LLM-Mixer: Multiscale Mixing in LLMs for Time Series Forecasting

Md Kowsher[1], Md. Shohanur Islam Sobuj[2], Nusrat Jahan Prottasha[1], E. Alejandro Alanis[3], Ozlem Ozmen Garibay[1], Niloofar Yousefi[1]

[1]University of Central Florida, USA, [2]Anymate Me, Germany, [3]Microsoft, USA
https://github.com/Kowsher/LLMMixer

Abstract

Time series forecasting is a challenging task, especially when dealing with data that contains both short-term variations and long-term trends. In this study, we introduce LLM-Mixer, a novel framework that combines multiscale time-series decomposition with the power of pre-trained Large Language Models (LLMs). LLM-Mixer breaks down time-series data into multiple temporal resolutions using downsampling and processes these multiscale representations with a frozen LLM, guided by a carefully designed text prompt that encodes information about the dataset's features and structure. To understand the role of downsampling, we conduct a detailed analysis using Neural Tangent Kernel (NTK) distance, showing that incorporating multiple scales improves the model's learning dynamics. We evaluate LLM-Mixer across a diverse set of forecasting tasks, including long-term multivariate, short-term multivariate, and long-term univariate scenarios. Experimental results demonstrate that LLM-Mixer achieves competitive performance compared to recent state-of-the-art models across various forecasting horizons. Code is available at: https://github.com/Kowsher/LLMMixer

1 Introduction & Related Work

Time series forecasting is essential in numerous fields, including finance (Zhang et al., 2024), energy management (Martín et al., 2010), healthcare (Morid et al., 2023), climate science (Mudelsee, 2019), and industrial operations (Wang et al., 2020). Traditional forecasting models, such as AutoRegressive Integrated Moving Average (ARIMA) (Box et al., 2015) and exponential smoothing techniques (Hyndman, 2018), are widely used for straightforward predictive tasks. However, these models assume stationarity and linearity, which limit their effectiveness when applied to complex, nonlinear, and multivariate time series often found in real-world scenarios (Cheng et al., 2015). The advent of deep learning has significantly advanced time series forecasting. CNNs (Wang et al., 2023; Tang et al., 2020; Kirisci and Cagcag Yolcu, 2022) have been utilized for capturing temporal patterns, while RNNs (Siami-Namini et al., 2019; Zhang et al., 2019; Karim et al., 2019) are adept at modeling temporal state transitions. However, both CNNs and RNNs have limitations in capturing long-term dependencies (Wang et al., 2024; Tang et al., 2021; Zhu et al., 2023). Recently, Transformer architectures (Vaswani et al., 2017) have demonstrated strong capabilities in handling both local and long-range dependencies, making them suitable for time series forecasting (Liu et al., 2024b; Nie et al., 2022; Woo et al., 2022).

In parallel, pre-trained LLMs such as GPT-3 (Brown, 2020), GPT-4 (Achiam et al., 2023), and LLaMA (Touvron et al., 2023) have achieved remarkable generalization in natural language processing tasks (Friha et al., 2024) due to capabilities of few-shot or zero-shot transfer learning (Brown, 2020), multimodal knowledge (Jia et al., 2024) and reasoning (Liu et al., 2024a). These models are now being applied across various fields, including computer vision (Bendou et al., 2024), healthcare (Gebreab et al., 2024), and finance (Zhao et al., 2024). Recently, a few studies have explored using LLMs for time series forecasting due to their impressive capabilities (Jin et al., 2024, 2023; Gruver et al., 2023). However, adapting LLMs to time series data presents challenges because there are significant differences between token-based text data and continuous time series data (Morales-García et al., 2024). LLMs are built to handle discrete tokens, which limits their ability to capture the continuous and often irregular patterns found in time series data. Additionally, time series data has multiple time scales, from short-term fluctuations to long-term trends, making it difficult for traditional LLMs to capture all these patterns at once. LLMs typically process fixed-length sequences, which

Proceedings of the 4th Table Representation Learning Workshop, pages 156–165
July 31, 2025 ©2025 Association for Computational Linguistics

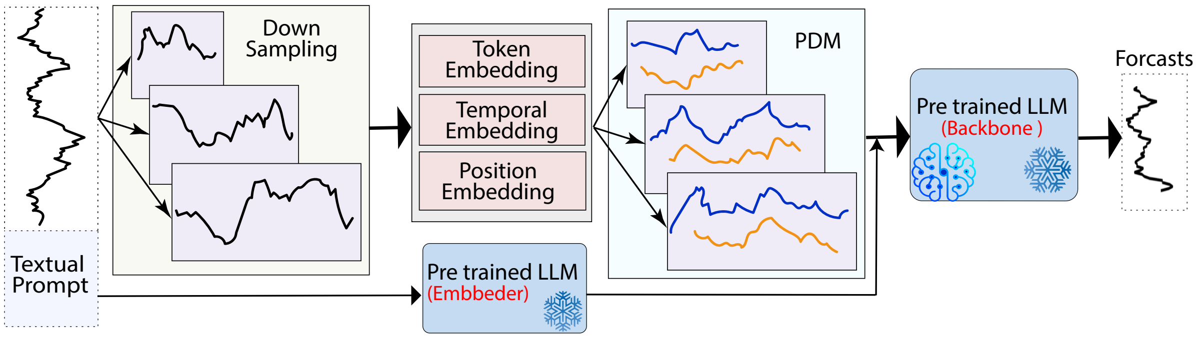


Figure 1: The LLM-Mixer framework for time series forecasting. Time series data is downsampled to multiple scales and enriched with embeddings. These multiscale representations are processed by the Past-Decomposable-Mixing (PDM) module and then input into a pre-trained LLM, which, guided by a textual description, generates the forecast.

means they may only capture short-term dependencies if the sequence length (i.e., the window of time steps) is small. However, extending the sequence length to capture long-term trends increases computational costs and may dilute the model's ability to focus on short-term fluctuations within the same sequence. Previous studies using LLMs on time series data have mostly fed the original or a single sequence directly into a frozen LLM, making it hard for the model to fully understand these sequences (Jin et al., 2024, 2023; Gruver et al., 2023).

To address this, we introduce **LLM-Mixer**, which breaks down the time series data into multiple time scales. By creating various resolutions (Figure 1), our model can capture both short-term details and long-term patterns more effectively. Since the LLM remains frozen during training, the multiscale decomposition provides a diverse range of temporal information, helping the model better understand complex time series data.

Our contributions of this paper are: **(1)** We propose **LLM-Mixer**, a new method that adapts LLMs for time series forecasting by breaking down the data into different time scales, helping the model capture both short-term and long-term patterns. **(2)** Our method creates multiple versions of the time series at different resolutions which helps the LLM to understand complex time series data more effectively. **(3)** Empirical results show that **LLM-Mixer** achieves competitive performance, improves forecasting accuracy on both multivariate and univariate data, and works effectively for both short-term and long-term forecasting tasks.

2 LLM Mixer

Preliminaries: In multivariate time series forecasting, we are given historical data $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\} \in \mathbb{R}^{T \times M}$, where T is the number of time steps and M is the number of features. The goal is to predict the future values for the next K time steps, denoted as $\mathbf{Y} = \{\mathbf{x}_{T+1}, \ldots, \mathbf{x}_{T+K}\} \in \mathbb{R}^{K \times M}$. For convenience, let $\mathbf{X}_{t,:}$ represent the data at time step t, and $\mathbf{X}_{:,m}$ represent the full time series for variable $m \in M$.

Now, suppose we have a prompt $\mathbf{P}$, which includes textual information about the time sequence (e.g., source, features, distribution, statistics). We use a pre-trained language model $\mathbb{F}(\cdot)$ with frozen parameters Θ, then the prediction is made as follows:

$$\hat{\mathbf{Y}} = \mathbb{F}(\mathbf{X}, \mathbf{P}; \Theta, \mathbf{\Phi})$$

Here $\mathbf{\Phi}$ is a small set of trainable parameters to adjust the model for the specific forecasting task.

Multi-scale View of Time Data: Time series data contains patterns at various levels—small scales capture detailed changes, while larger scales highlight overarching trends (Liu et al., 2022; Mozer, 1991). Analyzing data at multiple scales helps to understand these complex patterns (Wang et al., 2024). Following (Wang et al., 2024), we apply a multiscale mixing strategy. First, we downsample the time series $\mathbf{X}$ into τ scales using average pooling, resulting in a multiscale representation $\mathcal{X} = \{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_\tau\}$, where each $\mathbf{x}_i \in \mathbb{R}^{\frac{T}{2^i} \times M}$. Here, $\mathbf{x}_0$ contains the finest temporal details, while $\mathbf{x}_\tau$ captures the broadest trends.

Next, we project these multiscale series into deep features using three types of embeddings: token, temporal, and positional embeddings. Token embeddings are obtained via 1D convolutions (Kiranyaz et al., 2021), temporal embeddings represent day, week, and month (Jiménez-Navarro et al., 2023), and positional embeddings encode sequence positions.

We then use stacked Past-Decomposable-Mixing (PDM) blocks by following the framework from (Wang et al., 2024; Jiménez-Navarro et al., 2023) to mix past information across different scales. PDM works by breaking down complex time series data into separate seasonal and trend components at multiple scales, allowing for targeted processing of each component by using the framework from (Wang et al., 2024; Wu et al., 2021). For the l-th layer, PDM is defined as

$$\mathcal{X}^l = PDM(\mathcal{X}^{l-1}), \quad l \in L$$

where L is the total number of layers, and $\mathcal{X}^l = \{\mathbf{x}_0^l, \mathbf{x}_1^l, \ldots, \mathbf{x}_\tau^l\}$, with each $\mathbf{x}_i^l \in \mathbb{R}^{\frac{T}{2^i} \times d}$, where d is the model's dimension.

Prompt Embedding: Prompting is an effective technique for guiding LLMs by using task-specific information (Sahoo et al., 2024; Li et al., 2023). Studies like (Xue and Salim, 2023) show promising results by treating time series inputs as prompts for forecasting. (Jin et al., 2024) further improved time series predictions by embedding dataset descriptions in the prompts. Inspired by this, we embed dataset descriptions (e.g., features, statistics, distribution) as prompts. We use a textual description for all samples in a dataset, as suggested by (Jin et al., 2024), and generate its embedding using the pre-trained LLM's word embeddings, denoted by $E \in \mathbb{R}^{V \times d}$, where V is the LLM's vocabulary size. This prompt leverages the LLM's semantic knowledge to improve the prediction task.

Multi-scale Mixing in LLM: After processing through L PDM blocks, we obtain the multiscale past information $\mathcal{X}^L$. Since different scales focus on different variations, their predictions offer complementary strengths. To fully utilize this, we concatenate all the scales and input them into a frozen pre-trained LLM along with the prompt as $\mathbb{F}(E \oplus \mathcal{X}^L)$. Finally, a trainable decoder (simple linear transformation) with parameters $\mathbf{\Phi}$ is applied to the last hidden layer of the LLM to predict the next K future time steps.

3 Experiments

We evaluate our LLM-Mixer on several datasets commonly used for benchmarking long-term and short-term multivariate forecasting and compared with SOTA baselines. For long-term forecasting, we use the ETT datasets (ETTh1, ETTh2, ETTm1, ETTm2) from (Zhou et al., 2021), as well as the Weather, Electricity, and Traffic datasets from (Zeng et al., 2023). For short-term forecasting, we use the PeMS dataset (Chen et al., 2001), which consists of four public traffic network datasets (PEMS03, PEMS04, PEMS07, and PEMS08) with time series collected at various frequencies. We used RoBERTa-base (Liu et al., 2019) as a medium-sized language model and LLaMA2-7B (Touvron et al., 2023) as a large language model as the backbone of our framework.

Baselines We compare our model with well-established time-series forecasting baselines such as TimeMixer (Wang et al., 2024), iTransformer (Liu et al., 2024b), TimeLLM (Jin et al., 2024), RLinear (Li et al., 2024), SCINet (LIU et al., 2022), TimesNet (Wu et al., 2022), TiDE (Das et al., 2023), DLinear (Zeng et al., 2023), PatchTST (Nie et al., 2022), FEDformer (Zhou et al., 2022), Stationary (Liu et al., 2022), ESTformer (Woo et al., 2022), LightTS (Campos et al., 2023), and Autoformer (Chen et al., 2021). Additionally, we include LLM-based systems such as TimeLLM (Jin et al., 2024) and GPT2TS (Zhou et al., 2023). For multivariate time series forecasting, we follow the setup of (Wang et al., 2024). For short-term forecasting, we adopt the settings from (Liu et al., 2024b), and for univariate forecasting, we adhere to the approach in (Zeng et al., 2023).

Implementation Details All experiments in this work are implemented using PyTorch. We utilize the Hugging Face library for the LLM model. Experiments were conducted on an NVIDIA H100 GPU with 80 GB RAM.

Hyperparameters: For long-term experiments, a look-back window of 96 is used to predict the next 96 (future context) and 192 (forecast horizons), while short-term experiments use windows of 24 and 48. All experiments run for 10 epochs with a batch size of 64 for RoBERTa and a batch size of 8 with gradient accumulation of 4 for LLaMA2. The ADAM optimizer is employed with default settings $(\beta_1, \beta_2) = (0.9, 0.999)$ and a learning rate of 0.0001. Downsampling levels range from 2 to 5 across all experiments. For the baseline models, we have followed their original works, with differences only in batch size and learning rate to align with our experimental setup.

Multivariate forecasting results: LLM-Mixer demonstrates competitive performance in multivariate long forecasting, as shown in Table 1. Averaged over four forecasting horizons (96, 192, 384, and 720), LLM-Mixer achieves consistently low

Methods		LLM-Mixer (llama2)		LLM-Mixer (roberta)		TIME-LLM		TimeMixer		iTransformer		RLinear		DLinear		PatchTST		TimesNet		TiDE		TimesNet		Crossformer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.368**	**0.395**	0.372	0.399	0.369	0.397	0.375	0.400	0.386	0.405	0.386	0.395	0.397	0.412	0.460	0.447	0.384	0.402	0.479	0.464	0.384	0.402	0.423	0.448
	192	**0.406**	**0.417**	0.439	0.470	0.411	0.428	0.429	0.421	0.441	0.436	0.437	0.424	0.446	0.441	0.512	0.477	0.436	0.429	0.525	0.492	0.436	0.429	0.471	0.474
	336	0.446	**0.444**	0.458	0.467	**0.440**	0.447	0.484	0.458	0.487	0.458	0.479	0.446	0.489	0.467	0.546	0.496	0.638	0.469	0.565	0.515	0.491	0.469	0.570	0.546
	720	**0.461**	**0.475**	0.465	0.480	0.462	0.477	0.498	0.482	0.503	0.491	0.481	0.470	0.513	0.510	0.544	0.517	0.521	0.500	0.594	0.558	0.521	0.500	0.653	0.621
	Avg	**0.420**	**0.433**	0.434	0.454	0.421	0.437	0.447	0.440	0.454	0.447	0.446	0.434	0.461	0.457	0.516	0.484	0.495	0.450	0.541	0.507	0.458	0.450	0.529	0.522
ETTh2	96	**0.274**	**0.334**	0.284	0.347	0.278	0.338	0.289	0.341	0.297	0.349	0.288	0.338	0.340	0.394	0.308	0.355	0.340	0.374	0.400	0.440	0.340	0.374	0.745	0.584
	192	0.339	0.384	0.343	0.389	**0.338**	**0.384**	0.372	0.392	0.380	0.400	0.374	0.390	0.482	0.479	0.393	0.405	0.402	0.414	0.528	0.509	0.402	0.414	0.877	0.656
	336	0.380	**0.408**	**0.375**	0.409	0.389	0.411	0.386	0.414	0.428	0.432	0.415	0.426	0.591	0.541	0.427	0.436	0.452	0.452	0.643	0.571	0.452	0.452	1.043	0.731
	720	**0.390**	**0.431**	0.394	0.438	0.393	0.432	0.412	0.434	0.427	0.445	0.420	0.440	0.839	0.661	0.436	0.450	0.462	0.468	0.874	0.679	0.462	0.468	1.104	0.763
	Avg	**0.345**	**0.389**	0.349	0.395	0.349	0.391	0.364	0.395	0.383	0.407	0.374	0.398	0.563	0.519	0.391	0.411	0.414	0.427	0.611	0.550	0.414	0.427	0.942	0.684
ETTm1	96	0.294	0.346	0.304	0.348	**0.293**	**0.343**	0.320	0.357	0.334	0.368	0.355	0.376	0.346	0.374	0.352	0.374	0.338	0.375	0.364	0.387	0.338	0.375	0.404	0.426
	192	**0.348**	**0.367**	0.350	0.377	0.350	0.368	0.361	0.381	0.377	0.391	0.391	0.392	0.382	0.391	0.390	0.393	0.374	0.387	0.398	0.404	0.374	0.387	0.450	0.451
	336	0.387	0.392	0.395	0.409	**0.382**	**0.391**	0.390	0.404	0.426	0.420	0.424	0.415	0.415	0.415	0.421	0.414	0.410	0.411	0.428	0.425	0.410	0.411	0.532	0.515
	720	0.439	0.442	0.448	0.450	**0.443**	0.451	0.454	**0.441**	0.491	0.459	0.487	0.450	0.473	0.451	0.462	0.449	0.478	0.450	0.487	0.461	0.478	0.450	0.666	0.589
	Avg	**0.367**	**0.387**	0.374	0.396	0.367	0.388	0.381	0.395	0.407	0.410	0.414	0.407	0.404	0.408	0.406	0.407	0.400	0.406	0.419	0.419	0.400	0.406	0.513	0.495
ETTm2	96	**0.160**	**0.251**	0.160	0.253	0.160	0.251	0.175	0.252	0.180	0.264	0.182	0.265	0.193	0.293	0.183	0.270	0.187	0.267	0.207	0.305	0.187	0.267	0.287	0.366
	192	0.226	**0.290**	0.229	0.297	**0.220**	0.292	0.237	0.299	0.250	0.309	0.246	0.304	0.284	0.361	0.255	0.314	0.249	0.309	0.290	0.364	0.249	0.309	0.414	0.492
	336	**0.283**	0.339	0.299	0.346	0.284	**0.337**	0.298	0.340	0.311	0.348	0.307	0.342	0.382	0.429	0.309	0.347	0.321	0.351	0.377	0.422	0.321	0.351	0.597	0.542
	720	0.392	0.398	0.399	0.405	0.391	0.397	**0.391**	**0.396**	0.412	0.407	0.407	0.398	0.558	0.525	0.412	0.404	0.365	0.359	0.558	0.524	0.408	0.403	1.730	1.042
	Avg	0.265	0.320	0.272	0.325	**0.264**	**0.319**	0.275	0.323	0.288	0.332	0.286	0.327	0.354	0.402	0.290	0.334	0.291	0.333	0.358	0.404	0.291	0.333	0.757	0.610
Weather	96	0.149	0.202	0.151	0.203	**0.148**	**0.202**	0.163	0.209	0.174	0.214	0.192	0.232	0.195	0.252	0.186	0.227	0.172	0.220	0.202	0.261	0.172	0.220	0.195	0.271
	192	**0.197**	**0.239**	0.209	0.249	0.199	0.242	0.208	0.250	0.221	0.254	0.240	0.271	0.237	0.295	0.234	0.265	0.219	0.261	0.242	0.298	0.219	0.261	0.209	0.277
	336	0.270	0.282	0.310	0.281	0.262	**0.279**	0.251	0.287	0.278	0.296	0.292	0.307	0.282	0.331	0.284	0.301	**0.246**	0.337	0.287	0.335	0.280	0.306	0.273	0.332
	720	0.323	0.332	0.339	0.342	0.330	0.334	0.339	0.341	0.358	0.347	0.364	0.353	0.282	0.331	0.356	0.349	0.365	0.359	0.287	0.335	**0.280**	**0.306**	0.379	0.401
	Avg	**0.235**	**0.264**	0.252	0.269	0.235	0.264	0.240	0.271	0.258	0.278	0.272	0.291	0.265	0.315	0.265	0.285	0.251	0.294	0.271	0.320	0.259	0.287	0.264	0.320
Electricity	96	0.143	**0.233**	0.150	0.241	**0.142**	0.234	0.153	0.247	0.148	0.240	0.201	0.281	0.210	0.302	0.190	0.296	0.168	0.272	0.237	0.329	0.168	0.272	0.219	0.314
	192	**0.151**	0.242	0.166	0.259	0.152	**0.241**	0.166	0.256	0.162	0.253	0.201	0.283	0.210	0.305	0.199	0.304	0.184	0.322	0.236	0.330	0.184	0.289	0.231	0.322
	336	**0.178**	0.267	0.180	0.281	0.180	**0.263**	0.185	0.277	0.178	0.269	0.215	0.298	0.223	0.319	0.217	0.319	0.198	0.300	0.249	0.344	0.198	0.300	0.246	0.337
	720	**0.213**	**0.305**	0.221	0.311	0.218	0.308	0.225	0.310	0.225	0.317	0.257	0.331	0.258	0.350	0.258	0.352	0.220	0.320	0.284	0.373	0.220	0.320	0.280	0.363
	Avg	**0.171**	**0.253**	0.174	0.273	0.173	0.261	0.182	0.272	0.178	0.270	0.219	0.298	0.225	0.319	0.216	0.318	0.193	0.304	0.251	0.344	0.192	0.295	0.244	0.334
Traffic	96	**0.380**	**0.264**	0.394	0.274	0.382	0.268	0.462	0.285	0.395	0.268	0.649	0.389	0.650	0.396	0.526	0.347	0.593	0.321	0.805	0.493	0.593	0.321	0.644	0.429
	192	0.396	0.269	0.399	0.276	**0.394**	**0.267**	0.473	0.296	0.417	0.276	0.601	0.366	0.598	0.370	0.522	0.332	0.617	0.336	0.756	0.474	0.617	0.336	0.665	0.431
	336	**0.423**	**0.274**	0.439	0.280	0.425	0.281	0.498	0.296	0.433	0.283	0.609	0.369	0.605	0.373	0.517	0.334	0.629	0.336	0.762	0.477	0.629	0.336	0.674	0.420
	720	**0.458**	**0.296**	0.460	0.298	0.460	0.300	0.506	0.313	0.467	0.302	0.647	0.387	0.645	0.394	0.552	0.352	0.640	0.350	0.719	0.449	0.640	0.350	0.683	0.424
	Avg	**0.414**	**0.265**	0.433	0.282	0.415	0.279	0.484	0.297	0.428	0.282	0.626	0.378	0.625	0.383	0.529	0.341	0.620	0.336	0.760	0.473	0.620	0.336	0.667	0.426

Table 1: Full long-term multivariate forecasting results. **Red:** the best, Blue: the second best.

Methods		LLM-Mixer (llama2)		LLM-Mixer (roberta)		TIME-LLM		TimeMixer		iTransformer		RLinear		PatchTST		Crossformer		TiDE		TimesNet		DLinear		SCINet	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
PEMS03	12	0.069	0.173	0.082	0.190	0.092	0.201	0.082	0.189	0.071	0.174	0.126	0.236	0.099	0.216	0.090	0.203	0.178	0.305	0.085	0.192	0.122	0.243	**0.066**	**0.172**
	24	0.090	0.200	0.092	0.201	0.095	0.207	0.090	0.199	0.093	0.201	0.246	0.334	0.142	0.259	0.121	0.240	0.257	0.371	0.118	0.223	0.201	0.317	**0.085**	**0.198**
	48	**0.123**	**0.232**	0.126	0.237	0.127	0.237	0.125	0.235	0.125	0.236	0.551	0.529	0.211	0.319	0.202	0.317	0.379	0.463	0.155	0.260	0.333	0.425	0.127	0.238
	96	0.165	**0.274**	0.166	0.276	0.165	0.274	0.167	0.275	0.164	0.275	1.057	0.787	0.269	0.370	0.262	0.367	0.490	0.539	**0.028**	0.317	0.457	0.515	0.178	0.287
	Avg	**0.112**	**0.220**	0.116	0.226	0.120	0.230	0.116	0.225	0.113	0.221	0.495	0.472	0.180	0.291	0.169	0.281	0.326	0.419	0.147	0.248	0.278	0.375	0.114	0.224
PEMS04	12	0.072	**0.177**	0.076	0.183	**0.071**	0.177	0.073	0.179	0.078	0.183	0.138	0.252	0.105	0.224	0.098	0.209	0.182	0.324	0.087	0.195	0.148	0.272	0.073	0.177
	24	0.095	0.201	0.105	0.211	0.106	0.214	0.097	0.205	0.108	0.205	0.258	0.387	0.150	0.266	0.135	0.250	0.309	0.454	0.099	0.217	0.225	0.367	**0.084**	**0.193**
	48	**0.099**	**0.216**	0.103	0.220	0.101	0.218	0.099	0.217	0.120	0.233	0.572	0.544	0.229	0.339	0.205	0.353	0.470	0.539	0.135	0.253	0.355	0.437	0.099	0.217
	96	0.120	**0.225**	0.130	0.232	0.121	0.225	0.121	0.225	0.150	0.267	1.159	0.947	0.309	0.520	0.299	0.467	0.656	0.637	**0.043**	0.317	0.550	0.541	0.129	0.227
	Avg	**0.097**	**0.205**	0.104	0.211	0.100	0.209	0.098	0.207	0.111	0.221	0.526	0.491	0.195	0.307	0.209	0.314	0.353	0.475	0.129	0.245	0.329	0.395	0.119	0.234
PEMS07	12	**0.065**	**0.165**	0.072	0.180	0.068	0.166	0.070	0.168	0.067	0.165	0.118	0.235	0.097	0.226	0.093	0.209	0.155	0.324	0.081	0.185	0.118	0.272	0.068	0.174
	24	**0.087**	**0.105**	0.091	0.198	0.088	0.192	0.087	0.105	0.088	0.190	0.271	0.449	0.153	0.276	0.138	0.251	0.338	0.475	0.096	0.223	0.207	0.381	0.087	0.180
	48	**0.106**	**0.215**	0.117	0.224	0.109	0.219	0.106	0.217	0.110	0.215	0.596	0.621	0.343	0.459	0.309	0.401	0.532	0.547	0.145	0.264	0.593	0.484	0.149	0.233
	96	0.147	0.266	0.146	0.265	0.150	0.269	0.151	0.269	**0.141**	**0.245**	1.096	0.795	0.346	0.490	0.329	0.443	0.674	0.650	0.203	0.307	0.789	0.531	0.191	0.267
	Avg	**0.101**	**0.204**	0.107	0.217	0.104	0.212	0.104	0.218	0.101	0.204	0.504	0.478	0.213	0.303	0.215	0.326	0.355	0.499	0.129	0.245	0.529	0.387	0.119	0.234
PEMS08	12	0.082	0.186	0.086	0.190	0.080	0.184	0.083	0.183	**0.079**	**0.182**	0.133	0.247	0.168	0.232	0.152	0.267	0.215	0.367	0.154	0.276	0.172	0.291	0.087	0.184
	24	0.107	0.213	0.109	0.217	0.105	0.208	0.109	0.218	0.105	0.209	0.242	0.360	0.189	0.321	0.174	0.314	0.258	0.430	0.178	0.307	0.290	0.346	**0.104**	**0.193**
	48	0.187	0.235	0.192	0.240	0.193	0.242	0.192	0.241	0.125	0.237	0.596	0.556	0.269	0.389	0.247	0.388	0.433	0.512	0.210	0.345	0.418	0.422	**0.124**	**0.216**
	96	**0.140**	**0.245**	0.142	0.249	0.147	0.256	0.143	0.252	0.167	0.275	1.043	0.841	0.344	0.470	0.326	0.459	0.534	0.571	0.283	0.387	0.593	0.514	0.155	0.253
	Avg	0.119	0.226	0.132	0.224	0.131	0.223	0.132	0.224	0.119	0.226	0.503	0.501	0.243	0.353	0.225	0.357	0.360	0.470	0.206	0.329	0.368	0.393	**0.118**	**0.212**

Table 2: Full short-term multivariate forecasting results. **Red:** the best, Blue: the second best.

Methods		LLM-Mixer (llama2)		LLM-Mixer (Roberta)		Linear		NLinear		DLinear		FEDformer-f		FEDformer-w		Autoformer		Informer		LogTrans	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.052**	**0.175**	0.053	0.177	0.189	0.359	0.055	0.176	0.056	0.180	0.079	0.215	0.080	0.214	0.071	0.206	0.193	0.377	0.283	0.468
	192	**0.064**	**0.200**	0.066	0.203	0.078	0.212	0.069	0.204	0.071	0.204	0.104	0.245	0.105	0.256	0.114	0.262	0.217	0.395	0.234	0.409
	336	**0.080**	**0.226**	0.081	0.226	0.091	0.237	0.084	0.228	0.098	0.244	0.119	0.270	0.120	0.269	0.107	0.258	0.202	0.381	0.386	0.546
	720	**0.075**	**0.222**	0.078	0.223	0.172	0.340	0.080	0.226	0.189	0.359	0.142	0.299	0.127	0.280	0.126	0.283	0.183	0.355	0.475	0.629
	Avg	**0.068**	**0.206**	0.071	0.207	0.133	0.287	0.071	0.208	0.104	0.247	0.111	0.257	0.108	0.255	0.106	0.252	0.199	0.377	0.345	0.513
ETTh2	96	0.125	0.274	**0.123**	0.276	0.133	0.283	0.129	0.278	0.131	0.279	0.128	**0.271**	0.156	0.306	0.153	0.306	0.213	0.373	0.217	0.379
	192	**0.166**	0.322	0.169	0.324	0.176	**0.330**	0.169	0.324	0.176	0.329	0.185	0.330	0.238	0.380	0.204	0.351	0.227	0.387	0.281	0.429
	336	**0.193**	**0.353**	0.194	0.356	0.213	0.371	0.194	0.355	0.209	0.367	0.231	0.378	0.271	0.412	0.246	0.389	0.242	0.401	0.293	0.437
	720	0.222	**0.380**	0.225	0.381	0.292	0.440	0.225	0.381	0.276	0.426	0.278	0.420	0.288	0.438	0.268	0.409	0.291	0.439	**0.218**	0.387
	Avg	**0.177**	**0.332**	0.178	0.334	0.204	0.356	0.179	0.335	0.198	0.350	0.205	0.350	0.238	0.384	0.218	0.364	0.243	0.400	0.252	0.408
ETTm1	96	**0.023**	**0.118**	0.026	0.125	0.028	0.125	0.026	0.122	0.028	0.123	0.033	0.140	0.036	0.149	0.056	0.183	0.109	0.277	0.049	0.171
	192	**0.033**	**0.145**	0.036	0.147	0.043	0.154	0.039	0.149	0.045	0.156	0.058	0.186	0.069	0.206	0.081	0.216	0.151	0.310	0.157	0.317
	336	0.053	**0.172**	0.054	0.176	0.059	0.180	**0.052**	0.172	0.061	0.182	0.084	0.231	0.071	0.209	0.076	0.218	0.427	0.591	0.289	0.459
	720	**0.071**	0.205	0.072	**0.204**	0.080	0.211	0.073	0.207	0.080	0.210	0.102	0.250	0.105	0.248	0.110	0.267	0.438	0.586	0.430	0.579
	Avg	**0.045**	**0.161**	0.047	0.163	0.053	0.167	0.048	0.163	0.054	0.168	0.069	0.202	0.070	0.203	0.081	0.221	0.281	0.441	0.231	0.381
ETTm2	96	**0.062**	**0.180**	0.064	0.181	0.066	0.189	0.063	0.182	0.063	0.183	0.067	0.198	0.063	0.189	0.065	0.189	0.088	0.225	0.075	0.208
	192	0.090	0.222	**0.089**	**0.220**	0.094	0.230	0.090	0.223	0.092	0.227	0.102	0.245	0.110	0.252	0.118	0.256	0.132	0.283	0.129	0.275
	336	**0.114**	**0.255**	0.116	0.257	0.120	0.263	0.117	0.259	0.119	0.261	0.130	0.279	0.147	0.301	0.154	0.305	0.180	0.336	0.154	0.302
	720	0.169	**0.313**	0.171	0.314	0.175	0.320	0.170	0.318	0.175	0.320	0.178	0.325	0.219	0.368	0.182	0.335	0.300	0.435	**0.160**	0.321
	Avg	**0.109**	**0.243**	0.110	0.243	0.114	0.250	0.110	0.246	0.112	0.248	0.119	0.262	0.135	0.279	0.130	0.271	0.150	0.295	0.130	0.277

Table 3: Full univariate long sequence time-series forecasting results on ETT full benchmark. **Red:** the best, Blue: the second best.

MSE and MAE values across most datasets, particularly excelling on ETTh1, ETTh2, and Electricity. Compared to other models such as TIME-LLM, TimeMixer, and PatchTST, LLM-Mixer performs favorably, showing that its design effectively captures both short- and long-term dependencies. Notably, LLM-Mixer also exhibits robustness on challenging datasets such as Traffic, where it outperforms several baseline models. These results highlight the efficacy of the LLM-Mixer in handling complex temporal patterns over extended horizons.

Short-term forecasting results: In Table 2, we present the short-term multivariate forecasting results, across four forecasting horizons: 12, 24, 48, and 96 time steps. Our proposed model consistently achieves low MSE and MAE values across the PEMS datasets, indicating a strong short-term predictive performance. Specifically, LLM-Mixer demonstrates competitive accuracy on PEMS03, PEMS04, and PEMS07, outperforming several baseline models, including TIME-LLM, TimeMixer, and PatchTST. Additionally, the LLM-Mixer shows robustness on PEMS08, where it delivers superior results compared to iTransformer and DLinear. These results emphasize the effectiveness of the LLM-Mixer in capturing essential temporal dynamics for short-horizon forecasting tasks.

Univariate forecasting results: Table 3 presents the univariate long forecasting results on the ETT benchmark and averaged over horizons of 96, 192, 384, and 720-time steps. LLM-Mixer achieves the lowest MSE and MAE values across all datasets, consistently outperforming other methods like Linear, NLinear, and FEDformer. LLM-Mixer demonstrates superior accuracy, particularly on most of the datasets. These results confirm the effectiveness of the LLM-Mixer in capturing complex temporal dependencies, solidifying its capability for univariate long-term forecasting.

3.1 Ablation Study

Effect of Downsampling on Learning Dynamicse: To evaluate the impact of different downsampling levels on the learning dynamics of LLM-Mixer, we conducted an ablation study using the Neural Tangent Kernel (NTK) (Jacot et al., 2018). Specifically, we aimed to understand how the number of downsampling levels affects the model's ability to capture multiscale information. First, we used DeepEcho (Patki et al., 2016) to generate synthetic multivariate time series datasets for this study. We trained 10 versions of LLM-Mixer, each with a different number of downsampling levels $\tau \in \{1, 2, \ldots, 10\}$. For each model, we calculated the NTK on 300 sample pairs from both the training and test sets. The NTK, denoted as $\mathbf{K}(\mathbf{x}, \mathbf{x}')$, is computed as the inner product of the gradients of the model outputs with respect to its parameters:

$$\mathbf{K}(\mathbf{x}, \mathbf{x}') = \nabla_{\boldsymbol{\theta}} \theta_t(\mathbf{x}; \boldsymbol{\theta})^\top \nabla_{\boldsymbol{\theta}} \theta_t(\mathbf{x}'; \boldsymbol{\theta}),$$

where $\nabla_{\boldsymbol{\theta}} \theta_t(\mathbf{x}; \boldsymbol{\theta})$ is the gradient of the model output with respect to its parameters at iteration t.

Data Leakage Prevention Protocol: To ensure fair comparison and avoid data leakage, we construct prompts using only metadata and statistics computed exclusively from the training set. Specifically, we include: (1) dataset description (e.g., "electricity consumption data"), (2) feature names and units, (3) basic statistics (mean, standard deviation, data frequency) computed only from training samples. No information from validation or test sets is incorporated into the prompt construction process. We validate this approach through ablation studies comparing models with and without statistical information in prompts.

To measure how the NTK structure changes with different 10 levels, we used the Frobenius norm to calculate the distance between the NTK of each model ($\mathbf{K}_\tau$) and a reference NTK ($\mathbf{K}_{10}$), which corresponds to the model with the maximum downsampling levels. The NTK distance is defined as:

$$d_{\text{NTK}}(\tau) = \|\mathbf{K}_{10} - \mathbf{K}_\tau\|_F,$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Smaller NTK distances indicate that the model's learning dynamics are closer to the reference model.

Our results, shown in Figure 2, reveal that as the number of downsampling levels τ decreases, the NTK distance increases. The largest distance is observed when $\tau = 1$, indicating that using only one downsampling level significantly alters the model's learning dynamics.However, more downsampling levels are not always better. While increasing τ enhances the model's ability to capture multiscale patterns, excessive downsampling may smooth out critical fine-grained details, which are essential for tasks with significant short-term variations. In Figure 3, we visualize the NTK of the reference model across different downsampling levels τ and the normalized absolute differences.

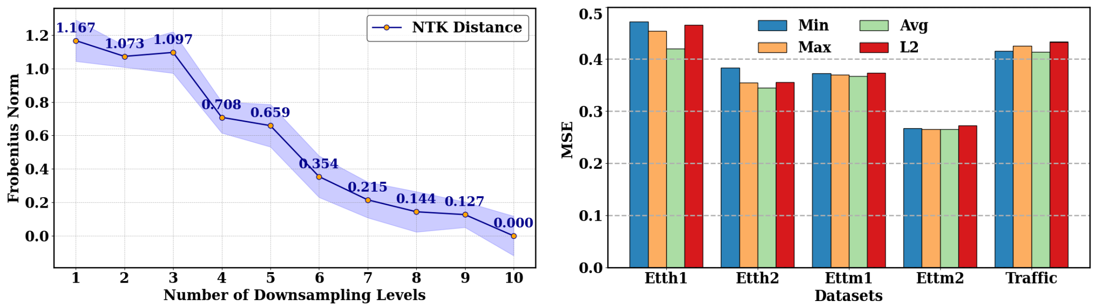
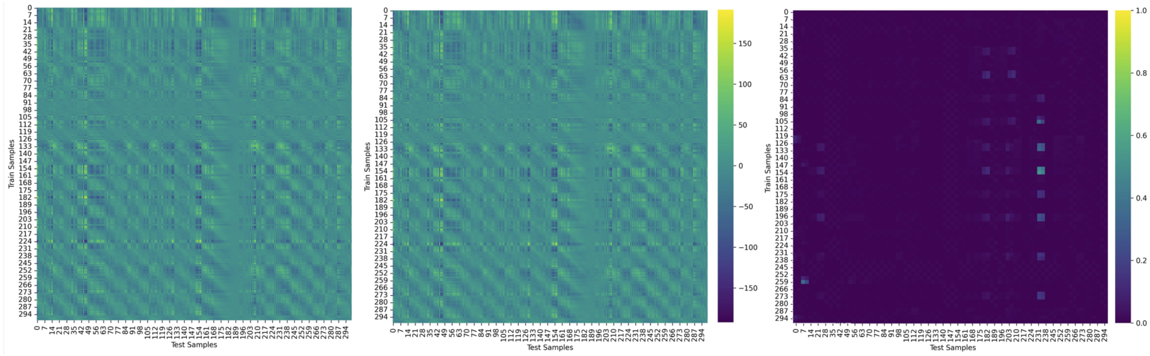
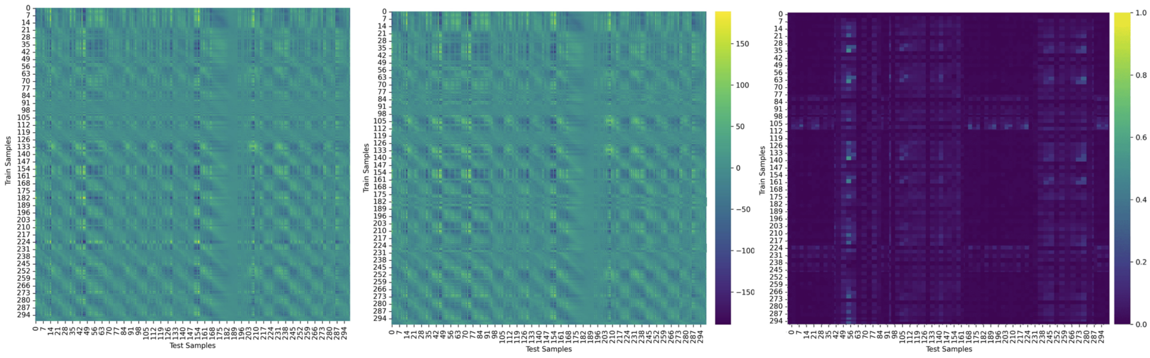


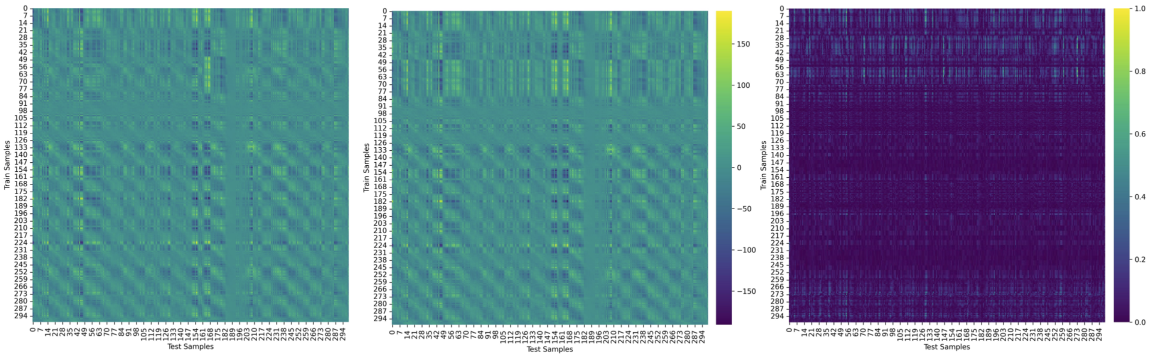
Figure 2: (**Left**) Frobenius norm of NTK distance. (**Right**) Pooling technique for Multi-scale Mixing



(a) $\tau = 9$



(b) $\tau = 7$



(c) $\tau = 5$

Figure 3: Visualization of (a) $\tau = 9$, (b) $\tau = 7$, and (c) $\tau = 5$. Each subfigure displays the reference NTK at $\tau = 10$, the NTK at the respective τ level, and their absolute difference.

Multi-scale Mixing by Pooling: We conducted an ablation study to explore the effects of various Multi-scale Mixing techniques. The techniques examined were Min, Max, Avg, and L2, each applying a unique method for aggregating downsampling information across scales. Figure 2 (right) presents the MSE for each downsampling method across different datasets. Notably, average pooling consistently yielded a lower MSE, suggesting that this method is better suited for capturing multi-scale dependencies in the data.

4 Conclusion

This work introduces the LLM-Mixer, a novel framework that combines multiscale time-series decomposition with pre-trained LLMs for improved forecasting. By leveraging multiple temporal resolutions, the LLM-Mixer effectively captures both short-and long-term patterns, enhancing the model's predictive accuracy. Our experiments demonstrate that the LLM-Mixer achieves competitive performance across various datasets, outperforming recent state-of-the-art methods.

5 Limitations and Future Directions

Although LLM-Mixer improves forecasting accuracy, several limitations warrant discussion.

Computational Requirements: The use of pre-trained language models introduces significant computational overhead, which may limit deployment in real-time or resource-constrained environments. **Prompt Engineering:** Model performance depends on prompt quality and domain expertise for optimal prompt design, which may limit accessibility for non-experts.

Out-of-Distribution Robustness: When training and test data distributions differ significantly, the fixed prompt approach may not adapt effectively to distributional shifts.

Limited Classical Baseline Analysis: Our evaluation focuses primarily on deep learning methods and would benefit from comprehensive comparison with statistical approaches like ARIMA and exponential smoothing.

Data Leakage Potential: While we implement protocols to prevent information leakage, the prompt-based approach requires careful validation to ensure fair comparison.

Domain Generalization: Testing on more diverse domains (finance, healthcare, climate) would strengthen claims about broad applicability. Future work should address these limitations through adaptive prompting strategies, efficiency optimizations, and expanded empirical validation.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. arXiv preprint arXiv:2303.08774.

Yassir Bendou, Giulia Lioi, Bastien Pasdeloup, Lukas Mauch, Ghouthi Boukli Hacene, Fabien Cardinaux, and Vincent Gripon. 2024. Llm meets vision-language models for zero-shot one-class classification. arXiv preprint arXiv:2404.00675.

George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. 2015. Time series analysis: forecasting and control. John Wiley & Sons.

Tom B Brown. 2020. Language models are few-shot learners. arXiv preprint arXiv:2005.14165.

David Campos, Miao Zhang, Bin Yang, Tung Kieu, Chenjuan Guo, and Christian S Jensen. 2023. Lightts: Lightweight time series classification with adaptive ensemble distillation. Proceedings of the ACM on Management of Data, 1(2):1–27.

Chao Chen, Karl Petty, Alexander Skabardonis, Pravin Varaiya, and Zhanfeng Jia. 2001. Freeway performance measurement system: mining loop detector data. Transportation research record, 1748(1):96–102.

Minghao Chen, Houwen Peng, Jianlong Fu, and Haibin Ling. 2021. Autoformer: Searching transformers for visual recognition. In Proceedings of the IEEE/CVF international conference on computer vision, pages 12270–12280.

Changqing Cheng, Akkarapol Sa-Ngasoongsong, Omer Beyca, Trung Le, Hui Yang, Zhenyu Kong, and Satish TS Bukkapatnam. 2015. Time series forecasting for nonlinear and non-stationary processes: a review and comparative study. Iie Transactions, 47(10):1053–1071.

Abhimanyu Das, Weihao Kong, Andrew Leach, Shaan K Mathur, Rajat Sen, and Rose Yu. 2023. Long-term forecasting with tiDE: Time-series dense encoder. Transactions on Machine Learning Research.

Othmane Friha, Mohamed Amine Ferrag, Burak Kantarci, Burak Cakmak, Arda Ozgun, and Nassira Ghoualmi-Zine. 2024. Llm-based edge intelligence: A comprehensive survey on architectures, applications, security and trustworthiness. IEEE Open Journal of the Communications Society.

Senay A Gebreab, Khaled Salah, Raja Jayaraman, Muhammad Habib ur Rehman, and Samer Ellaham. 2024. Llm-based framework for administrative task automation in healthcare. In 2024 12th International Symposium on Digital Forensics and Security (ISDFS), pages 1–7. IEEE.

Nate Gruver, Marc Anton Finzi, Shikai Qiu, and Andrew Gordon Wilson. 2023. Large language models are zero-shot time series forecasters. In Thirty-seventh Conference on Neural Information Processing Systems.

RJ Hyndman. 2018. Forecasting: principles and practice. OTexts.

Arthur Jacot, Franck Gabriel, and Clément Hongler. 2018. Neural tangent kernel: Convergence and generalization in neural networks. Advances in neural information processing systems, 31.

Bonian Jia, Huiyao Chen, Yueheng Sun, Meishan Zhang, and Min Zhang. 2024. Llm-driven multimodal opinion expression identification. arXiv preprint arXiv:2406.18088.

Manuel Jesús Jiménez-Navarro, María Martínez-Ballesteros, Francisco Martínez-Álvarez, and Gualberto Asencio-Cortés. 2023. Embedded temporal feature selection for time series forecasting using deep learning. In International Work-Conference on Artificial Neural Networks, pages 15–26. Springer.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. 2024. Time-LLM: Time series forecasting by reprogramming large language models. In The Twelfth International Conference on Learning Representations.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. 2023. Time-llm: Time series forecasting by reprogramming large language models. arXiv preprint arXiv:2310.01728.

Fazle Karim, Somshubra Majumdar, and Houshang Darabi. 2019. Insights into lstm fully convolutional networks for time series classification. Ieee Access, 7:67718–67725.

Serkan Kiranyaz, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, and Daniel J Inman. 2021. 1d convolutional neural networks and applications: A survey. Mechanical systems and signal processing, 151:107398.

Melih Kirisci and Ozge Cagcag Yolcu. 2022. A new cnn-based model for financial time series: Taiex and ftse stocks forecasting. Neural Processing Letters, 54(4):3357–3374.

Lei Li, Yongfeng Zhang, and Li Chen. 2023. Prompt distillation for efficient llm-based recommendation. In Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, pages 1348–1357.

Zhe Li, Shiyi Qi, Yiduo Li, and Zenglin Xu. 2024. Revisiting long-term time series forecasting: An investigation on affine mapping.

Jingyu Liu, Jiaen Lin, and Yong Liu. 2024a. How much can rag help the reasoning of llm? arXiv preprint arXiv:2410.02338.

Keqin Liu, Teng Zhang, Bingjie Dang, Lin Bao, Liying Xu, Caidie Cheng, Zhen Yang, Ru Huang, and Yuchao Yang. 2022. An optoelectronic synapse based on α-in2se3 with controllable temporal dynamics for multimode and multiscale reservoir computing. Nature Electronics, 5(11):761–773.

Minhao LIU, Ailing Zeng, Muxi Chen, Zhijian Xu, Qiuxia LAI, Lingna Ma, and Qiang Xu. 2022. SCINet: Time series modeling and forecasting with sample convolution and interaction. In Advances in Neural Information Processing Systems.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. 2024b. itransformer: Inverted transformers are effective for time series forecasting. In The Twelfth International Conference on Learning Representations.

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. 2022. Non-stationary transformers: Exploring the stationarity in time series forecasting. Advances in Neural Information Processing Systems, 35:9881–9893.

Luis Martín, Luis F Zarzalejo, Jesus Polo, Ana Navarro, Ruth Marchante, and Marco Cony. 2010. Prediction of global solar irradiance based on time series analysis: Application to solar thermal power plants energy production planning. Solar Energy, 84(10):1772–1781.

Juan Morales-García, Antonio Llanes, Francisco Arcas-Túnez, and Fernando Terroso-Sáenz. 2024. Developing time series forecasting models with generative large language models. ACM Transactions on Intelligent Systems and Technology.

Mohammad Amin Morid, Olivia R Liu Sheng, and Joseph Dunbar. 2023. Time series prediction using deep learning methods in healthcare. ACM Transactions on Management Information Systems, 14(1):1–29.

Michael C Mozer. 1991. Induction of multiscale temporal structure. Advances in neural information processing systems, 4.

Manfred Mudelsee. 2019. Trend analysis of climate time series: A review of methods. Earth-science reviews, 190:310–322.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2022. A time series is worth 64 words: Long-term forecasting with transformers. arXiv preprint arXiv:2211.14730.

Neha Patki, Roy Wedge, and Kalyan Veeramachaneni. 2016. The synthetic data vault. In 2016 IEEE international conference on data science and advanced analytics (DSAA), pages 399–410. IEEE.

Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. 2024. A systematic survey of prompt engineering in large language models: Techniques and applications. arXiv preprint arXiv:2402.07927.

Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. 2019. The performance of lstm and bilstm in forecasting time series. In 2019 IEEE International conference on big data (Big Data), pages 3285–3292. IEEE.

Wensi Tang, Guodong Long, Lu Liu, Tianyi Zhou, Jing Jiang, and Michael Blumenstein. 2020. Rethinking 1d-cnn for time series classification: A stronger baseline. arXiv preprint arXiv:2002.10061, pages 1–7.

Yuqing Tang, Fusheng Yu, Witold Pedrycz, Xiyang Yang, Jiayin Wang, and Shihu Liu. 2021. Building trend fuzzy granulation-based lstm recurrent neural network for long-term time-series forecasting. IEEE transactions on fuzzy systems, 30(6):1599–1613.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. Advances in neural information processing systems, 30.

Huiqiang Wang, Jian Peng, Feihu Huang, Jince Wang, Junhui Chen, and Yifei Xiao. 2023. MICN: Multi-scale local and global context modeling for long-term series forecasting. In The Eleventh International Conference on Learning Representations.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y Zhang, and Jun Zhou. 2024. Timemixer: Decomposable multiscale mixing for time series forecasting. arXiv preprint arXiv:2405.14616.

Yongjian Wang, Ke Yang, and Hongguang Li. 2020. Industrial time-series modeling via adapted receptive field temporal convolution networks integrating regularly updated multi-region operations based on pca. Chemical Engineering Science, 228:115956.

Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. 2022. Etsformer: Exponential smoothing transformers for time-series forecasting. arXiv preprint arXiv:2202.01381.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. 2022. Timesnet: Temporal 2d-variation modeling for general time series analysis. In The eleventh international conference on learning representations.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. Advances in neural information processing systems, 34:22419–22430.

Hao Xue and Flora D Salim. 2023. Promptcast: A new prompt-based learning paradigm for time series forecasting. IEEE Transactions on Knowledge and Data Engineering.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2023. Are transformers effective for time series forecasting? In Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence, pages 11121–11128.

Cheng Zhang, Nilam Nur Amir Sjarif, and Roslina Ibrahim. 2024. Deep learning models for price forecasting of financial time series: A review of recent advancements: 2020–2022. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 14(1):e1519.

Xuan Zhang, Xun Liang, Aakas Zhiyuli, Shusen Zhang, Rui Xu, and Bo Wu. 2019. At-lstm: An attention-based lstm model for financial time series prediction. In IOP Conference Series: Materials Science and Engineering, page 052037. IOP Publishing.

Huaqin Zhao, Zhengliang Liu, Zihao Wu, Yiwei Li, Tianze Yang, Peng Shu, Shaochen Xu, Haixing Dai, Lin Zhao, Gengchen Mai, et al. 2024. Revolutionizing finance with llms: An overview of applications and insights. arXiv preprint arXiv:2401.11641.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In 35th AAAI Conference on Artificial Intelligence, AAAI 2021, pages 11106–11115. Association for the Advancement of Artificial Intelligence.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. 2022. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In International conference on machine learning, pages 27268–27286. PMLR.

Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. 2023. One fits all: Power general time series analysis by pretrained lm. Advances in neural information processing systems, 36:43322–43355.

Chenglong Zhu, Xueling Ma, Weiping Ding, and Jianming Zhan. 2023. Long-term time series forecasting with multi-linear trend fuzzy information granules for lstm in a periodic framework. IEEE Transactions on Fuzzy Systems.