2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

HydraMamba: An Efficient and High-Performance Architecture for Time Series Classification through Multi-Mechanism Fusion

Peiqi Tang, Mengna Liu, Xin Qin, Yutao Jin and Xu Cheng*

School of Computer Science and Engineering, Tianjin University of Technology
Engineering Research Center of Learning-Based Intelligent System (Ministry of Education)
Key Laboratory of Computer Vision and System (Ministry of Education)
tgcandy@stud.tjut.edu.cn, xu.cheng@ieee.org

Abstract. Multivariate time series classification is a key task in fields such as healthcare, financial analysis, and industrial monitoring. However, existing methods still face challenges in modeling complex dependencies across different time scales, and their computational efficiency is relatively low. To address these issues, we propose an efficient and high-performance model architecture, HydraMamba, which enhances modeling capability by integrating three core mechanisms: The Time Feature Recalibration Module (TFRM) adaptively adjusts the feature weights of time segments to improve the model's ability to focus on key moments; the Multi-Receptive Field Feature Extractor (MRFFE) extracts local and global information in parallel using receptive fields of different sizes, enhancing feature representation; and the Dynamic State-Space Mixer (DSSM), based on state-space modeling, effectively integrates multi-scale temporal features. We conducted extensive experiments on the UEA multivariate time series classification benchmark datasets, where HydraMamba outperformed mainstream methods like TodyNet , achieving overall superior performance and ranking first on 8 datasets. The experimental results show that HydraMamba maintains high computational efficiency while offering superior classification performance, demonstrating strong generalization ability and application potential.

Keywords: Time Series Classification, Multiscale, Spatial Model.

1 Introduction

Multivariate Time Series (MTS) data are ubiquitous across various domains and inherently contain rich dynamic patterns. For this type of MTS data, a key and challenging task is Multivariate Time Series Classification (MTSC). Multivariate MTSC aims to classify MTS samples into predefined categories based on the temporal patterns and inter-variable relationships embedded within the MTS. This is a crucial and challenging task, holding broad application value in fields such as human activity recognition [1], industrial anomaly detection [2], and environmental state recognition [3].

* Corresponding author.

Although a large number of methods have been applied to MTSC, traditional models such as Recurrent Neural Networks (RNNs) and their variants have certain advantages in capturing sequence dependencies. However, effectively modeling long-term dependencies remains a significant challenge for these methods. Particularly when dealing with long sequences, RNNs often face the problem of vanishing or exploding gradients, which limits their practical effectiveness [4]. Additionally, their training process depends on time-step unfolding, leading to lower computational efficiency [5]. The Transformer model, with its self-attention mechanism, models dependencies across sequences globally, significantly improving feature extraction capabilities [6]. However, its computational complexity grows quadratically with the sequence length, and its memory overhead is enormous, making it difficult to apply in resource-constrained scenarios. In recent years, State Space Models (SSMs) have gained widespread attention due to their superior long-sequence modeling capabilities and linear computational complexity [7-9]. Representative works like S4, DSS, and Mamba have demonstrated excellent performance in several sequence modeling tasks [10].

Despite the progress in MTSC, existing methods still face critical challenges in practical applications, mainly in the following two aspects:

First, current approaches exhibit insufficient capability in capturing multi-scale information, which is essential for comprehensively understanding time series data. Existing SSM-based methods typically process temporal features at a single scale, lacking effective mechanisms to capture information across different temporal resolutions. However, time series data are inherently multi-scale, where both short-term patterns and long-term trends carry crucial signals [11]. Single-scale processing is akin to viewing through a narrow tunnel—it struggles to integrate complementary information across temporal granularities, thereby limiting the model's representational power and reducing classification accuracy and robustness.

Second, most existing models lack a dynamic importance adjustment mechanism for different time segments, making them ineffective in addressing the challenge of temporal heterogeneity. Multivariate time series data exhibit significant temporal heterogeneity—different segments contribute unequally to the classification task.Current SSM-based models, do not incorporate adaptive mechanisms to dynamically adjust the importance of time segments, making it difficult to focus on informative parts of the sequence. Consequently, their performance degrades in the presence of redundant information or sparse key features.

To address these challenges, we propose a novel MTSC framework named HydraMamba, which introduces three key architectural innovations designed to enhance multi-scale modeling, dynamic temporal awareness, and effective feature integration . Specifically, the Multi-Receptive Field Feature Extractor (MRFFE) leverages a multi-branch convolutional structure to jointly capture local and global temporal features, thereby improving the model’s capacity for multi-scale representation . The Time Feature Recalibration Module (TFRM) combines temporal pooling with channel attention mechanisms to dynamically emphasize the representation of critical time steps . Finally, the Dynamic State-Space Mixer (DSSM) integrates state-space modeling with deep feature fusion strategies, using a dynamic weighting mechanism to efficiently aggregate short-term and long-term information .

The innovative nature of the HydraMamba framework is further highlighted by the following three points:

- By synergistically combining its three core modules, HydraMamba effectively overcomes the limitations of existing methods in multi-scale modeling, temporal dynamic perception, and feature fusion, thereby significantly enhancing the model's classification performance and generalization ability in high-dimensional, long-sequence tasks.
- MRFFE: It innovatively achieves multi-scale feature extraction, dynamic temporal feature adjustment, and effective long-range dependency modeling by integrating multi-scale convolution, the TFRM, and the Hydra Attention mechanism, thus significantly improving the performance of multivariate time series classification.
- Extensive experiments have been conducted, demonstrating the effectiveness of the HydraMamba model and its core modules, MRFFE and DSSM. HydraMamba has achieved leading classification accuracy and average ranking across multiple datasets. Ablation experiments clearly showcase the contribution of the MRFFE and DSSM modules to the model's performance, and sensitivity analysis [12] reveals the potential impact of model depth on performance.

2 Related Work

In the field of MTSC, numerous studies have proposed various methods to tackle the challenges of sequence modeling. This section reviews and discusses representative methods and compares them with our proposed innovations.

2.1 RNN-based Models

Traditional RNNs and their variants such as LSTM and GRU have been widely adopted in MTSC tasks due to their ability to model temporal dependencies in sequential data [13, 14]. These methods unfold the recurrent structure across time to capture dependencies between time steps. However, their sequential nature leads to limited modeling capability for long sequences, often suffering from vanishing or exploding gradients. This limitation becomes prominent in high-dimensional time series where efficiency and long-range dependency modeling are critical. While models like LSTM-FCN [14] achieve competitive performance by combining CNN and LSTM layers, they still struggle to capture long-range dependencies among multiple variables.

2.2 Transformer-based Models

Transformer architectures [6] leverage self-attention mechanisms to globally model dependencies, and have shown great success in both NLP and time series applications [15, 18], as evidenced by models like TST [19] and Tapnet [20]. For example, TST utilizes attention and positional encoding to model temporal patterns, while Tapnet introduces attention mechanisms for temporal pattern learning. In contrast, CNN-based models like XDM-CNN [21] offer alternative approaches for multivariate time series classification. Nevertheless, the quadratic time and memory complexity of Transformer architectures limits their scalability to long sequences, which is particularly problematic in real-world MTSC scenarios. Recent efforts also explore self-supervised Transformer models, such as MLSTM [14] and TST, which learn general-purpose sequence representations. Although they enhance generalization and data efficiency, their transformer backbone still suffers from scalability issues in ultra-long sequences.

2.3 State Space Models and Mamba

To address the inefficiencies of Transformers in long sequence modeling, structured SSMs [7] and recurrent-based designs [22] have been proposed. Among these, Mamba [10] stands out as a recent breakthrough, offering efficient long-range modeling through selective SSMs. Mamba eliminates the need for explicit attention or MLP blocks by employing input-dependent dynamic state updates, leading to hardware-efficient recurrent computation with linear scalability. With its high throughput and strong performance across language, audio, and genomics, Mamba demonstrates its potential as a robust alternative to Transformers for time series tasks.

However, most SSM-based models still focus on single-scale feature extraction, lacking the capacity to model hierarchical or multiscale temporal structures. This limits their applicability in complex MTSC tasks where both local and global patterns are critical. Building on these foundations, we introduce HydraMamba for MTSC, featuring three key innovations for enhanced multi-scale modeling, dynamic temporal awareness, and effective feature integration. Unlike traditional methods, HydraMamba's MRFFE uses multi-branch convolutions for joint local and global temporal feature capture, improving multi-scale representation. Differing from static temporal feature processing, TFRM dynamically emphasizes critical time steps via temporal pooling and channel attention. Instead of traditional sequence model architectures, DSSM integrates state-space modeling with deep feature fusion and dynamic weighting for efficient short and long-term information aggregation. HydraMamba offers a more effective, comprehensive framework for complex MTSC data challenges.

3 Method

We propose HydraMamba, a novel time series classification model designed to capture long-range dependencies and multi-resolution patterns. The core of the model consists of three key components: MRFFE, TFRM, and DSSM. These modules work synergistically to enhance temporal modeling, adaptive feature weighting, and multi-scale information extraction. Each module is described in detail below.

3.1 Multi-Receptive Field Feature Extractor

The MRFFE is introduced to enhance the model's capacity to capture multi-scale temporal patterns by integrating multi-scale convolutional perception and attention mechanisms. MRFFE begins by aligning input feature dimensionality with a 1×1 convolution. Subsequently, it employs parallel or cascaded convolution layers, $Conv_{k_i}$, each with a distinct kernel size k_i from a predefined set, to extract multi-scale temporal patterns as

$$X_i = \mathrm{Conv}_{k_i}(X_{i-1}), \quad i = 1, \dots, L$$

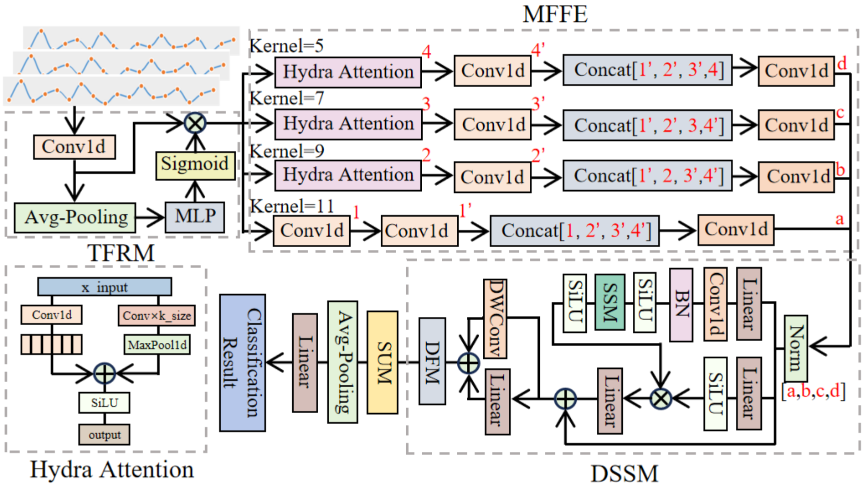


Fig. 1. llustration of the proposed HydraMamba, its three key components: 1) Multi-Receptive Field Feature Extractor, 2) Temporal Feature Recalibration Module, and 3) Dynamic State Space Mixer. Traditional models typically employ a single-scale approach, and when prcessing time series data, struggle to focus on key time segments and suppress redundant information. In contrast, we adopt a multi-scale approach and effectively handle time heterogeneity. Through these modules, we achieve a more comprehensive understanding of the data, enhance the model's robustness in complex and noisy environments, and achieve higher computational efficiency.

Following initial convolutions, the TFRM dynamically adjusts temporal feature responses. Finally, MRFFE aggregates all intermediate layer outputs $\{X\_1, X\_2, \dots, X\_L\}$ into a multi-scale feature set, creating a hierarchical representation that captures both shallow local details and deeper semantic structures for downstream tasks.

To further enhance the model's perception capabilities, particularly in capturing long-range dependencies, MRFFE incorporates Hydra Attention. This mechanism replaces traditional convolution operations in subsequent layers, adaptively focusing on crucial time steps. Hydra Attention achieves this by learning position-sensitive weights and is applied after a demensional transformation and before restoring the dimensional order, as expressed by

$$X_i = \mathrm{HydraAttn}(X_i^{\top})^{\top}$$

This attention mechanism enables the network to effectively model long-range dependencies while preserving temporal structure, proving particularly beneficial for complex temporal patterns.

3.2 Temporal Feature Recalibration Module

To further enhance the modeling of crucial temporal features, we introduce the TFRM, inspired by the SE module, designed to capture time steps exhibiting key dynamic patterns while suppressing redundant temporal information. TFRM commences by processing the input feature tensor $X \in R^{B \times C \times T}$ through adaptive temporal pooling along the time dimension, resulting in a global channel context $X_{avg} \in R^{B \times C \times \mathbb{1}}$. These compressed features are then channeled into a two-layer fully connected bottleneck structure, incorporating a ReLU activation in between, to generate dynamic recalibration weights $W \in R^{B \times C \times \mathbb{1}}$ as per the formula

$$W = \sigma\left(W_2\left(\mathrm{ReLU}\left(W_1(X_{\mathrm{avg}})\right)\right)\right)$$

where $W_1 \in R^{C \times C/r}$, $W_2 \in R^{C/r \times C}$, σ represents the Sigmoid function, and r is the compression ratio. Ultimately, feature recalibration is accomplished by applying these generated weights W to the original input features X through channel-wise multiplication, expressed as $\tilde{X} = X \otimes W$. This effectively modulates feature responses by amplifying channels with strong activations at key time points and diminishing segments less informative to the model. Through this mechanism of temporal pooling and channel recalibration, TFRM dynamically refines feature responses to concentrate on pivotal temporal patterns.

3.3 Multi-Receptive Field Feature Extractor

The DSSM is designed to enhance the model's ability to model long-range dependencies and complex temporal dynamics, and to achieve comprehensive multi-scale temporal feature modeling in a computationally efficient manner. The core building block of DSSM is the DynamiX Layer. This layer first expands the channel dimension of the input features from $d \rightarrow 2d$ through a linear transformation to enhance representation capacity. Subsequently, it utilizes 1D convolution to extract local patterns and introduces activation functions (such as SiLU) and BatchNorm for normalization, as shown in the following formula:

$$X_{conv} = \mathrm{BN}\left(\mathrm{SiLU}\left(\mathrm{Conv1D}(X)\right)\right) \tag{1}$$

where X is the input feature and X_{conv} is the convolved feature. Critically, the DynamiX Layer introduces a SSM to effectively model temporal dependencies and enhances information retention through a residual multiplication fusion mechanism:

$$X_{res} = X \odot \mathrm{SSM}(X_{conv}) \tag{2}$$

Here, X_{res} represents the residually fused feature, and $\odot$ denotes element-wise multiplication. Following each DynamiX Layer, a Depthwise Feature Mixer (DFM) is introduced to enhance feature fusion. DFM employs a combination of fully connected layers and depthwise separable convolution (DWConv) to perform local enhancement and nonlinear modeling, incorporating GELU activation functions, Dropout regularization, and outputting mixed features through the final fully connected layer. To ensure gradient stability and information coherence, DSSM incorporates residual connections in both DynamiX and DFM sub-modules:

$$\tilde{X} = X + \mathcal{F}(X) \tag{3}$$

where $\mathcal{F}(\cdot)$ represents the processing function of the DynamiX or DFM module, and $\tilde{X}$ is the module output. Furthermore, the model adopts normalization techniques such as LayerNorm or RMSNorm to mitigate covariate shift during training and accelerate convergence. Finally, to further enhance the training stability of deep networks, DSSM introduces the LayerScale mechanism. A learnable scaling factor γ is added before the output of each sub-module, with the formula:

$$\mathcal{F}_{\text{sae}}(X) = \gamma \cdot \mathcal{F}(X) \tag{4}$$

LayerScale plays a crucial role in regulating gradient updates and feature responses in deep structures.

4 Experiments

The UEA (University of East Anglia) multivariate time series classification archive serves as a valuable resource for evaluating the performance of various multivariate time series classifiers. This comprehensive archive contains a total of 30 distinct datasets that have been specifically curated for the task of MTSC and has gained considerable recognition within the research community for its utility.These datasets are systematically organized into six primary categories, each characterized by specific attributes. Therefore, this structured presentation of information allows researchers and practitioners to easily compare and select appropriate datasets for their classification tasks, facilitating a more effective assessment of the classifiers' capabilities in handling multivariate time series data. For this study, we used 30 datasets to evaluate the performance of our method, as showed Table 1.

4.1 Experiment settings

The PC used for the trials in this study has the following components: two NVIDIA GeForce GTX 3090 GPUs. All neural networks are built using Python 3.8/3.9, pytroch 11.3 as the backend. Evaluation Metrics Given that the comparison experiments are performed across a diverse range of datasets from different domains, a variety of evaluation metrics are selected to demonstrate the classifiers' performance. The classification accuracy on the test sets serves as the most straightforward indicator of model effectiveness. Therefore, we consider average accuracy, average rank, providing a comprehensive assessment of a model's classification performance across various datasets.

4.2 Evaluation Metrics

Given that the comparison experiments are performed across a diverse range of datasets from different domains, a variety of evaluation metrics are selected to demonstrate the classifiers' performance. The classification accuracy on the test sets serves as the most straightforward indicator of model effectiveness. Therefore, we consider average accuracy, average rank, providing a comprehensive assessment of a model’s classification performance across various datasets.

Table 1. Classification Accuracy Comparison on UEA Datasets.

Methods Dataset	EDI	DTWD	WEA+MU	MLSTM	Tapnet	TST	XDM-CNN	To-dyNet	OURS
ArticularyWordRecognition	0.970	0.987	0.990	0.973	0.987	0.983	**0.993**	0.987	0.990
AtrialFibrillation	0.267	0.220	0.333	0.267	0.333	0.200	0.600	0.467	**0.667**
BasicMotions	0.676	0.975	**1.000**	0.950	**1.000**	0.975	**1.000**	**1.000**	**1.000**
CharacterTrajectories	0.964	0.989	0.990	0.985	**0.997**	0.000	0.972	N/A	0.930
Cricket	0.944	**1.000**	**1.000**	0.917	0.958	0.958	N/A	**1.000**	0.986
DuckDuckGeese	0.275	0.600	0.575	**0.675**	0.575	0.480	**0.675**	0.580	0.300
ERing	0.133	0.929	0.133	0.133	0.133	0.933	**0.967**	N/A	0.133
EigenWorms	0.549	0.618	**0.890**	0.504	0.489	N/A	0.756	0.840	0.542
Epilepsy	0.666	0.964	**1.000**	0.761	0.971	0.920	**1.000**	0.971	0.986
EthanolConcentration	0.293	0.323	**0.430**	0.373	0.323	0.337	0.399	0.350	0.416
FaceDetection	0.519	0.529	0.545	0.545	0.556	**0.681**	**0.681**	0.627	0.573
FingerMovements	0.550	0.530	0.490	0.580	0.530	**0.776**	0.590	0.570	0.590
HandMovementDirection	0.278	0.231	0.365	0.365	0.378	0.608	0.608	**0.649**	0.473
Handwriting	0.200	0.286	**0.605**	0.286	0.357	0.305	0.498	0.436	0.399
Heartbeat	0.619	0.717	0.727	0.663	0.751	0.712	0.717	0.756	**0.922**
InsectWingbeat	0.128	N/A	N/A	0.167	0.208	**0.684**	N/A	N/A	0.100
JapaneseVowels	0.924	0.949	0.973	0.976	0.965	**0.994**	N/A	N/A	0.927
LSST	0.456	0.551	0.590	0.373	0.568	0.381	0.547	0.615	**0.663**
Libras	0.833	0.870	0.878	0.856	0.850	0.844	0.800	0.850	**0.917**
MotorImagery	0.510	0.500	0.500	0.510	0.590	N/A	0.600	**0.640**	0.550
NATOPS	0.850	0.883	0.870	0.889	0.939	0.900	0.883	0.972	**0.978**
PEMS-SF	**0.973**	0.711	N/A	0.699	0.751	0.919	0.863	0.780	0.809
PenDigits	0.705	0.977	0.948	0.978	0.980	0.974	**0.987**	N/A	0.104
PhonemeSpectra	0.104	0.151	0.190	0.110	0.175	0.088	0.231	0.309	**0.421**
RacketSports	0.868	0.803	**0.934**	0.803	0.868	0.829	0.822	0.803	0.908
SelfRegulationSCP1	0.771	0.775	0.710	0.874	0.652	**0.925**	0.922	0.898	0.867
SelfRegulationSCP2	0.483	0.539	0.460	0.472	0.550	0.589	0.494	0.550	**0.628**
SpokenArabicDigits	0.967	0.963	0.982	0.990	0.983	**0.993**	N/A	N/A	0.312
StandWalkJump	0.200	0.200	0.333	0.067	0.400	0.267	**0.600**	0.467	0.400
UWaveGestureLibrary	0.881	0.903	**0.916**	0.891	0.894	0.903	0.897	0.850	0.863
Average rank	5.933	4.800	3.867	4.967	4.000	4.167	3.433	3.900	3.400
Number of top-1	1	1	8	1	2	6	8	4	8

4.3 Overall Performance Comparison

Table 1 showcases a comparative analysis of classification accuracy on the UEA benchmark datasets, contrasting our proposed HydraMamba model against seven other state-of-the-art time series classification methods. The HydraMamba model achieved a remarkable leading average rank of 3.40 across 30 datasets, tying for first place with the XDM-CNN model. This demonstrates its exceptional generalization capability in diverse time series tasks. Notably, the HydraMamba model secured Top-1 accuracy on 8 datasets, sharing the top position with WEASEL+MUSE and XDM-CNN, and outperforming a range of robust competitive methods such as TST and DTWD.

On datasets characterized by more complex temporal dynamics, such as Heartbeat (0.922), SelfRegulationSCP1 (0.867), and PhonemeSpectra (0.421), the HydraMamba model exhibited a significant advantage over other methods. These performance gains are primarily attributed to the uniquely designed MRFFE and the DSSM within the model. The MRFFE effectively captures crucial features across multiple temporal scales, while the DSSM enhances the model's ability to model long-range dependencies. Furthermore, the TFRM, embedded within the MRFFE, dynamically adjusts the weights of time series features, effectively suppressing redundant information and noise interference.

In contrast, models such as MLSTM and EDI demonstrated relatively unstable performance across multiple datasets. This is mainly due to their inherent limitations in modeling hierarchical temporal dependencies and judging feature importance, which restricts their effectiveness in complex time series tasks.

Table 2. Ablation Study: Performance Impact of Removing DSSM and MRFFE on Representative Datasets.

Dataset \ Method	w/o DSSM	w/o MRFFE	OURS
EthanolConcentration	0.316	0.342	**0.416**
Heartbeat	0.751	0.839	**0.898**
JapaneseVowels	0.868	0.922	**0.927**
PEMS-SF	0.792	0.746	**0.804**
Phoneme	0.239	0.336	**0.421**
SelfRegulationSCP2	0.622	0.533	**0.628**
Average	0.598	0.620	**0.682**

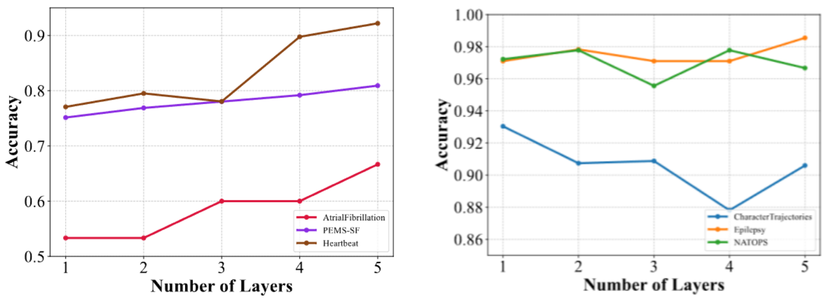
4.4 Ablation Study

To further validate the contribution of each module to the overall performance, we conducted ablation studies on six representative datasets by removing the DSSM module based on Mamba and the MRFFE module separately. The results in Table 2 clearly demonstrate that removing either module leads to a performance degradation. For instance, on the Heartbeat dataset, the accuracy decreases from 0.8976 with the full model to 0.8390 without MRFFE, and further to 0.7512 without DSSM, highlighting the significance of both modules in modeling crucial dynamic patterns. On the Phoneme dataset, removing DSSM results in a drastic performance drop to 0.2392, indicating its effectiveness in capturing complex temporal structures within speech data. Even on relatively stable datasets such as PEMS-SF, the complete model still outperforms both variants, demonstrating its superior generalization capability. On average, our full HydraMamba model achieves an accuracy of 0.6821, significantly outperforming the MRFFE-removed variant (0.6196) and the DSSM-removed variant (0.5979). In summary, the ablation study confirms that the synergy between DSSM and MRFFE is pivotal to HydraMamba's strong performance. Their combination not only enhances multiscale temporal representation but also improves the robustness and expressiveness of the model.

4.5 Sensitivity Analysis

To further evaluate the impact of each layer on overall performance, we based on the results shown in Figure 2, model performance sensitivity to the number of layers is dataset-dependent, but the general trend indicates that deeper models tend to improve performance, especially with 5-layer configurations often exhibiting optimal or near-optimal performance across multiple datasets.While some datasets are sensitive to the number of layers or achieve good performance with shallower architectures, 5-layer models, overall, demonstrate greater potential.Determining the optimal number of layers still requires dataset-specific experimental tuning to balance performance, generalization, and efficiency.

This analysis suggests that model depth should be adapted to the data, rather than adopting a one-size-fits-all approach. Experimentation remains crucial for selecting the appropriate number of layers.



(a) Layer Number Impact on Datasets (b) Layer Number Impact on Datasets

Fig. 2. Sensitivity Analysis

5 Conclusions

Introducing HydraMamba, this paper presents a novel multiscale time series classification model engineered to effectively capture temporal dynamics through a synergistic architecture of three key modules: the MRFFE, designed to extract discriminative features across multiple temporal resolutions; the TFRM, which adaptively highlights salient temporal patterns; and the DSSM, inspired by dynamic state-space modeling for computationally efficient enhancement of long-range dependency modeling. The efficacy of HydraMamba is comprehensively validated by experiments conducted on 30 UEA benchmark datasets, demonstrating its superior performance by achieving the best average rank (3.400) and matching the highest number of top-1 results (8 datasets), thus outperforming a range of state-of-the-art baselines, further supported by ablation studies that confirm the substantial contributions of both MRFFE and DSSM to these performance gains, underscoring the powerful synergy between multiscale representation and dynamic sequence modeling inherent in HydraMamba.

In the future, we plan to extend HydraMamba to multivariate, online, and long-form time series tasks, and explore its adaptability in real-world applications such as healthcare monitoring, financial forecasting, and industrial process analysis.

Acknowledgments. This work was partially supported by the National Natural Science Foundation of China under Grants 62306212, T2422015, and 62020106004, as well as by the Tianjin Natural Science Foundation under Grant 23JCJQJC000070. Xu Cheng also received support from the Marie Skłodowska-Curie Actions (MSCA) under Project No. 101111188.

References

1. Henry Friday Nweke, Ying Wah Teh, Mohammed Ali Al-Garadi, and Uzoma Rita Alo. Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges. Expert Systems with Applications, 105:233–261, 2018.
2. Achyut Mani Tripathi and Rashmi Dutta Baruah. Anomaly detection in multivariate time series using fuzzy adaboost and dynamic naive bayesian classifier. In 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), pages 1938–1944. IEEE, 2019.
3. Xu Cheng, Guoyuan Li, Andre Listou Ellefsen, Shengyong Chen, Hans Petter Hildre, and Houxiang Zhang. A novel densely connected convolutional neural network for sea-state estimation using ship motion data. IEEE Transactions on Instrumentation and Measurement, 69(9):5984–5993, 2020.
4. Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.
5. Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. IEEE Transactions on Audio, Speech, and Language Processing, 22(4):584–595, 2013.
6. A Vaswani. Attention is all you need. Advances in Neural Information Processing Systems, 2017.
7. Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. arXiv preprint arXiv:2111.00396, 2021.
8. Albert Gu, Karan Goel, Ankit Gupta, and Christopher Ré. On the parameterization and initialization of diagonal state space models. Advances in Neural Information Processing Systems, 35:35971–35983, 2022.

9. Antonio Orvieto, Samuel L Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. Resurrecting recurrent neural networks for long sequences. In International Conference on Machine Learning, pages 26670–26698. PMLR, 2023.
10. Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. arXiv preprint arXiv:2312.00752, 2023.
11. Hegui Zhu, Jiapeng Zhang, Hao Cui, Kai Wang, and Qingsong Tang. Tcran: Multivariate time series classification using residual channel attention networks with time correction. Applied Soft Computing, 114:108117, 2022.
12. Andrea Saltelli. Sensitivity analysis. In Encyclopedia of Quality of Life and WellBeing Research, pages 6325–6326. Springer, 2024.
13. 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.
14. Fazle Karim, Somshubra Majumdar, Houshang Darabi, and Samuel Harford. Multivariate lstm-fcns for time series classification. Neural networks, 116:237–245, 2019.
15. 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.
16. Rundong Zuo, Guozhong Li, Byron Choi, Sourav S Bhowmick, Daphne Ngar-yin Mah, and Grace LH Wong. Svp-t: a shape-level variable-position transformer for multivariate time series classification. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 37, pages 11497–11505, 2023.
17. Navid Mohammadi Foumani, Chang Wei Tan, Geoffrey I Webb, and Mahsa Salehi. Improving position encoding of transformers for multivariate time series classification. Data Mining and Knowledge Discovery, 38(1):22–48, 2024.
18. Shuxin Li, Xu Cheng, Fan Shi, Hanwei Zhang, Hongning Dai, Houxiang Zhang, and Shengyong Chen. A novel robustness-enhancing adversarial defense approach to ai-powered sea state estimation for autonomous marine vessels. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 55(1):28–42, 2025.
19. George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. A transformer-based framework for multivariate time series representation learning. In Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining, pages 2114–2124, 2021.
20. Xuchao Zhang, Yifeng Gao, Jessica Lin, and Chang-Tien Lu. Tapnet: Multivariate time series classification with attentional prototypical network. In Proceedings of the AAAI conference on artificial intelligence, volume 34, pages 6845–6852, 2020.
21. Wei Cai, Xiaomin Zhu, Kaiyuan Bai, Aihui Ye, and Runtong Zhang. An explainable dual-mode convolutional neural network for multivariate time series classification. Knowledge-Based Systems, 299:112015, 2024.
22. Matteo Tiezzi, Michele Casoni, Alessandro Betti, Tommaso Guidi, Marco Gori, and Stefano Melacci. On the resurgence of recurrent models for long sequences: Survey and research opportunities in the transformer era. arXiv preprint arXiv:2402.08132, 2024.