector $M$, treating the short time series $X$ as the Query and each long time series $m_i$ as Key and Value. By calculating the correlation between them, the short time series can focus on the key information from the long time series $m_i$, generating an enhanced representation $X_i^*$ of the short time series:

$$X_i^* = softmax(\frac{Xm_i^T}{\sqrt{D}})m_i. \tag{1}$$

Additionally, we employ a multi-head attention mechanism to execute cross-attention operations in parallel. The specific process is:

$$\begin{aligned} MultiHead(Q,K,V) &= Concat(head_1, \cdots, head_h)W^{output} \\ head_i &= Attention(QW_{Qi}, KW_{Ki}, VW_{Vi}) \\ Attention(Q,K,V) &= softmax(\frac{QK^T}{\sqrt{D}})V. \end{aligned} \tag{2}$$

where the projections of Query, Key, Value, and the output layer are parameter matrices $W_{Qi} \in R^{D\times d_k}$, $W_{Ki} \in R^{D\times d_k}$, $W_{Vi} \in R^{D\times d_v}$, and $W^{output} \in R^{hd_v\times D}$, $h$ represents the number of heads, $d_k = d_v = D/h$. Next, we repeat the operations mentioned above for all long time series in the memory component, intelligently focusing on crucial information from different long sequences. Finally, by averaging all updated short time series, we obtain the input embedding $I$ for the informer encoder:

$$I = Mean(X_1^*, X_2^*, \cdots, X_n^*). \tag{3}$$

This strategy enhances the model input by introducing an external memory module to the Transformer architecture. Learning patterns from other sequences effectively compensates for the limitations of short time series forecasting.

**Encoder** The Informer network [27] is a successful model based on the Transformer framework, specifically designed for time series forecasting. After enhancing the short time series using the memory module, we follow the design of the Informer encoder to encode the enhanced sequence. The encoder consists of multiple identical blocks, each including a multi-head probsparse self-attention module, a feed-forward network, and a self-attention distillation module. Exploiting the sparsity characteristic of the vanilla self-attention mechanism, the probsparse self-attention module selectively focuses on the most relevant parts of the input sequence by calculating only a few important dot-product from the entire self-attention scores. This reduces computational complexity and improves efficiency, and is defined as follows:

$$Attention(\bar{Q}, K, V) = softmax(\frac{\bar{Q}K^T}{\sqrt{D}})V. \tag{4}$$

where $\bar{Q}$ is a sparse matrix that only contains the major attention. Self-attention distillation module further extracts crucial attention information, the distillation operation from the $j$-th layer to the $(j+1)$-th layer is as follows:

$$X_{j+1}^t = MaxPool(ELU(Conv1d([X_j^t]))). \tag{5}$$

where $X_j^t$ represents the $j$-th layer of the $t$-th time series input, $[\cdot]$ represents the attention block, including the ProbSparse self-attention and essential operations. $Conv1d(\cdot)$ represents the convolutional filter, $ELU(\cdot)$ is the activation function. $MaxPool(\cdot)$ reduces the output dimension of each layer through max pooling.

**Decoder** After the encoder processes the enhanced input time series, the decoder generates the prediction series. It is constructed with multiple identical blocks, each encompassing a masked multi-head probsparse self-attention module, a multi-head attention module, and a feed-forward network. To avoid cumulative errors during inference, it predicts all future elements of the sequence at once instead of recursively. Specifically, the decoder input is formulated as:

$$X_{de}^t = Concat(X_{start}^t, X_0^t) \in R^{(L_{start}+L)\times D}. \tag{6}$$

where $X_{start}^t \in R^{L_{start}\times D}$ represents the start token sampled from the short time series $X$, indicating an earlier slice before the predicted sequence. $X_0^t \in R^{L\times D}$ is a placeholder for the target series, with all elements set to 0. $D$ is the variable dimension of the time series. The decoder's output length is $L_{start}+L$, we only take the latter $L$ length as the predicted sequence.

### 3.2 Time Series Decomposition

Inspired by the time series decomposition approach in NeuralProphet [19], we designed a decomposition module for MEMSTSF to capture key temporal features of raw short time series comprehensively. The module includes the following components, with trend and seasonality components enabled by default.

**Trend** MEMSTSF models the trend component as a continuous piecewise linear series, representing the trend as a combination of offset and growth rate, which are expressed through time-dependent growth rate $\delta(t)$ and offset $\rho(t)$. The trend effect between two timestamps is determined by multiplying the constant growth rate by the time difference, which can be represented as $T(t) = \delta(t) \cdot t + \rho(t)$. Furthermore, to enhance the flexibility of the trend component, we allow the growth rate to vary at multiple locations. Representing the set of $n_c$ changepoints as $C = (c_1, c_2, \cdots, c_n)$, the growth rate remains constant between changepoints. The growth rate and the offset of the first segment are represented by $k$ and $g$, respectively. Vector $\delta = (\delta_1, \delta_2, \cdots, \delta_{n_c})$ represents the growth rate adjustments at each changepoint. The growth rate at timestamp $t$ is calculated by summing the initial growth rate $k$ and the cumulative adjustments from all changepoints up to timestamp $t$. Similarly, Vector $\rho = (\rho_1, \rho_2, \cdots, \rho_{n_c})$ represents the offset

adjustments. The offset at timestamp $t$ is determined by adding the initial offset $g$ with the sum of offset adjustments at each changepoint up to timestamp $t$. Therefore, the trend $T(t)$ is defined as follows:

$$T(t) = (k + a(t)^T\delta) \cdot t + (g + a(t)^T\rho). \tag{7}$$

where $a(t) = (a_1(t), a_2(t), \cdots, a_{n_c}(t))$ is a binary vector indicating whether timestamp $t$ has passed each changepoint. Specifically, if $t \geq c_j$, then $a_j(t) = 1$; otherwise, $a_j(t) = 0$.

**Seasonality** Next, we utilize Fourier terms [6] to model the seasonal component, employing pairs of sine and cosine functions for flexible multi-seasonality modeling. For instance, let $P$ represent the period of the time series, fourier terms can be employed to model yearly seasonality ($P = 365.25$) or weekly seasonality ($P = 52.18$). We define multiple Fourier terms as follows for each seasonality to deal with multi-seasonality scenarios:

$$S_p(t) = \sum_{j=1}^{n}(a_j \cdot \cos(\frac{2\pi jt}{p}) + b_j \cdot \sin(\frac{2\pi jt}{p})). \tag{8}$$

where $n$ represents the number of Fourier terms defined for the seasonality with periodicity $p$. Ultimately, the seasonality at timestamp $t$ is represented by considering the combined effect of all seasonal factors, which is denoted as $S(t) = \sum_{p\in P}(S_p(t))$. We can automatically activate yearly, weekly, or daily seasonality based on the time series frequency and length.

**Events and Holidays** Considering factors such as events and holidays is crucial in time series forecasting. While these factors typically do not follow regular periodic patterns, the effects of specific holidays tend to exhibit similar across different years. We define each event $e$ as a binary variable $e \in [0, 1]$, indicating whether the event occurs. The vector $E \in R^{T\times n_e}$ represents the combination of all events, where $n_e$ is the number of events and $T$ is the length of the time series. The impact of events at timestamp $t$ is represented as $R(t) = \sum_{e\in E}(E_e(t))$, where $E_e(t) = z_e e(t)$, $z_e$ is the model coefficient corresponding to the event $e$.

### 3.3 Overall

MEMSTSF integrates the memory-augmented Transformer module and the time series decomposition module to form a decomposable time series model, with each module contributing an additional component to the prediction. Specifically, the final prediction $Y(t)$ is the sum of the outcomes from each module.

$$Y(t) = Transformer(t) + Trend(t) + Season(t) + Holiday(t). \tag{9}$$

Where $Transformer(t)$ represents the memory-augmented Transformer architecture handling data enhanced by long sequences, the latter three components represent the decomposition module handling the original short time series.

### 3.4 Objective Function

We utilize the Huber loss function to predict the target sequence and calculate the loss. The Huber loss combines the advantages of both Mean Squared Error (MSE) and Mean Absolute Error (MAE), offering a way to balance accuracy and robustness across different error ranges. Here, $\delta$ is an adjustable hyperparameter that controls the balance between MSE and MAE in the Huber loss function.

$$Loss(y, \hat{y}) = \begin{cases} \frac{1}{2}(y - \hat{y})^2, & if\ |y - \hat{y}| \leq \delta \\ \delta\,|y - \hat{y}| - \frac{1}{2}\delta^2, & if\ |y - \hat{y}| > \delta. \end{cases} \tag{10}$$

## 4 Experiments

In this section, we compare the performance of MEMSTSF with multiple baseline models on three datasets. Additionally, we conduct an ablation study, perform a model sensitivity analysis, and present an interpretable prediction example to demonstrate the model's reliability and effectiveness.

### 4.1 Experiment Setup

**Datasets** In the selection of datasets, although public benchmark datasets offer abundant data, such as ETT[1] and Electricity[2], they only contain one single time series, which do not align with our research scenario. We comprehensively evaluate MEMSTSF through extensive experiments on three real-world datasets, each containing multiple time series from the same scenario.

- **Funds**: This is an industrial dataset from the financial domain. It contains daily fund records of 31 distinct user groups, including 12 new user groups with only 90 days of data records, and 19 mature user groups with abundant historical data, averaging a length of 839 days.
- **HARTH**: The HARTH [13] dataset contains acceleration data from 22 participants, each wearing two 3-axis accelerometers, collected over 2 hours in a free-living environment.
- **HAR70+**: The HAR70+ [12] dataset contains acceleration data from 18 frail elderly individuals, each wearing two 3-axis accelerometers, collected over 40 minutes in a semi-structured free-living setting.

In the Funds dataset, the records of new and mature user groups can be regarded as short and long time series, respectively. All subjects within the HARTH and HAR70+ datasets have abundant records. To simulate cold start scenarios, we truncate data from certain subjects in HARTH and HAR70+ to 100-length short sequences, while data from other subjects are used as long sequences to provide auxiliary information. After preprocessing, the HARTH dataset contains 7 short sequences (length 100) and 10 long sequences (average length 2000), while the HAR70+ dataset contains 6 short sequences (length 100) and 10 long sequences (average length 743). Detailed statistics are shown in Table 1.

[1] https://github.com/zhouhaoyi/ETDataset.
[2] https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014.

Table 1: Statistics of datasets.

| Datasets | Funds | HARTH | HAR70+ |
|---|---|---|---|
| Variates | 19 | 6 | 6 |
| Number of long series | 19 | 10 | 10 |
| Number of short series | 12 | 7 | 6 |
| Timesteps of short series | 90 | 100 | 100 |
| Granularity | 1 day | 1 second | 1 second |

**Baselines** To thoroughly evaluate the prediction accuracy of our model, we selected 12 diverse baseline models for comparison, covering both traditional methods and neural network-based approaches commonly used in time series forecasting: Transformer [20], Informer [27], Autoformer [23], Fedformer [29], Crossformer [26], PatchTST [15], FiLM [28], LaST [21], DLinear [25], ARIMA [1], Prophet [18], NeuralProphet [19].

**Implementation Details** Our model employs the Huber loss function with AdamW [14] optimizer, with a default batch size of 32 and 8 heads in multi-head attention mechanism. The learning rate is selected through grid search from the set $\{1e-2, 1e-3, 1e-4, 1e-5, 5e-6\}$. Training is early stopped within 100 epochs and all experiments are repeated twice, with final results being the average of the two runs. The Transformer component contains 2 encoder layers and 1 decoder layer. All methods are trained and tested on a CentOS machine of 128G memory with two Intel(R) Xeon(R) Silver4210R 2.40GHz CPUs and an NVIDIA A40 GPU.

**Evaluation Metrics** We evaluate all models using two common metrics in time series forecasting: Mean Squared Error (MSE) and Mean Absolute Error (MAE), defined as $MSE = \frac{1}{n}\sum_{i=1}^{n}(y-\hat{y})^2$ and $MAE = \frac{1}{n}\sum_{i=1}^{n}|y-\hat{y}|$, where n is the number of samples, $y$ and $\hat{y}$ are ground truth and predictions, respectively. Lower values of these metrics indicate better performance.

### 4.2 Performance Comparison

To compare performance across different prediction horizons, we set a fixed input length 7 and prediction length $\in \{5, 7, 10, 15\}$. This setting aligns with the definition of short time series forecasting with insufficient historical data. Since baseline models lack memory module, they are unable to fully capture data patterns from longer sequences. To ensure a fair comparison, we supplement the baseline models with long sequences as additional training data, enabling them to better learn diverse time series patterns and improve performance. We report the average prediction results of all models in Table 2. The overall results demonstrate that MEMSTSF outperforms most baseline models

Table 2: Multivariate time series forecasting results on three datasets with input length $T = 7$ and prediction length $L \in \{5, 7, 10, 15\}$. The best and suboptimal results are highlighted in bold and underlined, respectively.

| Model | Metrics | Funds | | | | HARTH | | | | HAR70+ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 5 | 7 | 10 | 15 | 5 | 7 | 10 | 15 | 5 | 7 | 10 | 15 |
| Transformer | MSE | 0.020 | 0.024 | 0.030 | 0.033 | 0.547 | 0.562 | 0.568 | 0.572 | 0.930 | 0.949 | 0.953 | 0.981 |
| | MAE | 0.094 | 0.100 | 0.108 | 0.116 | 0.428 | 0.441 | 0.451 | 0.471 | 0.686 | 0.694 | 0.701 | 0.714 |
| Informer | MSE | 0.013 | 0.017 | 0.022 | 0.032 | 0.591 | 0.604 | 0.598 | 0.583 | 0.898 | 0.932 | 0.955 | 0.986 |
| | MAE | 0.080 | 0.087 | 0.098 | 0.120 | 0.461 | 0.471 | 0.479 | 0.486 | 0.673 | 0.690 | 0.700 | 0.710 |
| Autoformer | MSE | 0.042 | 0.049 | 0.082 | 0.145 | 0.623 | 0.661 | 0.711 | 0.786 | 0.956 | 0.993 | 1.039 | 1.109 |
| | MAE | 0.118 | 0.139 | 0.190 | 0.258 | 0.424 | 0.441 | 0.472 | 0.512 | 0.623 | 0.642 | 0.664 | 0.697 |
| Fedformer | MSE | 0.032 | 0.042 | 0.054 | 0.088 | 0.695 | 0.731 | 0.780 | 0.838 | 0.930 | 0.971 | 1.042 | 1.131 |
| | MAE | 0.108 | 0.128 | 0.151 | 0.205 | 0.453 | 0.473 | 0.498 | 0.533 | 0.621 | 0.640 | 0.669 | 0.708 |
| Crossformer | MSE | 0.048 | 0.057 | 0.069 | 0.093 | 0.508 | 0.522 | 0.538 | 0.553 | 0.883 | 0.908 | 0.926 | 0.942 |
| | MAE | 0.133 | 0.148 | 0.168 | 0.211 | 0.401 | 0.423 | 0.442 | 0.470 | 0.652 | 0.664 | 0.677 | 0.688 |
| PatchTST | MSE | 0.110 | 0.153 | 0.199 | 0.304 | 0.689 | 0.731 | 0.773 | 0.845 | 0.937 | 0.975 | 1.040 | 1.127 |
| | MAE | 0.183 | 0.238 | 0.280 | 0.379 | 0.439 | 0.461 | 0.481 | 0.518 | 0.609 | 0.627 | 0.654 | 0.693 |
| FiLM | MSE | 0.310 | 0.253 | 0.294 | 0.351 | 0.784 | 0.778 | 0.708 | 0.849 | 1.063 | 1.032 | 1.068 | 1.110 |
| | MAE | 0.441 | 0.318 | 0.346 | 0.391 | 0.633 | 0.607 | 0.523 | 0.645 | 0.773 | 0.756 | 0.770 | 0.787 |
| LaST | MSE | 0.091 | 0.112 | 0.145 | 0.191 | 0.643 | 0.677 | 0.709 | 0.755 | 0.876 | 0.893 | 0.922 | 0.965 |
| | MAE | 0.186 | 0.230 | 0.276 | 0.337 | 0.443 | 0.459 | 0.485 | 0.516 | 0.658 | 0.665 | 0.676 | 0.694 |
| DLinear | MSE | 0.483 | 0.424 | 0.415 | 0.419 | 0.754 | 0.745 | 0.749 | 0.755 | 1.266 | 1.313 | 1.247 | 1.280 |
| | MAE | 0.477 | 0.502 | 0.494 | 0.497 | 0.567 | 0.582 | 0.581 | 0.583 | 0.812 | 0.851 | 0.824 | 0.837 |
| ARIMA | MSE | 0.107 | 0.129 | 0.161 | 0.214 | 0.108 | 0.112 | 0.121 | 0.132 | 0.102 | 0.111 | 0.122 | 0.143 |
| | MAE | 0.205 | 0.229 | 0.257 | 0.297 | 0.158 | 0.162 | 0.170 | 0.179 | 0.184 | **0.193** | **0.207** | **0.227** |
| Prophet | MSE | 0.788 | 0.800 | 0.819 | 0.857 | 0.185 | 0.179 | 0.172 | 0.172 | 0.172 | 0.165 | 0.182 | 0.195 |
| | MAE | 0.755 | 0.764 | 0.776 | 0.798 | 0.324 | 0.325 | 0.338 | 0.344 | 0.319 | 0.322 | 0.328 | 0.332 |
| NeuralProphet | MSE | 0.044 | 0.044 | 0.046 | 0.051 | 0.076 | 0.078 | 0.081 | 0.087 | 0.087 | 0.108 | 0.117 | 0.126 |
| | MAE | 0.149 | 0.152 | 0.153 | 0.163 | 0.161 | 0.165 | 0.170 | 0.179 | 0.204 | 0.238 | 0.239 | 0.263 |
| **MEMSTSF** | MSE | **0.011** | **0.012** | **0.015** | **0.023** | **0.067** | **0.065** | **0.066** | **0.071** | **0.063** | **0.082** | **0.101** | **0.110** |
| | MAE | **0.074** | **0.079** | **0.087** | **0.109** | **0.153** | **0.150** | **0.154** | **0.160** | **0.179** | 0.215 | 0.238 | 0.246 |

in all settings. Specifically, under the input-7-predict-10 setting, compared to the best-performing baseline, MEMSTSF gives **32%**(0.022 → 0.015) MSE and **12%**(0.098 → 0.087) MAE reduction in Funds dataset. For the input-7-predict-15 setting of HARTH dataset, MEMSTSF makes **19%**(0.087 → 0.071) MSE reduction and **11%**(0.179 → 0.160) MAE reduction in comparison to the optimal baseline model. Overall, MEMSTSF achieves an average **21%** reduction in MSE, demonstrating its superior performance in short time series forecasting scenarios. Among the baseline models, Informer and NeuralProphet stand out for their strong performance in short time series forecasting.

Table 3: Ablation study on datasets Funds and HAR70+ with input length $T = 7$ and prediction length $L \in \{5, 7, 10, 15\}$.

| Model | | MEMSTSF | | w/o Decomposition | | w/o Memory | | w/o Transformer | | Transformer variant | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE |
| Funds | 5 | $\mathbf{0.011}_{\pm\mathbf{0.008}}$ | $\mathbf{0.074}_{\pm\mathbf{0.029}}$ | $0.023_{\pm0.008}$ | $0.112_{\pm0.022}$ | $0.020_{\pm0.014}$ | $0.093_{\pm0.037}$ | $0.128_{\pm0.065}$ | $0.250_{\pm0.067}$ | $0.013_{\pm0.008}$ | $0.083_{\pm0.027}$ |
| | 7 | $\mathbf{0.012}_{\pm\mathbf{0.008}}$ | $\mathbf{0.079}_{\pm\mathbf{0.028}}$ | $0.024_{\pm0.008}$ | $0.115_{\pm0.023}$ | $0.015_{\pm0.010}$ | $0.083_{\pm0.029}$ | $0.123_{\pm0.063}$ | $0.251_{\pm0.067}$ | $0.015_{\pm0.009}$ | $0.092_{\pm0.033}$ |
| | 10 | $\mathbf{0.015}_{\pm\mathbf{0.009}}$ | $\mathbf{0.087}_{\pm\mathbf{0.029}}$ | $0.025_{\pm0.009}$ | $0.119_{\pm0.023}$ | $0.019_{\pm0.012}$ | $0.098_{\pm0.034}$ | $0.124_{\pm0.063}$ | $0.258_{\pm0.070}$ | $0.016_{\pm0.011}$ | $0.096_{\pm0.034}$ |
| | 15 | $\mathbf{0.023}_{\pm\mathbf{0.012}}$ | $\mathbf{0.109}_{\pm\mathbf{0.027}}$ | $0.029_{\pm0.011}$ | $0.127_{\pm0.024}$ | $0.040_{\pm0.044}$ | $0.130_{\pm0.065}$ | $0.125_{\pm0.064}$ | $0.266_{\pm0.075}$ | $0.025_{\pm0.012}$ | $0.124_{\pm0.036}$ |
| HAR70+ | 5 | $\mathbf{0.063}_{\pm\mathbf{0.029}}$ | $\mathbf{0.179}_{\pm\mathbf{0.044}}$ | $0.125_{\pm0.072}$ | $0.264_{\pm0.119}$ | $0.112_{\pm0.064}$ | $0.246_{\pm0.089}$ | $0.424_{\pm0.243}$ | $0.521_{\pm0.192}$ | $0.095_{\pm0.044}$ | $0.226_{\pm0.072}$ |
| | 7 | $\mathbf{0.082}_{\pm\mathbf{0.036}}$ | $\mathbf{0.215}_{\pm\mathbf{0.061}}$ | $0.118_{\pm0.073}$ | $0.257_{\pm0.119}$ | $0.110_{\pm0.067}$ | $0.241_{\pm0.08}$ | $0.426_{\pm0.245}$ | $0.523_{\pm0.193}$ | $0.107_{\pm0.055}$ | $0.238_{\pm0.081}$ |
| | 10 | $\mathbf{0.101}_{\pm\mathbf{0.053}}$ | $\mathbf{0.238}_{\pm\mathbf{0.087}}$ | $0.137_{\pm0.093}$ | $0.285_{\pm0.142}$ | $0.136_{\pm0.082}$ | $0.283_{\pm0.101}$ | $0.430_{\pm0.247}$ | $0.526_{\pm0.194}$ | $0.124_{\pm0.069}$ | $0.260_{\pm0.095}$ |
| | 15 | $\mathbf{0.110}_{\pm\mathbf{0.06}}$ | $\mathbf{0.246}_{\pm\mathbf{0.089}}$ | $0.130_{\pm0.079}$ | $0.281_{\pm0.123}$ | $0.170_{\pm0.123}$ | $0.308_{\pm0.137}$ | $0.475_{\pm0.303}$ | $0.550_{\pm0.22}$ | $0.133_{\pm0.078}$ | $0.277_{\pm0.109}$ |

### 4.3 Ablation Study

To better understand the contribution of key components, we conduct ablation studies on the Funds and HAR70+ datasets under different prediction horizons. By disabling different components, we obtained the following variants:

- **w/o Memory**: Without the memory module, unable to leverage auxiliary information from long sequences.
- **w/o Transformer**: Disregarding the Transformer architecture, using only the time series decomposition module for predictions.
- **w/o Decomposition** : Remove the time series decomposition module and use only the Transformer architecture with memory module in MEMSTSF.
- **Transformer variant**: To further investigate the importance of the deep learning component, Replacing MEMSTSF's encoder-decoder with the vanilla Transformer's encoder-decoder for time series modeling.

The results of different variants and MEMSTSF are reported in Table 3. We observe that MEMSTSF outperforms the variants in all cases. Specifically, MEMSTSF surpasses the "w/o Memory" variant, emphasizing the key role of the memory module in leveraging auxiliary long-sequence information to enhance short time series forecasting. Additionally, MEMSTSF performs better than the "w/o Transformer" variant, which in turn outperforms the Transformer variant. This demonstrates the advantages of MEMSTSF in time series modeling as well as the superior ability of capturing temporal dependencies. Moreover, MEMSTSF also exhibits superior performance than the "w/o Decomposition" variant, showcasing the value of the time series decomposition module in capturing global patterns and intricate temporal relationships.

### 4.4 Model Sensitivity of Prediction Length

In Figure 2, we conducted a sensitivity analysis on prediction length, showcasing the MSE and MAE results of MEMSTSF and two baseline models on the

Funds and HARTH datasets. The results reveal that as the prediction length increases, the MAE and MSE gradually decrease while consistently maintaining a superior level compared to the baseline models with smooth transitions. This indicates that MEMSTSF excels in short time series forecasting and demonstrates exceptional robustness and adaptability across different prediction lengths.

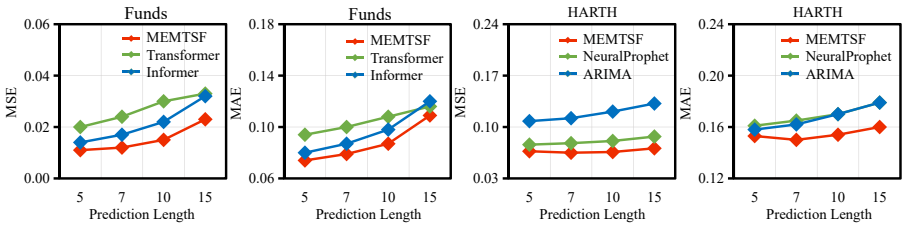


Fig. 2: Performance of MEMSTSF and two best-performing baseline models on datasets Funds and HARTH under different prediction lengths.

### 4.5 Interpretable Example

**Visualization of Forecasting Results** Here we present the true values of a time series from the Funds dataset, along with the prediction results from MEMSTSF and the optimal baseline model Informer under four different prediction length settings in Figure 3. The horizontal axis represents timestamps, while the vertical axis corresponds to the actual values. The GroundTruth curve shows a downward trend with unstable patterns and multiple change points. MEMSTSF accurately fits the changing trends of this sequence in all settings and adapts to multiple mutation points. As prediction length increases, MEMSTSF's fit to the true values slightly decreases but remains strong, showing its robustness across varying horizons. In contrast, Informer's predictions consistently deviate more from the GroundTruth, especially at change points, making it harder to accurately capture the true curve. This further demonstrates MEMSTSF's superiority in predicting unstable, short time series and capturing subtle changes.

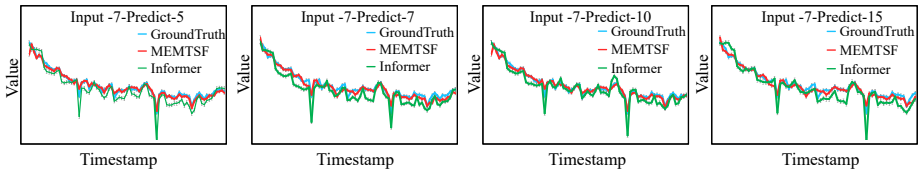


Fig. 3: Case study of GroundTruth and prediction results from MEMSTSF and the best-performing baseline on one time series in Funds dataset.

**Interpretable Component Decomposition** MEMSTSF is a decomposable time series forecasting model with multiple modules, each contributing to the final result and capable of individual visualization. This design allows users to intuitively analyze the contribution of each component to the final result, enhancing the interpretability of the predictions, which is very useful for practical applications. Here we take a daily fund time series from the Funds dataset as an example, visualizing its GroundTruth, predictions, and three components' results in Figure 4, where the horizontal axis denotes dates and the vertical axis denotes fund changes. The trend component precisely captures the overall downward trend, aligning with the predictions from the Transformer module. Furthermore, the seasonal component reveals the cyclical change pattern and periodic fluctuations in the funds at different times of each week. In this way, we can offer robust assistance in predicting future fund flows, providing reliable decision support for real-world applications based on time series forecasting.

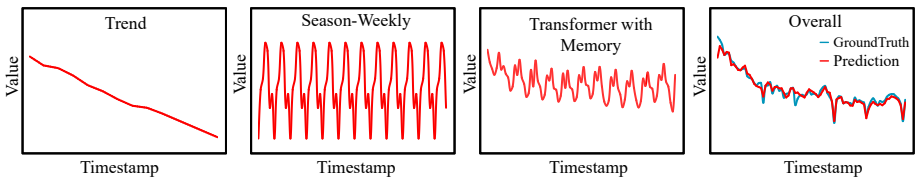


Fig. 4: Visualization of prediction results from different components on one fund time series in the Funds dataset.

## 5 Conclusion

In this paper, we propose MEMSTSF, a novel memory-augmented short time series forecasting model, which is particularly designed to address the issue of limited historical data in time series prediction. MEMSTSF designs an inventive Transformer architecture that captures the temporal dependencies and interactions within the time series and incorporates an innovative memory module. This is the key module for MEMSTSF to tackle the cold start problem, enabling MEMSTSF to utilize supplementary information of long time series from multiple data sources and assist short time series in extracting valuable data patterns and regularities. Furthermore, MEMSTSF employs time series decomposition techniques to capture temporal features such as seasonality, periodicity, and events, enhancing its understanding of complex temporal patterns. Extensive experiments on three datasets have demonstrated the outstanding effectiveness of MEMSTSF in enhancing short time series forecasting performance.

## Acknowledge

This work was supported by NSFC(No.62472154).

## References


1. Box, G.E., Jenkins, G.M.: Some recent advances in forecasting and control. Journal of the Royal Statistical Society. Series C (Applied Statistics) **17**(2), 91–109 (1968)
2. Chang, Y.Y., Sun, F.Y., Wu, Y.H., Lin, S.D.: A memory-network based solution for multivariate time-series forecasting. arXiv preprint arXiv:1809.02105 (2018)
3. Ebesu, T., Shen, B., Fang, Y.: Collaborative memory network for recommendation systems. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. pp. 515–524 (2018)
4. Graves, A., Wayne, G., Danihelka, I.: Neural turing machines. CoRR **abs/1410.5401** (2014), `http://arxiv.org/abs/1410.5401`
5. Graves, A., Wayne, G., Reynolds, M., Harley, T., Danihelka, I., Grabska-Barwinska, A., Colmenarejo, S.G., Grefenstette, E., Ramalho, T., Agapiou, J.P., Badia, A.P., Hermann, K.M., Zwols, Y., Ostrovski, G., Cain, A., King, H., Summerfield, C., Blunsom, P., Kavukcuoglu, K., Hassabis, D.: Hybrid computing using a neural network with dynamic external memory. Nature **538**(7626), 471–476 (2016)
6. Harvey, A.C., Shephard, N.: Structural time series models, vol. Vol. 11:Econometrics, pp. 261–302. North Holland, Amsterdam, (edited by g.s. maddala, c.r. rao and h.d. vinod) edn. (1993)
7. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
8. Hu, Q., Zhang, R., Zhou, Y.: Transfer learning for short-term wind speed prediction with deep neural networks. Renewable Energy **85**, 83–95 (2016)
9. Kumar, A., Irsoy, O., Ondruska, P., Iyyer, M., Bradbury, J., Gulrajani, I., Zhong, V., Paulus, R., Socher, R.: Ask me anything: Dynamic memory networks for natural language processing. In: Proceedings of the 33nd International Conference on Machine Learning. vol. 48, pp. 1378–1387. JMLR.org (2016)
10. Li, Y., Lu, X., Wang, Y., Dou, D.: Generative time series forecasting with diffusion, denoise, and disentanglement. In: Advances in Neural Information Processing Systems. vol. 35, pp. 23009–23022 (2022)
11. Liu, S., Yu, H., Liao, C., Li, J., Lin, W., Liu, A.X., Dustdar, S.: Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In: International conference on learning representations (2022)
12. Logacjov, A., Ustad, A.: HAR70+. UCI Machine Learning Repository (2023), DOI: https://doi.org/10.24432/C5CW3D
13. Logacjov, Aleksej, K.A.B.K.B.H.B., Mork, P.J.: HARTH. UCI Machine Learning Repository (2023), DOI: https://doi.org/10.24432/C5NC90
14. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: 7th International Conference on Learning Representations. OpenReview.net (2019)
15. Nie, Y., Nguyen, N.H., Sinthong, P., Kalagnanam, J.: A time series is worth 64 words: Long-term forecasting with transformers. In: The Eleventh International Conference on Learning Representations (2023)
16. Shi, Q., Yin, J., Cai, J., Cichocki, A., Yokota, T., Chen, L., Yuan, M., Zeng, J.: Block hankel tensor ARIMA for multiple short time series forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 5758–5766 (2020)
17. Sukhbaatar, S., Szlam, A., Weston, J., Fergus, R.: End-to-end memory networks. In: Advances in Neural Information Processing Systems. vol. 28, pp. 2440–2448 (2015)
18. Taylor, S.J., Letham, B.: Forecasting at scale. PeerJ Prepr. **5**, e3190 (2017)

19. Triebe, O., Hewamalage, H., Pilyugina, P., Laptev, N., Bergmeir, C., Rajagopal, R.: Neuralprophet: Explainable forecasting at scale. CoRR **abs/2111.15397** (2021), https://arxiv.org/abs/2111.15397
20. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems 30. vol. 30, pp. 5998–6008 (2017)
21. Wang, Z., Xu, X., Zhang, W., Trajcevski, G., Zhong, T., Zhou, F.: Learning latent seasonal-trend representations for time series forecasting. In: Advances in Neural Information Processing Systems (2022)
22. Weston, J., Chopra, S., Bordes, A.: Memory networks. In: 3rd International Conference on Learning Representations (2015), http://arxiv.org/abs/1410.3916
23. Wu, H., Xu, J., Wang, J., Long, M.: Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In: Advances in Neural Information Processing Systems. pp. 22419–22430 (2021)
24. Xu, H., Saenko, K.: Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In: European Conference on Computer Vision. vol. 9911, pp. 451–466 (2016)
25. Zeng, A., Chen, M., Zhang, L., Xu, Q.: Are transformers effective for time series forecasting? In: Thirty-Seventh AAAI Conference on Artificial Intelligence. pp. 11121–11128 (2023)
26. Zhang, Y., Yan, J.: Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In: The Eleventh International Conference on Learning Representations (2023)
27. Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., Zhang, W.: Informer: Beyond efficient transformer for long sequence time-series forecasting. In: The Thirty-Fifth AAAI Conference on Artificial Intelligence. vol. 35, pp. 11106–11115 (2021)
28. Zhou, T., Ma, Z., Wang, X., Wen, Q., Sun, L., Yao, T., Yin, W., Jin, R.: Film: Frequency improved legendre memory model for long-term time series forecasting. In: Advances in Neural Information Processing Systems. vol. 35, pp. 12677–12690 (2022)
29. Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., Jin, R.: Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In: Proc. 39th International Conference on Machine Learning. vol. 162, pp. 27268–27286 (2022)