TIMERAG: BOOSTING LLM TIME SERIES FORECASTING VIA RETRIEVAL-AUGMENTED GENERATION

Silin Yang[1,*]   *Dong Wang*[2,*]   *Haoqi Zheng*[2,†]   *Ruochun Jin*[2,†]

[1]School of Computer Science, Peking University, Beijing, China
[2]College of Computer, National University of Defense Technology, Changsha, China

ABSTRACT

Although the rise of large language models (LLMs) has introduced new opportunities for time series forecasting, existing LLM-based solutions require excessive training and exhibit limited transferability. In view of these challenges, we propose TimeRAG, a framework that incorporates Retrieval-Augmented Generation (RAG) into time series forecasting LLMs, which constructs a time series knowledge base from historical sequences, retrieves reference sequences from the knowledge base that exhibit similar patterns to the query sequence measured by Dynamic Time Warping (DTW), and combines these reference sequences and the prediction query as a textual prompt to the time series forecasting LLM. Experiments on datasets from various domains show that the integration of RAG improved the prediction accuracy of the original model by 2.97% on average.

Index Terms—Time Series Forecasting, Large Language Model(LLM), Retrieval-Augmented Generation(RAG), Dynamic Time Warping (DTW)

I. INTRODUCTION

Time series forecasting is critical in data science and machine learning research, covering wide applications including financial market analysis, demand forecasting, weather prediction, etc. Although deep-model-based forecasting methods such as LSTM [1], Reformer [2] and Informer [3] have achieved satisfactory performance on classical benchmarks [4], they can hardly capture the hidden complex patterns and dependencies in large-scale sequential data with staggering complexity and diversity [5]. In view of this challenge, researchers have explored the possibility of applying large language models (LLMs) to time series analysis and prediction across various domains [6], [7], since LLMs have demonstrated remarkable achievements in natural language processing. However, existing time series forecasting LLMs cannot easily adapt to different domains, as the training of LLMs is computationally costly and typically optimized for a specific domain [8]. Moreover, due to the "hallucination" of LLMs [9], LLMs may generate inaccurate predictions, outliers, or fabricated patterns when performing time series

forecasting, that do not align with the data, with no interpretability.

In order to resolve these issues, we propose to boost time series forecasting LLMs via Retrieval-Augmented Generation (RAG) [10]. Specifically, we first establish a time series knowledge base by collecting representative sequential data from the training set via K-means clustering. Then given the time series forecasting query as input, we employ Dynamic Time Warping (DTW) [11] as the distance metric to retrieve sequences, that share similar waveforms and trends with the query, from the time series knowledge base as referential sequences, since DTW is tolerant to temporal distortions. Finally, the input query and the referential sequences are rewritten as a natural language prompt and fed into the LLMs for prediction. We have experimentally verified our method on M4 datasets, a collection of varying-frequencies time series from different domains [12], where significant improvements of up to **13.12%** have been observed.

Different from existing time series LLMs that require massive training costs [13], [14] and previous RAG solutions [10], [15], to the best of our knowledge, we are the first to propose a RAG framework specifically designed for time series data prediction without modifying the foundational parameters of the underlying LLM. Experimental results confirm that our method exhibits strong competitiveness when compared to both similar LLMs and baseline models.

The key contributions of our work are as follows:

- To the best of our knowledge, we are the first to boost time series forecasting LLMs by Retrieval-Augmented Generation, which significantly improves prediction accuracy.
- We employ K-means clustering and Dynamic Time Warping to efficiently construct a time series knowledge base, which facilitates the LLM to easily adapt to different domains of time series.
- We experimentally verify that RAG contributes to an average improvement of **2.97%** in the accuracy of sequence forecasting.

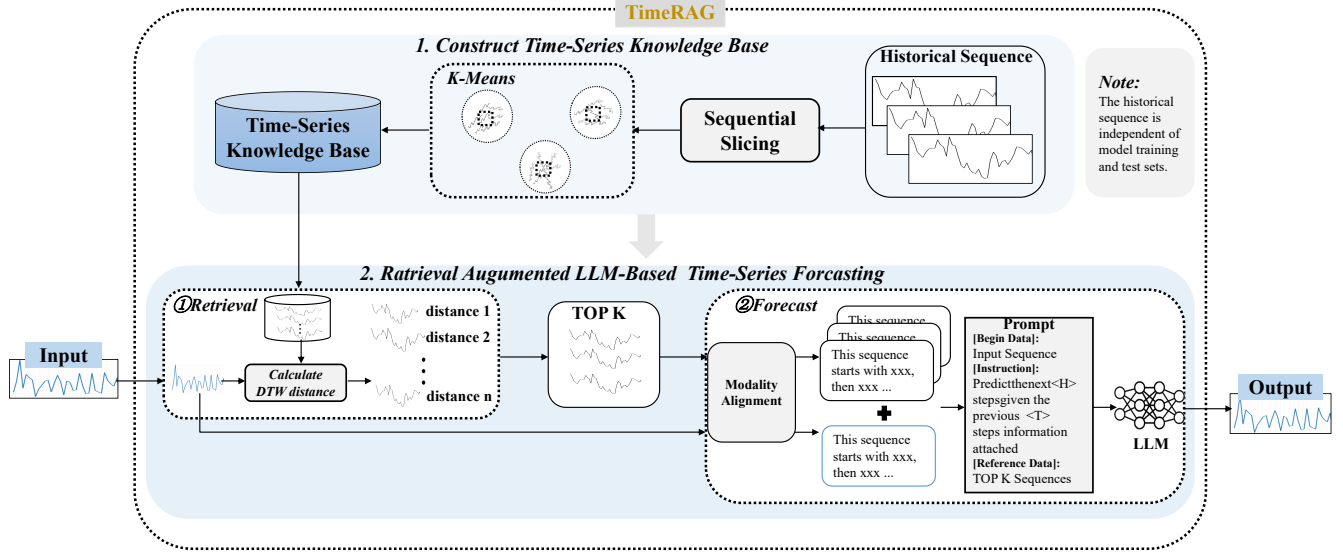*Equal Contribution
†Corresponding Authors

Fig. 1. Overview of our mechanism.

II. METHOD

II-A. Overview

As shown in Fig. 1, in order to enhance the performance of LLMs in time-series forecasting tasks, we propose a retrieval-augmented framework, named TimeRAG, that consists of two main components: a Time Series Knowledge Base (KB) (II-B) and an LLM-based time series forecasting model(II-C). Specifically, TimeRAG first sequentially slices the original sequence into segments and establishes a time series knowledge base by extracting representative segments from the training set using K-means clustering. Then input the time series forecasting query, we apply Dynamic Time Warping (DTW) as the distance metric to retrieve sequences from the time series knowledge base that exhibit similar waveforms and trends to the query, leveraging DTW's ability to handle temporal distortions. The input query and the retrieved sequences are then reformulated into a natural language prompt, which is subsequently input into the LLMs for prediction.

II-B. Time-Series Knowledge Base

In order to establish the Time-Series Knowledge Base, TimeRAG first performs sequential slicing on the original sequence through a sliding window, and employs clustering to select representative segments for storage. Instead of storing and retrieving complete raw sequences, our sequence-segmentation approach preserves the local information of the sequence, avoids long sequences where LLMs tend to miss key information [16], and improves the retrieval efficiency. Specifically, given a sequence $X = (x_t, ...x_{t+n}), X \in R^n$ of time-varying values from time t to $t + n$, TimeRAG first adopts a sliding window approach with a step size

of S and a window length of L to slice X into several sub-sequences X_L, where $X_L \in R^L$. Secondly, K-means clustering is applied to these fragments for capturing representative sequences. Given a set of N sequence fragments $Q_L = \{X_{L_i}\}, i \in [1, N]$, K-means first initializes a set of cluster centroids $C = \{X_{c_1}, ..., X_{c_k}\}, X_{c_i} \in Q_L, i \in [1, k]$ and assign each X_{L_i} to the closest centroid, where the distances between each sequence fragment X_{L_i} and all centroids are measured by the Euclidean distance as Eq.1:

$$d = \|X_{L_i} - X_{c_j}\|_2 \tag{1}$$

where $X_{L_i} \neq X_{c_j}$. After this initialization, K-means iteratively updates each centroid as the mean of sequences within each cluster and reassigns each sequence to the cluster whose centroid is the closest to the sequence, which gradually minimizes the total sum of distances between all points and their corresponding cluster centroids. Finally, TimeRAG constructs the Time-series Knowledge Base by collecting the sequence that is the closest to its centroid from each cluster.

II-C. Retrieval-Augmented LLM-based Time-series Forecasting

Although LLMs have demonstrated remarkable performance in time series forecasting [17], their prediction accuracy deteriorates when processing sequences that have not been previously trained. Moreover, LLMs show general performance degradation due to their tendency to forget [16], which may adversely affect the accuracy of time series forecasting. In view of these challenges, we introduce the retrieval-augmented LLM-based time-series forecasting that consists of the following two stages: (1) retrieval of similar sequences based on DTW [18], and (2) prediction by LLM

where both the original sequence and the retrieved similar sequences are combined to enhance forecast accuracy.

In the retrieval stage, given the prediction query and the Time-Series Knowledge Base, TimeRAG employs DTW to retrieve top-K sequences that are most similar to the query from the knowledge base. Specifically, given the input query sequence X_{input}, $X_{input} \in R^n$ for prediction, DTW first constructs an $n \times L$ matrix for each sequence X_L in the knowledge base, where the element (i,j) of the matrix represents the distance $d(i,j)$ between the X_{input_i} and X_{L_j}, which represent the i-th point of X_{input} and the j-th point of X_L respectively. The distance $d(i,j)$ is computed as following formula:

$$d(i,j) = (X_{input_i}, X_{L_j})^2 \qquad (2)$$

We refer to the path W from matrix element $(1,1)$ to (n,L), consisting of several adjacent and non-repeating matrix elements, as the warping path, where the m-th element of W is defined as $w_m = d(m_i, m_j)$, which is computed by Eq. 2. Thus, W can be given by:

$$W = w_1, ... w_m, ... w_M \qquad (3)$$

where $max(n, L) \leq M \leq n + L$ and $w_M = d(n, L)$.

The algorithm then employs dynamic programming to obtain the shortest warping path, which can be utilized to measure the similarity between X_{input} and X_L as Eq.4:

$$Simi(X_{input}, X_L) = min \frac{\sqrt{\sum_{m=1}^{M} w_m}}{M} \qquad (4)$$

where $Simi(X_{input}, X_L)$ denotes the similarity between the X_{input} and X_L. Finally, TimeRAG selects the top K sequences that are most similar to the query sequence as the retrieval results, measured by $Simi$.

In the model prediction stage, TimeRAG follows Time-LLM [13] that adopts a reprogramming layer to align the sequence modality with the natural language modality. As shown in Fig. 1, the input query sequence X_{input} and the retrieved sequences are transformed through the reprogramming layer and concatenated as one prompt, which enhances the prediction performance of the LLM.

III. EXPERIMENTS

III-A. Experiments Setup

Datasets. We evaluate TimeRAG on the M4 benchmark, a widely used dataset for time series forecasting that contains data from diverse domains, including finance, demographics, marketing, etc., with different sequential sampling frequencies: yearly, quarterly, monthly, weekly, daily, and hourly. Each frequency corresponds to specific prediction horizons and input lengths, which supports the comprehensive evaluation of forecasting models. More details of the dataset are provided in Tab. I.

Evaluation Metric. As introduced in [13], we adopt the following three widely accepted metrics for performance

Table I. The summary of the M4 dataset and the knowledge base. The number of sequences in the knowledge base is larger than the total quantity since one original sample can be sliced into multiple sequence segments.

Time Interval	Input Length	Prediction Length	Total Quantity	KB Size
Yearly	12	6	23000	57400
Quarterly	16	8	24000	50000
Monthly	36	18	48000	60000
Weekly	26	13	359	895
Daily	28	14	4227	10565
Hourly	96	48	414	1035
Total	-	-	100000	179895

evaluation: (1) Symmetric Mean Absolute Percentage Error (SMAPE): as a widely recognized measure in time series forecasting, SMAPE quantifies forecast accuracy relative to actuals by computing the percentage error. (2) Mean Absolute Scaled Error (MASE): this metric evaluates a model's predictive accuracy relative to a naive forecast strategy, offering scale independence and robustness across a series of varying magnitudes. (3) Overall Weighted Average (OWA): drawing from the methodology in N-BEATS [19], OWA integrates SMAPE and MASE to provide a holistic assessment of model performance. The smaller values of prediction results measured by SMAPE, MASE, and OWA, the higher prediction accuracy the model achieves.

Baselines. We compare TimeRAG with state-of-the-art time series models, including Transformer-based methods: iTransformer [20], FEDformer [21], Pyraformer [22], Autoformer [23], Informer [3], and Reformer [2]; as well as other competitive models: Time-LLM [13], DLinear [24], TSMixer [25], MICN [26], FiLM [27] and LightTS [28].

Training Settings. Inspired by [13], TimeRAG employs the reprogramming technique where the input time series are reprogrammed with text prototypes before fed into a frozen LLM to align the two modalities. In order to obtain a well-trained reprogramming layer, we trained TimeRAG based on Llama3 with a maximum of 50 training epochs, using 8 A100 GPUs, Adam optimizer, and SMAPE as the loss function. To mitigate over-fitting, we employ dynamic learning rate adjustment and an early stopping strategy, with the maximum learning rate set as 0.01.

Knowledge Base Implementation. As sequential data shows distinct waveform characteristics at different time frequencies, we built separate knowledge bases for each frequency in the M4 dataset and split the remaining data into training, validation, and test sets. Tab. I shows statistics of the knowledge bases for the M4 dataset at different frequencies. Once the knowledge bases were constructed, TimeRAG enhanced the test samples by retrieving the **top-five** most relevant entries, measured by DTW, from the corresponding knowledge base for each test case, which

Table II. Forecasting results on M4 dataset. Blue means the model is top-three at the current frequency and metric.

Frequency	Yearly			Quarterly			Monthly			Weekly			Daily			Hourly			Average		
Metric	SMAPE	MASE	OWA	SMAPE	MASE	OWA	SMAPE	MASE	OWA	SMAPE	MASE	OWA	SMAPE	MASE	OWA	SMAPE	MASE	OWA	SMAPE	MASE	OWA
Autoformer	20.900	3.741	1.138	14.913	1.745	1.324	18.023	1.715	1.331	16.758	4.445	1.509	4.851	4.756	1.469	26.874	8.847	2.001	17.053	4.208	1.462
DLinear	15.913	2.755	0.873	10.817	1.203	0.949	13.448	1.191	0.957	13.714	4.000	1.294	4.065	3.883	1.215	17.085	3.847	1.081	12.507	2.813	1.062
FEDformer	15.347	2.703	0.829	11.223	1.292	0.989	14.129	1.273	1.014	11.824	3.367	1.102	3.442	3.259	1.024	18.504	4.968	1.217	12.412	2.810	1.029
FiLM	15.850	2.724	0.846	11.054	1.291	0.981	13.436	1.190	0.956	12.385	3.778	1.195	3.637	3.536	1.096	22.328	9.654	1.985	13.115	3.696	1.177
Informer	16.717	2.919	0.899	15.664	1.911	1.421	18.750	1.810	1.395	14.307	4.191	1.353	5.066	4.820	1.511	36.071	21.378	4.026	17.763	6.172	1.768
iTransformer	19.762	3.506	1.071	11.659	1.330	1.022	16.653	1.547	1.215	15.860	3.930	1.383	4.040	3.763	1.192	19.195	5.513	1.316	14.528	3.265	1.200
LightTS	15.238	2.680	0.817	10.868	1.228	0.948	13.203	1.148	0.931	14.189	3.970	1.311	3.743	3.515	1.109	23.929	8.118	1.813	13.528	3.443	1.155
MICN	17.380	3.638	0.967	12.004	1.399	1.064	13.808	1.270	1.002	13.435	3.972	1.276	4.578	4.312	1.359	26.455	9.419	2.070	14.610	4.002	1.290
Pyraformer	17.981	3.160	0.970	12.234	1.415	1.080	15.263	1.387	1.101	12.507	3.732	1.194	3.819	3.629	1.138	30.359	15.852	3.084	15.361	4.863	1.428
Reformer	23.033	4.201	1.266	16.329	1.987	1.479	16.271	1.514	1.188	13.100	3.816	1.235	5.023	4.784	1.499	37.464	22.170	4.173	18.537	6.412	1.807
TSMixer	15.264	2.676	0.823	11.173	1.260	0.974	13.450	1.181	0.953	18.168	7.521	2.075	4.310	4.065	1.280	24.953	10.738	2.214	14.553	4.574	1.387
Time-LLM	15.318	2.679	0.825	10.942	1.229	0.952	13.628	1.227	0.978	14.968	4.330	1.406	3.578	3.355	1.059	18.164	4.307	1.117	12.766	2.855	1.056
TimeRAG(**Ours**)	15.317	2.678	0.825	10.941	1.230	0.952	13.529	1.141	0.940	14.227	3.762	1.279	3.572	3.348	1.057	18.150	4.151	1.095	12.623	2.718	1.025

enables retrieval-augmented time series forecasting.

III-B. Main Results

Tab. II presents a comprehensive comparison of forecasting accuracy across various models on the M4 dataset. The table is meticulously organized to display the performance metrics for different temporal granularities, including yearly, quarterly, monthly, weekly, daily, and hourly. The efficacy of each model is quantified by these three metrics: SMAPE, MASE and OWA.

TimeRAG is superior to the time series prediction LLM without RAG (Time-LLM). In our comparative analysis, our model outperforms Time-LLM, achieving an average reduction of **1.13%** in SMAPE, **4.78%** in MASE, and a notable **3.00%** decrease in OWA. Moreover, an overall improvement of **2.97%** has been observed, highlighting the model's enhanced predictive capabilities. Under optimal conditions, TimeRAG notably reduces SMAPE by **0.74** at the "Weekly" frequency and demonstrates the most significant enhancement of **13.12%** in MASE at the same interval.

These improvements across the board can be credited to the augmented knowledge base that our model incorporates. This supplementary data acts as a catalyst for the model's knowledge enhancement, effectively bolstering its predictive accuracy without modifying the foundational parameters of the underlying LLM.

TimeRAG also stands out among the current SOTA time series forecasting models, achieving the best values for both MASE and OWA metrics under the current training paradigm. On average, TimeRAG achieves a MASE score of **2.72**, which is the lowest among all evaluated models, underscoring its superior forecasting accuracy. FEDformer follows as the second, while DLinear ranks the third. Likewise, TimeRAG performs exceptionally well in the OWA metric, achieving an OWA score of **1.03**. It secures the leading position among all evaluated models, followed by FEDformer in second place and Time-LLM in third.

The empirical evidence garnered from our analysis substantiates the efficacy of integrating large models with RAG techniques for the execution of time series tasks. This amalgamation has significantly improved predictive accuracy and model responsiveness to temporal data patterns.

TimeRAG exhibits a consistent performance across all metrics and all temporal frequencies. In all 18 comparative analyses, our model secures the **top-three** positions more frequently than any other, with a total of 14 instances, consistently ranking within the **top-three** in terms of average performance. Time-LLM follows in second place, with DLinear coming in third. Our model's ability to sustain such a high level of performance across various analytical dimensions underscores its robustness and reliability in the domain of time series forecasting.

The superior performance of TimeRAG can be attributed to its highly effective alignment with the knowledge base. TimeRAG meticulously matches each time series with pertinent samples from the knowledge base, enabling the model to draw insights beyond the parameters established through training. This approach allows the LLMs to learn from a more authentic and reliable dataset, thereby minimizing randomness and improving the model's consistency.

IV. CONCLUSION

After integrating Retrieval-Augmented Generation that consists of clustering-based Time-Series Knowledge Base construction, Dynamic-Time-Warping-based similar reference sequence retrieval, and natural-language-alignment-based prompt rewriting, our TimeRAG framework significantly enhances the prediction accuracy of time series forecasting LLMs, achieving an average accuracy improvement of 2.97% over baseline models across diverse domains. Our work demonstrates the potential of RAG in amplifying LLM performance in time series forecasting, which offers a promising approach for future research in knowledge-enhanced sequential data management.

REFERENCES

[1] A. Sagheer and M. Kotb, "Time series forecasting of petroleum production using deep lstm recurrent networks," *Neurocomputing*, vol. 323, pp. 203–213, 2019.

[2] N. Kitaev, Ł. Kaiser, and A. Levskaya, "Reformer: The efficient transformer," *arXiv preprint arXiv:2001.04451*, 2020.

[3] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 12, 2021, pp. 11 106–11 115.

[4] B. Lim and S. Zohren, "Time-series forecasting with deep learning: a survey," *Philosophical Transactions of the Royal Society A*, vol. 379, no. 2194, p. 20200209, 2021.

[5] J. Ye, W. Zhang, K. Yi, Y. Yu, Z. Li, J. Li, and F. Tsung, "A survey of time series foundation models: Generalizing time series representation with large language mode," *arXiv preprint arXiv:2405.02358*, 2024.

[6] H. Xue and F. D. Salim, "Promptcast: A new prompt-based learning paradigm for time series forecasting," *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[7] T. Zhou, P. Niu, L. Sun, R. Jin *et al.*, "One fits all: Power general time series analysis by pretrained lm," *Advances in neural information processing systems*, vol. 36, pp. 43 322–43 355, 2023.

[8] Y. Jiang, Z. Pan, X. Zhang, S. Garg, A. Schneider, Y. Nevmyvaka, and D. Song, "Empowering time series analysis with large language models: A survey," *arXiv preprint arXiv:2402.03182*, 2024.

[9] Y. Bang, S. Cahyawijaya, N. Lee, W. Dai, D. Su, B. Wilie, H. Lovenia, Z. Ji, T. Yu, W. Chung *et al.*, "A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity," *arXiv preprint arXiv:2302.04023*, 2023.

[10] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 9459–9474. [Online]. Available:

[11] M. Müller, "Dynamic time warping," *Information retrieval for music and motion*, pp. 69–84, 2007.

[12] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The m4 competition: 100,000 time series and 61 forecasting methods," *International Journal of Forecasting*, vol. 36, no. 1, pp. 54–74, 2020.

[13] M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y.-F. Li, S. Pan, and Q. Wen, "Time-LLM: Time Series Forecasting by Reprogramming Large Language Models," Jan. 2024, arXiv:2310.01728 [cs]. [Online]. Available: http://arxiv.org/abs/2310.01728

[14] A. F. Ansari, L. Stella, C. Turkmen, X. Zhang, P. Mercado, H. Shen, O. Shchur, S. S. Rangapuram, S. P. Arango, S. Kapoor, J. Zschiegner, D. C. Maddix, H. Wang, M. W. Mahoney, K. Torkkola, A. G. Wilson, M. Bohlke-Schneider, and Y. Wang, "Chronos: Learning the Language of Time Series," May 2024, arXiv:2403.07815 [cs]. [Online]. Available: http://arxiv.org/abs/2403.07815

[15] Y. Liu, S. Yavuz, R. Meng, D. Radev, C. Xiong, and Y. Zhou, "Uni-parser: Unified semantic parser for question answering on knowledge base and database," *arXiv preprint arXiv:2211.05165*, 2022.

[16] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang, "Lost in the middle: How language models use long contexts," *Transactions of the Association for Computational Linguistics*, vol. 12, pp. 157–173, 2024. [Online]. Available: https://aclanthology.org/2024.tacl-1.9

[17] H. Liu, Z. Zhao, J. Wang, H. Kamarthi, and B. A. Prakash, "LSTPrompt: Large language models as zero-shot time series forecasters by long-short-term prompting," in *Findings of the Association for Computational Linguistics ACL 2024*, L.-W. Ku, A. Martins, and V. Srikumar, Eds. Bangkok, Thailand and virtual meeting: Association for Computational Linguistics, Aug. 2024, pp. 7832–7840. [Online]. Available: https://aclanthology.org/2024.findings-acl.466

[18] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE transactions on acoustics, speech, and signal processing*, vol. 26, no. 1, pp. 43–49, 1978.

[19] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "N-beats: Neural basis expansion analysis for interpretable time series forecasting," *arXiv preprint arXiv:1905.10437*, 2019.

[20] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long, "itransformer: Inverted transformers are effective for time series forecasting," *arXiv preprint arXiv:2310.06625*, 2023.

[21] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting," in *International conference on machine learning*. PMLR, 2022, pp. 27 268–27 286.

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

[23] M. Chen, H. Peng, J. Fu, and H. Ling, "Autoformer: Searching transformers for visual recognition," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 12 270–12 280.

[24] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 9, 2023, pp. 11 121–11 128.

[25] V. Ekambaram, A. Jati, N. Nguyen, P. Sinthong, and J. Kalagnanam, "Tsmixer: Lightweight mlp-mixer model for multivariate time series forecasting," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 459–469.

[26] H. Wang, J. Peng, F. Huang, J. Wang, J. Chen, and Y. Xiao, "Micn: Multi-scale local and global context modeling for long-term series forecasting," in *The eleventh international conference on learning representations*, 2023.

[27] T. Zhou, Z. Ma, Q. Wen, L. Sun, T. Yao, W. Yin, R. Jin *et al.*, "Film: Frequency improved legendre memory model for long-term time series forecasting," *Advances in neural information processing systems*, vol. 35, pp. 12 677–12 690, 2022.

[28] D. Campos, M. Zhang, B. Yang, T. Kieu, C. Guo, and C. S. Jensen, "Lightts: Lightweight time series classification with adaptive ensemble distillation," *Proceedings of the ACM on Management of Data*, vol. 1, no. 2, pp. 1–27, 2023.