2024

Efficient High-Resolution Time Series Classification via Attention Kronecker Decomposition

Aosong Feng
Yale University

Jialin Chen
Yale University

Juan Garza
The University of Texas Rio Grande Valley

Brooklyn Berry
The University of Texas Rio Grande Valley

Francisco Salazar
The University of Texas Rio Grande Valley

See next page for additional authors

Efficient High-Resolution Time Series Classification via Attention Kronecker Decomposition

Aosong Feng * Jialin Chen * Juan Garza † Brooklyn Berry † Francisco Salazar †
Yifeng Gao † Rex Ying * Leandros Tassiulas *

Abstract

The high-resolution time series classification problem is essential due to the increasing availability of detailed temporal data in various domains. To tackle this challenge effectively, it is imperative that the state-of-the-art attention model is scalable to accommodate the growing sequence lengths typically encountered in high-resolution time series data, while also demonstrating robustness in handling the inherent noise prevalent in such datasets. To address this, we propose to hierarchically encode the long time series into multiple levels based on the interaction ranges. By capturing relationships at different levels, we can build more robust, expressive, and efficient models that are capable of capturing both short-term fluctuations and long-term trends in the data. We then propose a new time series transformer backbone (**KronTime**) by introducing Kronecker-decomposed attention to process such multi-level time series, which sequentially calculates attention from the lower level to the upper level. Experiments on four long time series datasets demonstrate superior classification results with improved efficiency compared to baseline methods.

1 Introduction

Multivariate Time Series Classification (MTSC) problem is one of the most essential time series data mining tasks that have a great impact in various fields such as manufacturing[8, 25], astronomy[36, 24, 4], and entomology[39, 1, 2]. With the advancement of sensor technique, high-resolution and long-term passive monitoring time series[26, 32] has become increasingly available.

However, unlike traditional MTSC problems, the considerably long time series gathered can pose significant challenges in the classification task. The increasing time series length can impact the difficulty of the MTSC problem in two folds. First, the large length significantly increases the computation cost in a wide range of classification models such as Transformer based model, similarity comparison-based model[14], shapelet-based model[16], and bag-of-patterns based approach[31]. Furthermore, such issues are further amplified in the era of parameter-intensive deep learning models. For example, a Transformer[37] which needs memory and computation cost quadratic to the length will be difficult to use in long sequence MTSC tasks. Second, unlike image and video, the time series can potentially become highly noisy in high-resolution and passive monitoring applications, as the sensor will be sensitive and a large amount of noise and distortion may be included. Such a low Signal-to-Noise ratio across the entire time series requires the classification model to have the capability to accurately capture useful information across a long range of time and map the data into low dimensions without intervening by the presence of noises. Therefore, an effective deep learning model that can model and accurately model data behavior and dependency across long-range is necessary for addressing long-sequence MTSC tasks.

In this paper, to overcome the these challenges, we propose to hierarchically encode time series by considering multi-level interactions. As shown in Figure 1(a), each level of time series encodings defines time step correlations with a certain distance, gradually covering short-range to long-range interactions. Such hierarchical encoding alleviates the problem of short-range noisy patterns by including upper-level global information for high-resolution time series processing. Besides, the effective time series length is much less than the total length of the original time series and therefore avoids the quadratic computation costs. To accommodate the hierarchical encoding of the time series in the transformer model, we propose to decompose the original attention matrix using Kroncker decomposition according to the defined hierarchy and name the resulting model **KronTime**. Given finite attention window size and memory budget, KronTime captures correlations along different dimensions and therefore includes hierarchical multi-hop token interactions in the original sequence at multiple scales. Experiments on long-sequence time se-

*Yale University
†The University of Texas Rio Grande Valley

ries classification show superior classification results of KronTime compared to state-of-the-art attention and convolution-based models, with improved running time and memory usage.

2 Related Work

Time Series Classification. Traditional machine learning techniques like dynamic time warping (DTW) [14], hierarchical vote collective of transformation-based ensembles (HIVE-COTE) [18], and proximity forest [22] have long been prevalent in time series classification. However, they often fall short in terms of both efficiency and accuracy, particularly as datasets grow in size and complexity. To address the challenges of multivariate time series classification (MTSC), the field has witnessed a surge in the development of sophisticated deep learning models. Convolution-based architectures such as ResNet [35], InceptionTime [13], and TimesNet [38] have showcased their ability to capture local temporal patterns with high accuracy. Meanwhile, recurrent-based models [11, 23] excel in modeling sequential information but struggle with longer sequences. Transformer-based approaches [27, 19, 40] recently have emerged as powerful tools for capturing complex long-range dependencies and cross-variate interactions, due to their effective utilization of the self-attention mechanism [34]. Additionally, multi-resolution strategies [6, 5, 29] continue to be instrumental in enhancing the performance of these advanced models. Nonetheless, challenges persist, including issues of model efficiency and the handling of hierarchical data structures, especially when handling long time series, highlighting the need for further research and innovation in the field.

Efficient Transformer for Time Series. Transformers, originally prominent in natural language processing (NLP) [37, 15, 33], computer vision (CV) [21, 10], and speech recognition [7, 9], have recently attracted attention in addressing multivariate time series challenges. Several Transformer-based models have emerged, promising improved performance and enhanced model efficiency. For instance, LogTrans [17] introduces LogSparse attention, while Informer [42] proposes ProbSparse self-attention, both achieving logarithmic complexity concerning the time series length. Pyraformer [20] incorporates a pyramidal attention module to efficiently capture temporal dependencies, while FEDformer [43] leverages sparse representation in the frequency domain, presenting a frequency-enhanced Transformer with linear complexity. However, these advancements in self-attention efficiency may come at the cost of reduced expressiveness, leading to sub-optimal modeling performance. Additionally, these models often lack the ability to discern hierarchical data structures and capture multi-resolution information, both critical aspects in comprehensive time series analysis. Further research is warranted to address these limitations effectively.

3 Methodology

Long Time series often exhibit complex structures and dependencies that span multiple scales and granularities. By encoding the data hierarchically, we can represent these relationships more structured and interpretably. Kronecker decomposition, in particular, allows us to decompose the time series into a series of hierarchical tensors, where each level captures increasingly abstract representations of the underlying patterns. This hierarchical approach enables the model to learn representations at different levels of temporal granularity, from fine-grained details to higher-level trends and patterns.

3.1 Hierarchical Time Series Encoding Given input time series with n time steps, the transformer attention layer take inputs $\mathbf{x} \in \mathbb{R}^{n\times d}$, query $\mathbf{q}$, key $\mathbf{k}$ and value $\mathbf{v}$ which are derived from linear projections as $\mathbf{q} = \mathbf{x}\mathbf{W}_\mathbf{q}, \mathbf{k} = \mathbf{x}\mathbf{W}_\mathbf{k}, \mathbf{v} = \mathbf{x}\mathbf{W}_\mathbf{v}$. The attention process can then be written as

$$\mathbf{A} = \text{softmax}\left(\frac{\mathbf{q}\mathbf{k}^\intercal}{\sqrt{d}}\right), \mathbf{o} = \mathbf{A}\mathbf{v} \tag{3.1}$$

where softmax denotes the row-wise softmax normalization, $\mathbf{o}$ is the updated value vector as output.

To capture the time series at different levels and avoid the quadratic computation costs in the long sequence scenario, we propose to reshape the original sequence into a compact tensor as shown in Figure 1(a), with each dimension encoding the information of a certain level. Specifically, we reshape $\mathbf{q}, \mathbf{k}, \mathbf{v}$ into order-m tensors $\boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{K}}, \boldsymbol{\mathcal{V}} \in \mathbb{R}^{n_1\times\ldots n_m}$ $\prod_{i=1}^{m} n_i = n$, representing m-level decomposition. We then consider the attention interactions only at the same level, in other words, to update the i-th dimension of value tensor $\boldsymbol{\mathcal{V}}$, we calculate the attention between the i-th dimension of $\boldsymbol{\mathcal{Q}}$ and $\boldsymbol{\mathcal{K}}$ while treating other dimensions as batch dimensions. The update of $\boldsymbol{\mathcal{V}}$ can then be achieved by sequential updating from the first to the last level (irrelevant to the order of updating).

3.2 Attention Decomposition To model the tensor interaction between $\boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{K}}, \boldsymbol{\mathcal{V}}$, we propose attention Kronecker decomposition which sequentially models the interactions from the short range to the long range. Since the input length along each dimension is much smaller than the entire sequence, such decomposed attention can work with much less context window budget.

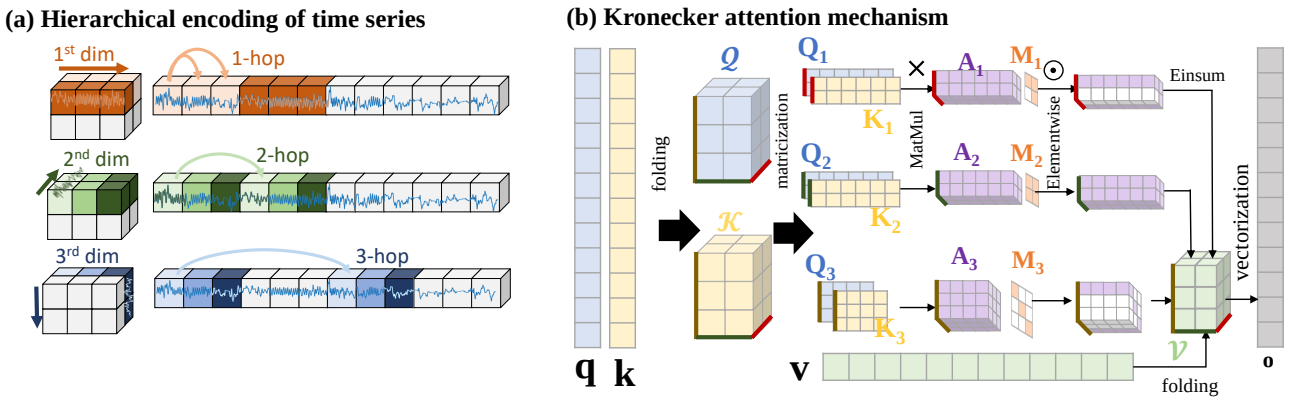

Figure 1: (a) The long time series after patchified can be decomposed into multiple levels. The first, second, and third level encodes adjacent, mid-range, and long-range global information, respectively (b) Input sequences $\mathbf{q}, \mathbf{k}, \mathbf{v}$ are first tensorized into $\mathcal{Q}, \mathcal{K}, \mathcal{V}$. Each row in the middle represents the attention along one matching dimension of tensors, and all dimensions except the matching dimension of $\mathcal{Q}$ and $\mathcal{K}$ are flattened. The result from each row is used to sequentially update the value tensor $\mathcal{V}$.

The attention process in Equation 3.1 can be decomposed with Kronecker decomposition as

$$\mathbf{A} = \otimes_{i=1}^{m} \mathbf{A}_i, \quad \mathbf{O} = \mathbf{A}\mathbf{V}, \tag{3.2}$$

where $\mathbf{A}_i$ models the i-th level token interaction. Considering the property of mixed Kronecker-matrix product, the above value tensor updating can be sequential, irrespective of the updating order. For efficient implementation, we adopt such sequential calculation of Equation 3.3 and model the i-th value updating as matrix multiplication

$$\mathbf{O}_i = \mathbf{A}_i \mathbf{V}_i \tag{3.3}$$

where $\mathbf{V}_i \in \mathbb{R}^{n_i \times (n_1..n_{i-1}n_{i+1}..n_m)}$ is the mode-i flattening of tensor $\mathcal{V}$, and $\mathbf{O}_i$ will be used as the value matrix in the next update. Mode-i flattening reshapes a tensor $\mathcal{T} \in \mathbb{R}^{n_1 \times .. \times n_m}$ into matrix $\mathbf{T}_i \in \mathbb{R}^{(n_1..n_{i-1}n_{i+1}..n_m) \times n_i}$ which can be interpreted as batching $n_1..n_{i-1}n_{i+1}..n_m$ vectors.

The attention matrix $\mathbf{A}_i$ is used to model the i-th level correlation between query and key

$$\mathbf{A}_i = \text{softmax}\left(\frac{\mathbf{Q}_i \mathbf{K}_i^\intercal}{\sqrt{d}}\right), \tag{3.4}$$

where $\mathbf{Q}_i, \mathbf{K}_i$ is the mode-i flattening of the hierarchical time series encoding tensors $\mathcal{Q}, \mathcal{K}$.

The sequential attention update is visualized in Figure 1 (b). For the i-th dimension update, $\mathcal{Q}, \mathcal{K}$ are matricized by mode-i flattening. The resulting batched attention matrix is then used to update the i-th dimension of the value tensor. The computational complexity is decreased from $\mathcal{O}(n^2)$ with full attention to $\mathcal{O}(nlogn)$. We summarize the forward pass of Kronecker-decomposed attention in Algorithm 1.

Algorithm 1 KronTime forward

Input: $\mathcal{Q}, \mathcal{K}, \mathcal{V} \in \mathbb{R}^{n_1 \times .. \times n_m}$
Initialize $\mathcal{O} = \mathcal{V}$.
for $i = 0$ **to** $m - 1$ **do**
  Mode-i flattening $\mathcal{Q}, \mathcal{K}$ into $\mathbf{Q}_i, \mathbf{K}_i$
  $\mathbf{A}_i = \text{softmax}(\mathbf{Q}_i \mathbf{K}_i^\intercal / \sqrt{d} \circ \mathbf{M}_i)$
  Mode-i flattening $\mathcal{O}$ into $\mathbf{O}_i$
  Value updates: $\mathbf{O}_i = \mathbf{A}_i \mathbf{O}_i$
  Mode-i folding $\mathbf{O}_i$ into $\mathcal{O}$
end for
Vectorize $\mathcal{O}$ to $\mathbf{o}$
Return $\mathbf{o}$

3.3 Model Implementations We choose SOTA time series transformers PatchTST [28] as the backbone, and replace the full attention in the attention layer with the Kronecker decomposed attention as discussed above. For classification applications, we add an additional classification head with linear projections which takes the last hidden states as inputs.

4 Experiment Evaluation

4.1 Baselines The proposed method will be compared with the following four baselines.

1-NN (ED)[30]: 1-NN (Euclidean Distance) is a simple yet effective algorithm for classification tasks, comparing each test instance to all training instances based on Euclidean distance.

ResNet18[12]: ResNet18 is a widely used convolu-

Table 1: Time Series Classification Performance Comparison

Model	BinaryHeartbeat	EigenWorms	FaultDetectionA	CatsDogs
1-NN (ED)	0.585	0.500	0.460	0.420
ResNet18	0.630	0.420	0.990	0.570
RandomShapelet	0.585	0.692	N/A	0.606
DLinear	0.658	0.423	0.532	0.516
TCN	**0.707**	**0.769**	0.986	0.575
PatchTST	**0.707**	0.692	0.991	0.810
KronTime	**0.707**	**0.769**	**0.996**	**0.844**

tional neural network architecture known for its depth and skip connections, which facilitate the effective learning of complex features in image data.

RandomShapelet[16]: RandomShapelet is a model leveraging shapelet transform for time series classification, extracting discriminative sub-series (shapelets) from the data to distinguish between different classes.

DLinear [41]: DLinear proposes to utilize a single linear layer that aggregates information from all history records. DLinear avoids temporal information loss caused by the nature of the permutation-invariant self-attention mechanism in transformer-based architectures and demonstrates promising performance on time series analysis.

Temporal Convolutional Network [3]: Temporal Convolutional Network (TCN) is a type of neural network specifically designed for processing temporal sequences, utilizing convolutional layers with dilated kernels to capture long-range dependencies and temporal patterns efficiently.

PatchTST: PatchTST proposed segmentation of time series into subseries-level patches as input tokens to Transformer.

4.2 Benchmark Dataset From the UEA Time Series Classification Archive, we use the following fine-resolution time series datasets that each have lengths greater than 10,000 and have a sufficient number of samples. As a result, we use the following datasets to evaluate the proposed method:

BinaryHeartbeat: BinaryHeartbeat dataset comprises heart sound recordings collected from healthy individuals and patients with cardiac conditions. The dataset contains 409 instances, recorded at 2,000Hz, with a time series length of 18,530. The classes include 110 normal and 299 abnormal instances.

EigenWorms: EigenWorms dataset focuses on Caenorhabditis elegans, a roundworm used extensively in genetics research as a model organism. The dataset collected the worm's movements represented by combinations of six base shapes, known as eigenworms. The dataset consists of 257 samples. Each samples consists of 17,891 in length. The goal is to classify individual worms as either wild-type (N2 reference strain) or one of four mutant types: goa-1, unc-1, unc-38, and unc-63.

FaultDetectionA: FaultDetection dataset contains 13,640 time series. Each one responding to the signal recorded from rolling bearing monitor system. There are total three types of classes. Undamaged rolling bearing system, inner damaged rolling bearing system, and outer damaged rolling bearing system. Each record consists of 5,120 in length.

CatsDogs: CatsDogs dataset consists of total 277 samples sound records from cats and dogs. Each recording consists of 14,773 in length

4.3 Experiment Setting For each dataset, we uses 80% of data as training data, 10% of data as validation data, and the rest 10% of data as testing data. The performance is evaluated on the accuracy in the testing data as well as the efficiency of KronTime. We run experiments with the same train/validation/test split on the four datasets. The final test accuracy is obtained using the checkpoint of the lowest validation loss, with the early-stop patience epochs 20. All experiments are performed on A6000 GPUs.

4.4 Classification Performance Evaluation The classification result is shown in Table 1. KronTime achieves the same or superior classification accuracy compared to SOTA models. Notably, KronTime achieves such performance with improved efficiency as shown in Figure 2. To make fair comparisons, we apply FlashAttention-2 to PatchTST to replace its original slow PyTorch implementation for memory-efficient calculations. Results show that KronTime achieves around 0.3× running time compared to PatchTST at length 16k and is comparable to the conventional convolution-

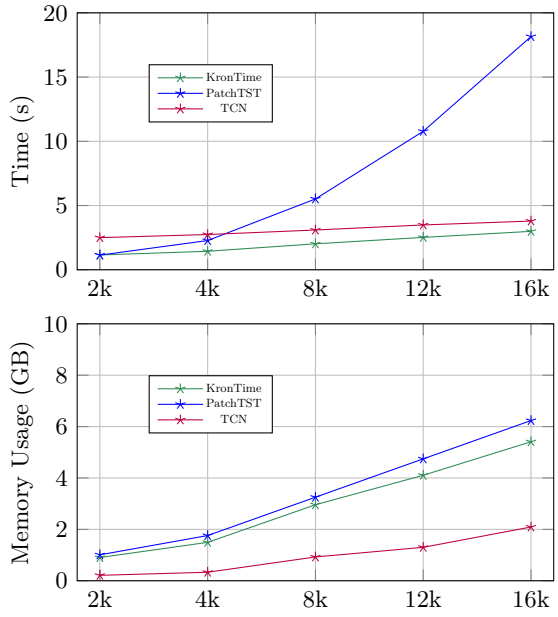

Figure 2: Comparison of running time and GPU memory usage with different input lengths.

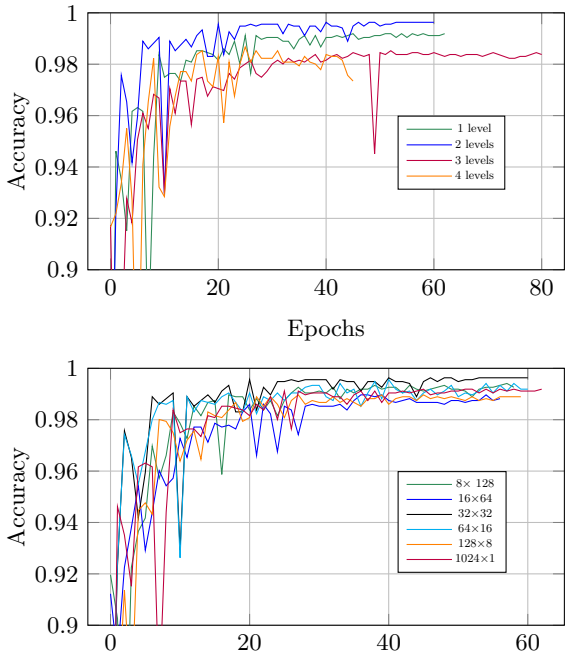

Figure 3: The validation accuracy with different Kronecker decomposition strategies (upper: number of levels decomposed; lower: different decomposition with 2 levels) during the training phase.

based TCN. Such advantage of improved running time from using attention decomposition is larger as input length grows. Besides, the memory usage of PatchTST and KronTime stays relatively constant as input length grows, because of FlashAttention-2 and hierarchical decomposition, respectively.

4.5 Parameter Testing We next demonstrate the influence of the Kronecker decomposition by comparing the training curves (with training stop tolerance 20 epochs) under different decomposition strategies in KronTime. We perform such ablation studies on FaultDetectionA dataset with 1024 input length after tokenized. We change the total number of levels decomposed and the size of each level while keeping other model and training hyperparameters unchanged for fair comparisons. As shown in Figure 3(a), the training with 2-level decomposition (16×16) converges to the higher validation accuracy with faster speed, compared to no decomposition (1-level decomposition), 3-level decomposition ($16 \times 16 \times 4$), and 4-level decomposition ($16 \times 4 \times 4 \times 4$). This indicates that for FaultDetectionA dataset with 1024 length, 2-level Kronecker decomposition leads to the optimal result. We then compare different decomposition strategies with 2 levels, and results in Figure 3(b) show that decompositing the 1024 sequence into 32×32 achieves superior results compared to other decompositions for this dataset.

5 Conclusion

The high-resolution time series classification problem is essential due to the increasing availability of high-fidelity time series data. The growth of such high-resolution data will pose challenges in designing classification models. To tackle this challenge, we proposed Kronecker-decomposed attention (KronTime) to extract features from time series over 10,000 lengths effectively. The experiment demonstrates that KronTime can achieve superior classification results with improved efficiency compared to baselines.

References

[1] Mehenika Akter, Mohammad Shahadat Hossain, Tawsin Uddin Ahmed, and Karl Andersson. Mosquito classification using convolutional neural network with data augmentation. In *International Conference on Intelligent Computing & Optimization*, pages 865–879. Springer, 2020.

[2] Hernan S Alar and Proceso L Fernandez. Accurate and efficient mosquito genus classification algorithm using candidate-elimination and nearest centroid on extracted features of wingbeat acoustic properties. *Computers in Biology and Medicine*, 139:104973, 2021.

[3] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

[4] Saksham Bassi, Kaushal Sharma, and Atharva Gomekar. Classification of variable stars light curves using long short term memory network. *Frontiers in Astronomy and Space Sciences*, 8:718139, 2021.

[5] Wei Chen and Ke Shi. Multi-scale attention convolutional neural network for time series classification. *Neural Networks*, 136:126–140, 2021.

[6] Zhicheng Cui, Wenlin Chen, and Yixin Chen. Multi-scale convolutional neural networks for time series classification. *arXiv preprint arXiv:1603.06995*, 2016.

[7] Linhao Dong, Shuang Xu, and Bo Xu. Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5884–5888. IEEE, 2018.

[8] Shu-Kai S Fan, Chia-Yu Hsu, Chih-Hung Jen, Kuan-Lung Chen, and Li-Ting Juan. Defective wafer detection using a denoising autoencoder for semiconductor manufacturing processes. *Advanced Engineering Informatics*, 46:101166, 2020.

[9] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*, 2020.

[10] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):87–110, 2022.

[11] Michael Hüsken and Peter Stagge. Recurrent neural networks for time series classification. *Neurocomputing*, 50:223–235, 2003.

[12] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data mining and knowledge discovery*, 33(4):917–963, 2019.

[13] Hassan Ismail Fawaz, Benjamin Lucas, Germain Forestier, Charlotte Pelletier, Daniel F Schmidt, Jonathan Weber, Geoffrey I Webb, Lhassane Idoumghar, Pierre-Alain Muller, and François Petitjean. Inceptiontime: Finding alexnet for time series classification. *Data Mining and Knowledge Discovery*, 34(6):1936–1962, 2020.

[14] Young-Seon Jeong, Myong K Jeong, and Olufemi A Omitaomu. Weighted dynamic time warping for time series classification. *Pattern recognition*, 44(9):2231–2240, 2011.

[15] Katikapalli Subramanyam Kalyan, Ajit Rajasekharan, and Sivanesan Sangeetha. Ammus: A survey of transformer-based pretrained models in natural language processing. *arXiv preprint arXiv:2108.05542*, 2021.

[16] Isak Karlsson, Panagiotis Papapetrou, and Henrik Boström. Generalized random shapelet forests. *Data mining and knowledge discovery*, 30:1053–1085, 2016.

[17] Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *Advances in neural information processing systems*, 32, 2019.

[18] Jason Lines, Sarah Taylor, and Anthony Bagnall. Hive-cote: The hierarchical vote collective of transformation-based ensembles for time series classification. In *2016 IEEE 16th international conference on data mining (ICDM)*, pages 1041–1046. IEEE, 2016.

[19] Minghao Liu, Shengqi Ren, Siyuan Ma, Jiahui Jiao, Yizhou Chen, Zhiguang Wang, and Wei Song. Gated transformer networks for multivariate time series classification. *arXiv preprint arXiv:2103.14438*, 2021.

[20] Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International conference on learning representations*, 2021.

[21] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[22] Benjamin Lucas, Ahmed Shifaz, Charlotte Pelletier, Lachlan O'Neill, Nayyar Zaidi, Bart Goethals, François Petitjean, and Geoffrey I Webb. Proximity forest: an effective and scalable distance-based classifier for time series. *Data Mining and Knowledge Discovery*, 33(3):607–635, 2019.

[23] Pankaj Malhotra, Vishnu TV, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. Timenet: Pre-trained deep recurrent neural network for time series classification. *arXiv preprint arXiv:1706.08838*, 2017.

[24] Gal Matijevič, Andrej Prša, Jerome A Orosz, William F Welsh, Steven Bloemen, and Thomas Barclay. Kepler eclipsing binary stars. iii. classification of kepler eclipsing binary light curves with locally linear embedding. *The Astronomical Journal*, 143(5):123, 2012.

[25] Samia MELLAH, Youssef TRARDI, Guillaume GRATON, Bouchra ANANOU, EL El Mostafa, and Mustapha OULADSINE. Early semiconductor anomaly detection based on multivariate time-series classification using multilayer perceptron. *IFAC-PapersOnLine*, 55(10):3082–3087, 2022.

[26] Oliver C Metcalf, Jos Barlow, Stuart Marsden, Nargila Gomes de Moura, Erika Berenguer, Joice Ferreira, and Alexander C Lees. Optimizing tropical forest bird surveys using passive acoustic monitoring and high temporal resolution sampling. *Remote Sensing in Ecology and Conservation*, 8(1):45–56, 2022.

[27] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and

Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

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

[29] Bin Qian, Yong Xiao, Zhenjing Zheng, Mi Zhou, Wanqing Zhuang, Sen Li, and Qianli Ma. Dynamic multi-scale convolutional neural network for time series classification. *IEEE access*, 8:109732–109746, 2020.

[30] Alejandro Pasos Ruiz, Michael Flynn, James Large, Matthew Middlehurst, and Anthony Bagnall. The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, 35(2):401–449, 2021.

[31] Pavel Senin and Sergey Malinchik. Sax-vsm: Interpretable time series classification using sax and vector space model. In *2013 IEEE 13th international conference on data mining*, pages 1175–1180. IEEE, 2013.

[32] Huynh Thi Thu Thuy, Duong Tuan Anh, and Vo Thi Ngoc Chau. Segmentation-based methods for top-k discords detection in static and streaming time series under euclidean distance. In *Context-Aware Systems and Applications: 10th EAI International Conference, ICCASA 2021, Virtual Event, October 28–29, 2021, Proceedings 10*, pages 147–163. Springer, 2021.

[33] Lewis Tunstall, Leandro Von Werra, and Thomas Wolf. *Natural language processing with transformers*. " O'Reilly Media, Inc.", 2022.

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

[35] Zhiguang Wang, Weizhong Yan, and Tim Oates. Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International joint conference on neural networks (IJCNN)*, pages 1578–1585. IEEE, 2017.

[36] RE Wilson. Binary star light-curve models. *Publications of the Astronomical Society of the Pacific*, 106(703):921, 1994.

[37] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45, 2020.

[38] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023.

[39] Myat Su Yin, Peter Haddawy, Borvorntat Nirandmongkol, Tup Kongthaworn, Chanaporn Chaisumritchoke, Akara Supratak, Chaitawat Sa-ngamuang, and Patchara Sriwichai. A lightweight deep learning approach to mosquito classification from wingbeat sounds. In *Proceedings of the conference on information technology for social good*, pages 37–42, 2021.

[40] Yuan Yuan and Lei Lin. Self-supervised pretraining of transformers for satellite image time series classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:474–487, 2020.

[41] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 11121–11128, 2023.

[42] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11106–11115, 2021.

[43] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning*, pages 27268–27286. PMLR, 2022.