Journal of Applied Analysis and Computation
Volume 16, Number 1, February 2026, 246–269
Website:http://www.jaac-online.com
DOI:10.11948/20250101

USING HOMOTOPY MULTI-HIERARCHICAL ENCODER REPRESENTATION FROM TRANSFORMERS (HMHERT) FOR TIME SERIES CHAOS CLASSIFICATION

Di Yu[1] and Xue Yang[2,†]

Abstract The Transformer architecture, renowned for its exceptional capacity in processing long-sequence data, inspires our framework that leverages its self-attention mechanism to classify time series through relational analysis between sequence elements. In this article, we propose a Homotopy Multi-Hierarchical Encoder Representation from Transformers (HMHERT), for chaotic/non-chaotic sequence classification. Empirical investigations into the linear combination coefficients of multi-head attention reveal that constrained homotopy coefficients significantly enhance model performance, with homotopy constrained configurations outperforming their unconstrained coefficient counterparts. Through systematic comparative analysis of Confusion Matrix, classification accuracy, F1-scores, and Matthews Correlation Coefficient (MCC), HMHERT exhibits significantly enhanced generalization performance, outperforming conventional models including Time-Delayed Reservoir Computing (RC), Fully Connected Neural Network (FCNN), Long Short Term Memory (LSTM), and Convolutional Neural Network (CNN) by 0.5097-0.9204 across MCC metrics. Furthermore, compared to the baseline Transformer encoder architecture, HMHERT achieves performance improvement, demonstrating the critical role of our proposed architectural modifications in chaotic pattern recognition.

Keywords Chaotic classification, homotopy, multi-hierarchical, Transformer.

MSC(2010) 68T10.

1. Introduction

In the field of nonlinear science, the "butterfly effect" is a widely recognized concept that transcends academic disciplines and captures the public's imagination. First introduced by Lorenz [22] during his weather forecasting research, it vividly demonstrates the sensitive dependence of deterministic dynamical systems on initial conditions. This seminal insight sparked extensive research into chaos theory, which later became the cornerstone of nonlinear science, with applications in weather prediction, celestial mechanics, economics, and biology.

Chaotic phenomena exhibit complex behaviors such as bifurcations, strange attractors, and multi-stability, which prompt the development of tools to identify and classify chaos. The Lyapunov exponent remains a classical metric but suffers from limitations, such as misclassifying quasi-periodic signals as chaotic [3]. In contrast, Shannon entropy requires predefined thresholds, introducing subjectivity [3]. Early studies explored entropy-based measures, such as topological sequence entropy [4] and the entropy of planar curves [2]. Recent advancements have shifted

[†]The corresponding author.
[1]School of Mathematics and Statistics, Northeast Normal University, Changchun 130024, China
[2]College of Mathematics, Jilin University, Changchun 130012, China
Email: yud140@nenu.edu.cn(D. Yu), xueyang@jlu.edu.cn(X, Yang)

toward leveraging artificial neural networks for chaos classification. Carroll [5] proposed a RC approach for chaotic signal discrimination, employing time-delayed training signals to capture temporal dependencies in chaotic systems. Liedji et al. [18, 19] demonstrated effective classification of hyper-chaotic, chaotic, and regular signals through a single-node delayed RC architecture. The model has a correct rate of regular signal classification of more than 95% and a correct rate of chaotic signal classification of more than 99%. Under the standard mapping data training, the generalization classification accuracy rate of the Lorenz system reached 74.42% of the model. Boullé et al. [3] developed a large-kernel CNN for chaotic signal classification, eliminating the need for prior knowledge of complex system dynamics while exhibiting exceptional generalization capabilities in cross-system classification tasks. Mukhopadhyay and Banerjee [24] addressed the challenge of distinguishing chaotic from stochastic dynamical systems under varying noise conditions by transforming time series into time-frequency spectrograms and threshold-free recurrence texture images, leveraging CNNs for discrimination. Szczęsna et al. [30] established a comprehensive dataset of dynamical system time series and implemented both LSTM networks and CNNs for chaotic data classification. These studies address the classification challenge of distinguishing chaotic from non-chaotic dynamics in time series analysis, with non-chaotic subtypes selectively operationalized across studies as combinations of periodic, aperiodic, quasi-periodic, and stochastic dynamics rather than encompassing all categories.

To overcome these challenges, we turn our attention to the Transformer model introduced by Vaswani et al. [31] in 2017. This architecture employs self-attention mechanisms to process sequential data, excelling in tasks that require long-range dependency modeling and parallel computation. Its applications now extend beyond NLP to fields including image processing [14], biology [1], chemistry [32], mechanics [11], and time series forecasting [35]. The release of ChatGPT by OpenAI in 2022 represents a pivotal advancement in the evolution of large language models. This achievement is underpinned by the Transformer architecture, which serves as the foundational framework for the model. The development of ChatGPT builds upon OpenAI's earlier contributions, notably the introduction of a semi-supervised learning approach in 2018, which laid the groundwork for subsequent innovations in this domain [26]. In the same year, Google [9] introduced the Bidirectional Encoder Representations from Transformers, which leverages unmasked self-attention mechanisms to enable bidirectional contextual modeling. In subsequent years, Transformer-based models have been further refined and adapted to address specific challenges. For instance, the Informer model, proposed by Zhou et al. [37] in 2021, optimized the Transformer architecture for long-sequence time series forecasting by mitigating issues such as quadratic time complexity and high memory demands. Geneva and Zabaras [12] extended these advancements to dynamical systems, employing Koopman embeddings to enhance predictive accuracy. However, the effectiveness of Transformer-based models for long-term forecasting has been questioned, as evidenced by Zeng et al.'s study [34], which demonstrated that simple linear models could outperform Transformer-based approaches on real-world datasets in multi-step forecasting tasks. Zhornyak et al. [36] showcased the application of Transformer architectures for inferring bifurcation diagrams of dynamical systems from noisy data, further expanding the scope of these models.

In the optimization research of Transformer model, improving the attention mechanism is an important direction to improve the performance of the model. Scholars have proposed a variety of innovative solutions for different application scenarios, forming a multi-dimensional technology evolution path. Taking cross-modal modeling as the starting point, the X-Linear attention module proposed by Pan et al. [25] effectively captures the interaction between cross-modal or unimodal features by introducing bilinear pooling technology. By stacking multiple layers of

X-Linear modules, the scheme can further establish high-order feature interactions, which shows significant advantages in image description generation tasks. In terms of computational efficiency optimization, the Shifted Window Self-Attention developed by Liu et al. [21] reduces computational complexity through local window calculations and adopts a window shifting strategy to achieve cross-window information interaction. This method significantly improves computational efficiency while ensuring the expressiveness of the model. The BiFormer model proposed by Zhu et al. [38] introduces Bi-Level Routing Attention. Its innovation lies in the design of a dynamic sparse attention mechanism: First, the feature map is divided into several regions, and the top k most relevant neighbors of each region are retained based on regional correlation, and then fine-grained token-level attention calculations are performed. This hierarchical screening mechanism reduces computational complexity while maintaining the ability to model long-distance dependencies. In terms of innovation in the attention mechanism architecture, the Agent Attention designed by Han et al. [13] creatively introduces agent tokens as intermediaries between queries and key-value pairs in the traditional attention framework. This hybrid architecture effectively combines the high expressiveness of Softmax attention and the computational efficiency of linear attention. In view of the particularity of visual features, Sun et al. [29] proposed Histogram Self-Attention, which made a breakthrough in using image pixel intensity as the basis for feature organization: First, the histogram intervals are divided by intensity value, and then the attention calculations within and across intervals are performed. This feature reorganization strategy based on statistical characteristics shows stronger semantic perception ability in image segmentation tasks. In terms of model light-weighting, the Single-Head Vision Transformer developed by Yun and Ro [33] adopts a hybrid attention architecture: Single-Head Self-Attention is used in specific channels to replace the traditional multi-head mechanism, while integrating local and global features. This design effectively reduces memory consumption while maintaining model performance. Su et al. [28] integrated recursive graphs with Transformer to extract temporal features from two different branches, where the Transformer takes a transposed form to comprehensively analyze the relationship between multiple variables in the time series. Compared with traditional machine learning and deep learning methods, this model achieves higher prediction accuracy.

Incorporating diverse technologies and theories into neural networks has become a prevalent approach for enhancing model performance. Among these advancements, multi-scale techniques represent a well-established and mature methodology for improving neural networks. By accounting for dynamic changes across different temporal resolutions, multi-scale methods have demonstrated efficacy in various domains, including medical image segmentation [27], time series analysis [7], image recognition [6, 10] and engineering detection [8]. When coupled with self-attention, multi-scale methods bolster the Transformer's ability to handle data spanning multiple scales, thereby improving predictive accuracy [15, 20, 23]. Furthermore, Li and Li [16, 17] introduced the homotopy theory into neural network architectures, combining activation functions in a manner inspired by homotopy theory, resulting in a notable improvement in the predictive capability of neural networks. This innovation opens new avenues for enhancing neural network designs.

In this study, we introduce HMHERT, a novel framework to address chaotic system classification challenges. This framework integrates multi-head attention mechanisms inspired by homotopy theory, enabling multi-hierarchical analysis to capture hierarchical relationships in time series data. HMHERT tackles the complexities of chaotic system classification by capturing hierarchical relationships in time series data. The remainder of this paper is structured as follows: Section 2 reviews the Transformer encoder and details our proposed enhancements; Section 3

provides an in-depth introduction to the five categories of data, encompassing both chaotic and non-chaotic types; and Section 4 examines the impact of homotopy coefficients on the proposed method, compares the classification performance of various artificial neural networks with that of the proposed approach, and explores the selection of homotopy coefficients as well as the influence of homotopy theory on the model's performance.

2. Transformer and homotopy theory

The Transformer model, introduced by Vaswani et al. [31] in 2017, is a deep learning architecture based on the self-attention mechanism. Unlike traditional neural networks, the Transformer model is centered around the self-attention mechanism, enabling efficient processing of long sequences and offering strong parallelization capabilities. The Transformer model was originally developed to address NLP tasks. In these tasks, the input sentences typically vary in length. Even when sentences exhibit identical character/word counts, tokenization heterogeneity inevitably induces divergent dimensions in their embedded vector sequences. In this study, we employ Transformer models to address the task of chaotic classification in time series data. In this task, we keep the input sequence with the same length. Each token represents an individual numerical element within the sequence, where linear projection layers replace conventional embedding layers to transform raw input tokens into feature representations. In our framework, the input sequence undergoes linear projection to derive latent features while maintaining strict dimensional consistenc across temporal positions.

2.1. Encoder of Transformer

Given time series data $X = [x_1, ..., x_k, ..., x_L] \in \mathbb{R}^L$, where L is the sequence length, x_k is the k th time series signal. In NLP tasks, language input sequences cannot be directly processed by the model; they must first pass through an embedding layer to be transformed into a suitable representation for computational operations. In contrast, in the chaotic classification task, we employ a linear layer f to obtain the sequence S.

$$f(X) = S : \mathbb{R}^L \Rightarrow \mathbb{R}^{L\times d}, \tag{2.1}$$

where d is an embedding dimension. Within the classical Transformer framework, position encoding information should be incorporated after obtaining the sequence S. In NLP tasks, position encoding plays a critical role as it enables Transformer models to capture the sequential order of elements. However, in the chaotic classification task proposed in this study, we compared the classification performance of models with and without position encoding and observed no significant differences. As a result, position encoding information is omitted in the Transformer-based models employed in this study. Consequently, the generated time series S is directly fed into the self-attention mechanism. The computation method for the self-attention mechanism is as follows [31]:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{2.2}$$

where $Q \in \mathbb{R}^{L_Q\times d_k}$, $K \in \mathbb{R}^{L_K\times d_k}$, and $V \in \mathbb{R}^{L_V\times d_k}$ represent the query, key, and value matrices, with d_k representing the dimension of the key vectors. The core of the Transformer model is composed of an encoder and a decoder. In numerous studies, encoders and decoders can used

to be employed independently due to variations in task requirements. Generally, the encoder is preferred for classification tasks, while the decoder is used for prediction tasks.

Similar to CNNs, both the encoder and decoder are constructed by stacking multiple identical blocks. Each block consists of a multi-head attention mechanism, a fully connected layer, and layer normalization, with a residual connection structure. The definition of multi-head attention is as follows [31]:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \cdots, \text{head}_h)W^O, \tag{2.3}$$

where $\text{head}_j = \text{Attention}(QW_j^Q, KW_j^K, VW_j^V)$. This study focuses on the Transformer's encoder for classification tasks. The core architecture consists of three identical stacked blocks, each with a multi-head attention mechanism (using two attention heads). A fully connected layer with 256 neurons is used in place of the word embedding layer, and positional encoding is omitted.

2.2. Homotopy multi-hierarchical encoder representation from Transformer

We focus on the encoder component of the Transformer model and aim to enhance its classification performance by refining its self-attention mechanism. Drawing inspiration from multi-scale studies in dynamic systems, we aspire for the new model to analyze sequences at different hierarchical levels. Furthermore, successful applications of homotopy theory in improving neural network predictive capabilities have also motivated our work. By integrating these two approaches, we propose a novel multi-head attention mechanism, which is defined as follows:

$$\text{New MultiHead}(Q, K, V) = \sum_{i=0}^{1} h_i * \text{Concat}(\text{head}_1^i, \cdots, \text{head}_h^i)W^{O_i}, \tag{2.4}$$

where $\text{head}j^i = \text{Attention}(Q^iW_j^{Q,i}, K^iW_j^{K,i}, V^iW_j^{V,i})$. Here, $Q^0 \in \mathbb{R}^{L_{Q_0} \times d_k}$ and $Q^1 \in \mathbb{R}^{m \times d_k}$ represent the query, $K^0 \in \mathbb{R}^{L_{K_0} \times d_k}$ and $K^1 \in \mathbb{R}^{m \times d_k}$ represent the key, $V^0 \in \mathbb{R}^{L_{V_0} \times d_k}$ and $V^1 \in \mathbb{R}^{m \times d_k}$ represent the value. Where $h_0 + h_1 = 1$, $h_0 \geq 0$ and $h_1 \geq 0$ denotes the homotopy coefficient. d_k denotes the dimension of the key vectors. This multi-head attention mechanism, which we call "Homotopy Multi-Hierarchical Multi-Head Attention Mechanism", is connected in a homotopy manner by two distinct multi-head attention mechanisms, as shown in Figure 1. The Scaled Dot-Product Attention [31] in Figure 1 is the same as the classic Transformer. This homotopy coefficient can either be predetermined before training or learned during the training process. The two multi-head attention mechanisms are capable of analyzing data at different hierarchical levels and can also operate at distinct temporal scales to analyze the input sequence.

The new multi-head attention mechanism simultaneously imposes new requirements on the input data. Given time series data $X = [x_1, ..., x_k, ..., x_L] \in \mathbb{R}^{L \times m}$, where L is the sequence length, m is multi-hierarchical coefficient, x_k is the kth time series signal. When $m > 1$, X can be utilized for analysis as multi-scale data. we employ a linear layer f to obtain the sequence S:

$$f(X) = S : \mathbb{R}^{L \times m} \Rightarrow \mathbb{R}^{L \times m \times d}, \tag{2.5}$$

where d is an embedding dimension.

The encoder incorporating the homotopy multi-hierarchical multi-head attention mechanism is referred to as HMHERT, and its structure is illustrated in Figure 2. Within identical parameter configuration domains, HMHERT maintains architectural alignment with the Transformer encoder, preserving structural consistency across shared components.

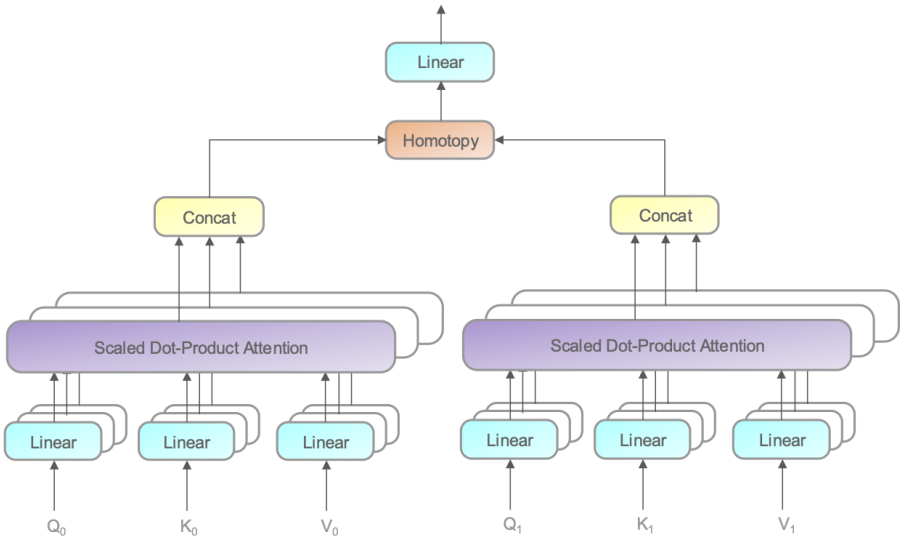

Figure 1. Homotopy multi-hierarchical multi-head attention.

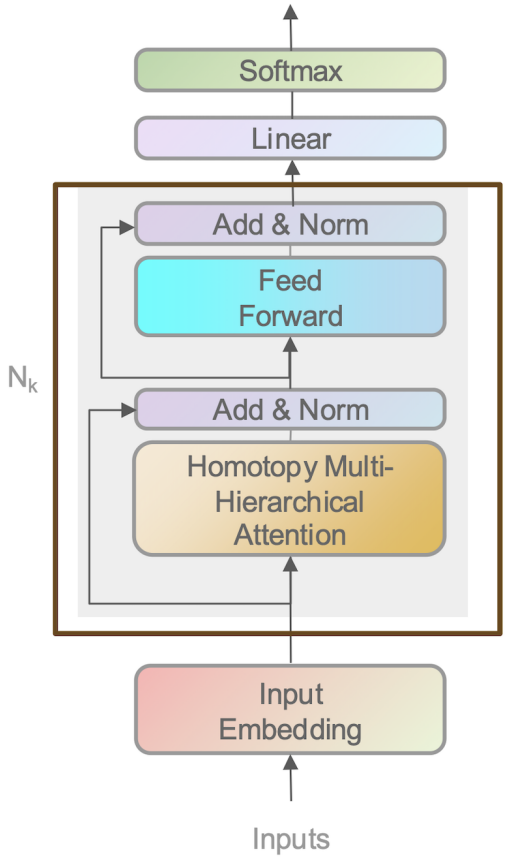

Figure 2. HMHERT-model architecture.

In this study, we investigate the model's generalization ability from one dataset to another. The dynamic systems of these two datasets are different. To ensure the model is not influenced by varying numerical magnitudes from different dynamical systems, we normalize all data to a range between 0 and 1.

3. Dynamical systems

In the course of this study, we consider classifying the data into two distinct categories: Chaotic data and non-chaotic data. Chaotic data refers to data that exhibits complex, unpredictable behavior within deterministic nonlinear dynamical systems. Such data types are often triggered by minute variations in the initial conditions within the system, observable even in fully deterministic frameworks. Conversely, non-chaotic data denote those data sets that do not exhibit the aforementioned chaotic characteristics and can be further subdivided into periodic data, quasi-periodic data, non-periodic data, and randomized data in this paper. Periodic data are characterized by a consistent pattern of repetition at fixed intervals; quasi-periodic data, while appearing periodic, vary in the length of their cycles over time; non-periodic data refers to data from continuous dynamical systems that lack any discernible repetitive patterns. Additionally, we consider completely random data to further increase the complexity of data classification tasks, aiming to evaluate the classification performance of neural networks. Szczęsna et al. [30] provided a series of data to validate the model's ability to classify chaos. The data types considered in this study are mostly derived from their work.

3.1. Periodic systems

We explore three types of oscillators within periodic systems: The undamped oscillator, the damped oscillator, and the rising oscillator. The undamped oscillator model is based on classical mechanics, particularly Newton's laws and Hamiltonian and Lagrangian mechanics, describing ideal oscillatory systems without friction or energy dissipation. This is represented by:

$$\begin{aligned} \dot{x} &= ay, \\ \dot{y} &= bx. \end{aligned} \tag{3.1}$$

The parameters a and b influence the system's frequency, period, and stability. If $ab > 0$, the system may become unstable; if $ab < 0$, it tends toward stability. We set $a = 0.5$ and $b = -0.5$, resulting in periodic and stable system. This system is denoted as OSC. Unlike other systems, the time interval here is set to 0.25 instead of 0.1. The system's trajectory is shown in Figure 3.

The damped oscillator accounts for friction or damping in real-world scenarios, characterized by decreasing amplitude over time until oscillation ceases. Its governing equations are:

$$\begin{aligned} \dot{x} &= y, \\ \dot{y} &= ax + by + c. \end{aligned} \tag{3.2}$$

Here, a affects stability and oscillatory behavior, b determines the rate of energy dissipation or gain (influencing damping or rising), and c represents a constant external force. When $b < 0$, the system is a damped oscillator; a smaller b means faster energy dissipation and a quicker decrease in amplitude. When $b > 0$, it becomes an rising oscillator, where amplitude increases under external forces.

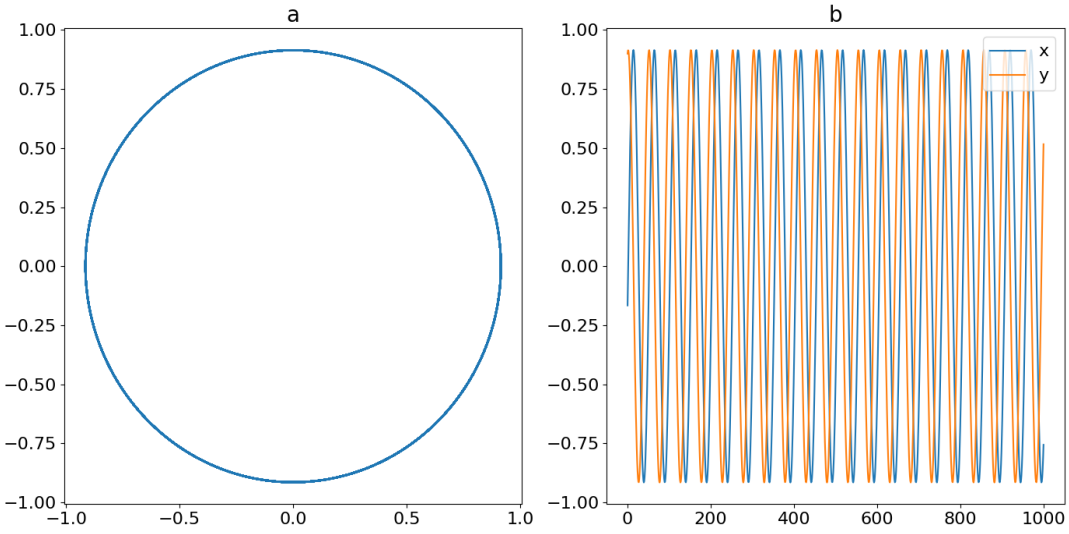

Figure 3. Trajectory of OSC: (a) phase-space plot, (b) time-domain plot.

For the damped oscillator (DOSC), parameters are set to $a = -0.2$, $b = -0.08$, and $c = 0.01$; its trajectory is illustrated in Figure 4. For the rising oscillator (IOSC), parameters are $a = -0.2$, $b = 0.08$, and $c = 0.01$; its trajectory is shown in Figure 5.

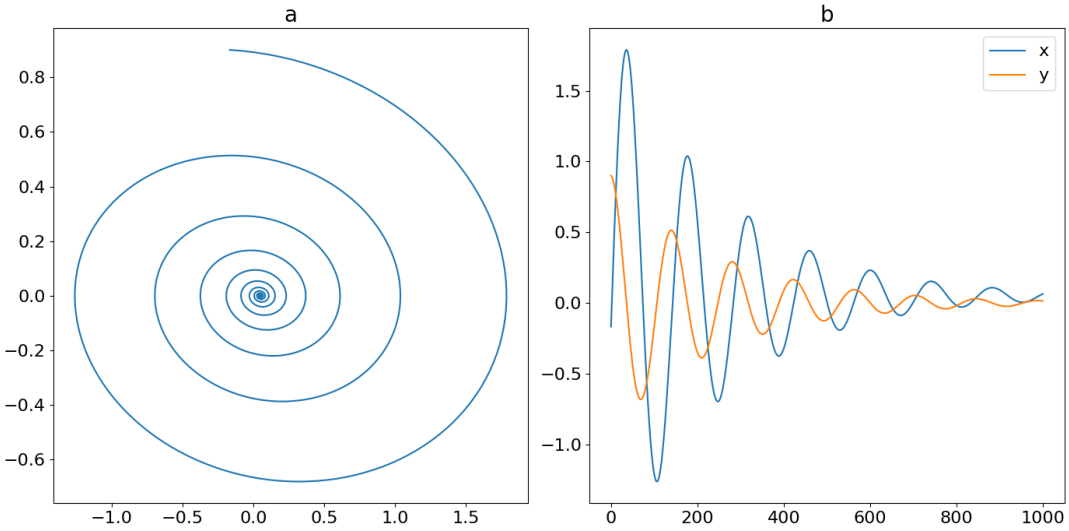

Figure 4. Trajectory of DOSC: (a) phase-space plot, (b) time-domain plot.

3.2. Quasi-periodic systems

Quasi-periodic systems exhibit behavior that closely approximates periodicity but never exactly repeats, leading to complex, non-repetitive patterns. The governing equations are:

$$x = \cos(a_1 t + x_0) + \cos(a_2 t + x_0), \tag{3.3}$$

$$x = \sin(a_1 t + x_0) + \cos(a_2 t + x_0). \tag{3.4}$$

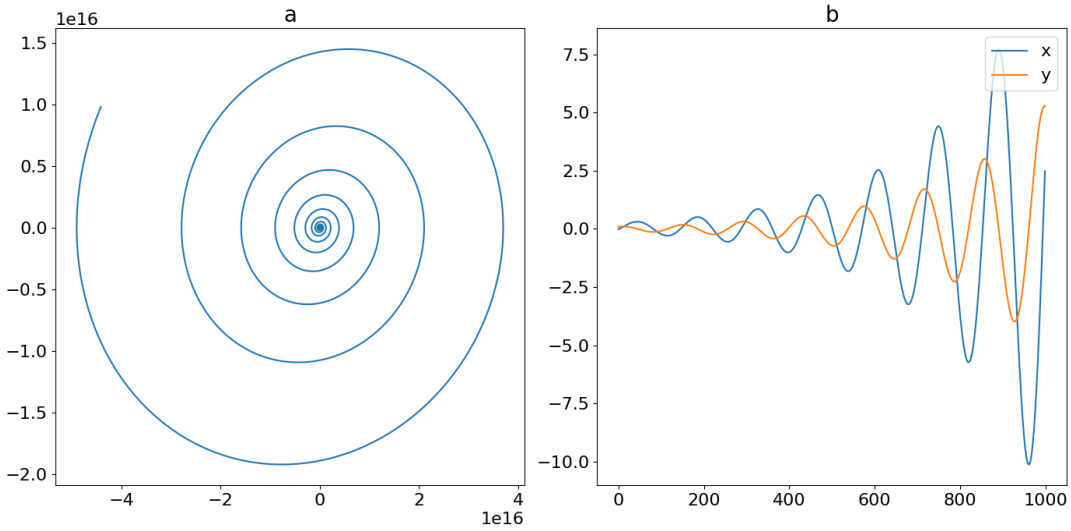

Figure 5. Trajectory of IOSC: (a) phase-space plot, (b) time-domain plot.

Parameters a_1 and a_2 determine the oscillation frequencies. When their ratio a_1/a_2 is irrational, the system exhibits increasingly complex dynamics due to interactions between different frequencies, preventing precise repetition over time.

We consider two quasi-periodic systems, Eq. (3.3) and Eq. (3.4), denoted as QPS_1 and QPS_2 respectively, where the initial condition is $x_0 = 2\pi$, and with parameters $(a_1, a_2) = (\pi/50, 1/50)$ and $((2+\sqrt{5})/30, 1/15)$ respectively. Their trajectories are shown in Figures 6 and 7.

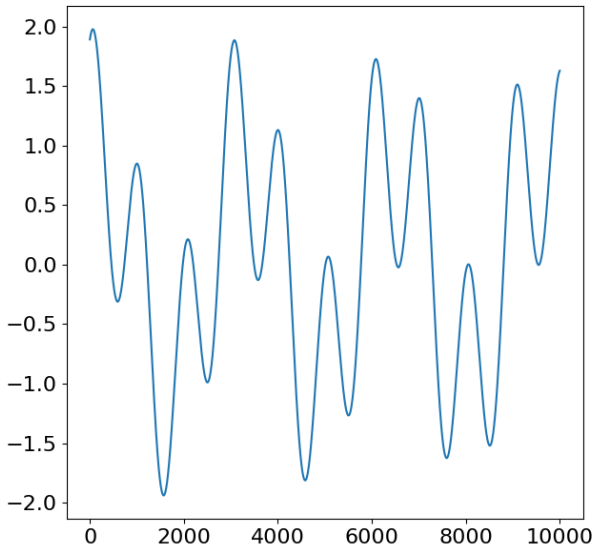

Figure 6. Trajectory of QPS_1: Time-domain plot.

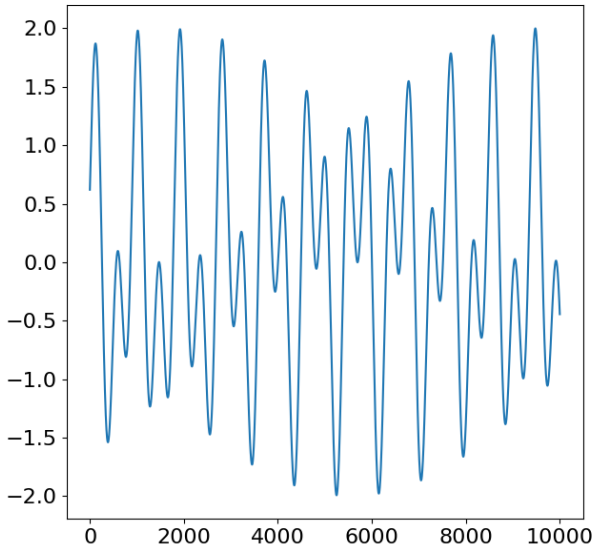

Figure 7. Trajectory of QPS_2: Time-domain plot.

3.3. Non-periodic systems

We consider a class of non-periodic, non-chaotic continuous dynamical systems with no periodic behavior or repeating patterns, even over long time scales. Over time, the system traces a space curve governed by:

$$\begin{aligned} \dot{x} &= a_1 x + a_2 y, \\ \dot{y} &= b_1 x + b_2 y, \\ \dot{z} &= c_1 x + c_2 z. \end{aligned} \tag{3.5}$$

We examine two such systems, denoted as DS_1 and DS_2, with model parameters listed in Table 1. Their trajectory diagrams are shown in Figures 8 and 9.

Table 1. Model parameters for DS_1 and DS_2.

	DS_1	DS_2
(a_1, a_2)	$(-0.01, 0.01)$	$(-0.05, 0.01)$
(b_1, b_2)	$(-0.001, -0.01)$	$(-0.001, -0.01)$
(c_1, c_2)	$(0.05, -0.03)$	$(0.05, -0.08)$

3.4. Chaotic systems

We explore chaotic dynamical systems that exhibit unpredictable behavior despite deterministic initial conditions and laws. Edward Lorenz's 1963 discovery of sensitive dependence on initial

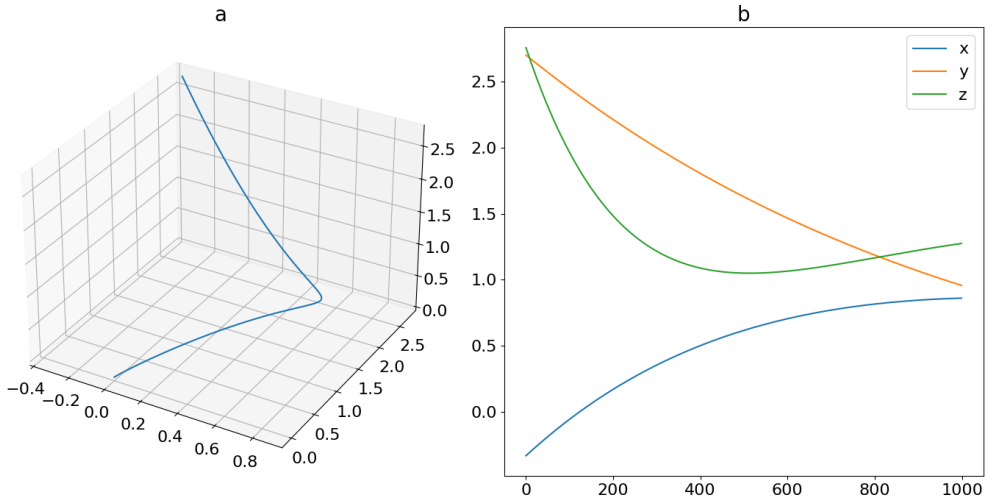

Figure 8. Trajectory of DS_1: (a) phase-space plot, (b) time-domain plot.

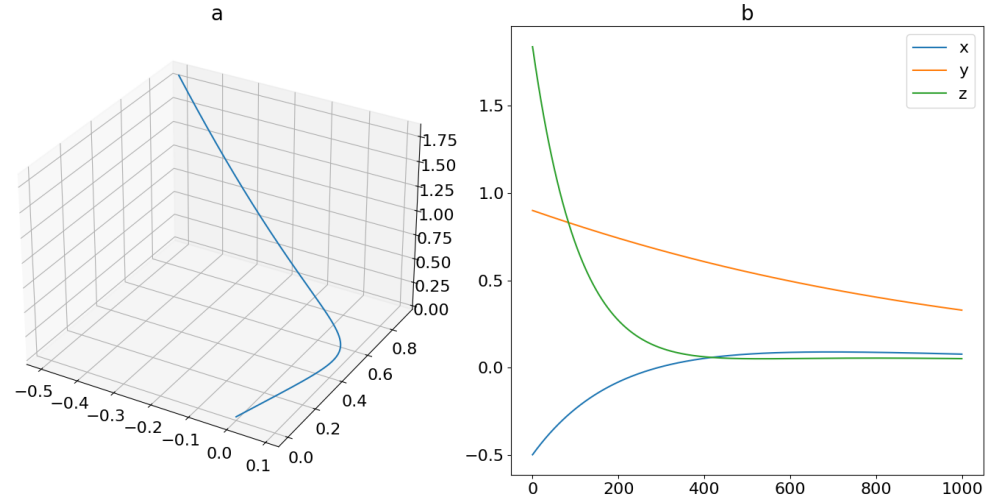

Figure 9. Trajectory of DS_2: (a) phase-space plot, (b) time-domain plot.

conditions in a simplified atmospheric convection model led to the "Lorenz Attractor", a milestone in chaos theory. The governing equations are:

$$\begin{aligned} \dot{x} &= -\sigma x + \sigma y, \\ \dot{y} &= \rho x - y - xz, \\ \dot{z} &= -\beta z + xy. \end{aligned} \tag{3.6}$$

In Lorenz's model, x, y, and z represent convective turnover rate, horizontal temperature variation, and vertical temperature variation, respectively, or coordinates in a three-dimensional phase space. Parameters σ (Prandtl number), ρ (Rayleigh number), and β define the system's dynamics. We use classical chaotic parameters $\sigma = 10$, $\rho = 28$, $\beta = 8/3$, making the Lorenz system classically chaotic, denoted as CHA_1. Its trajectory is shown in Figure 10.

The Rössler system, introduced by Otto Rössler in 1976, is a nonlinear system with a chaotic attractor exhibiting a twisted, disk-like structure. Its trajectories display fractal structures,

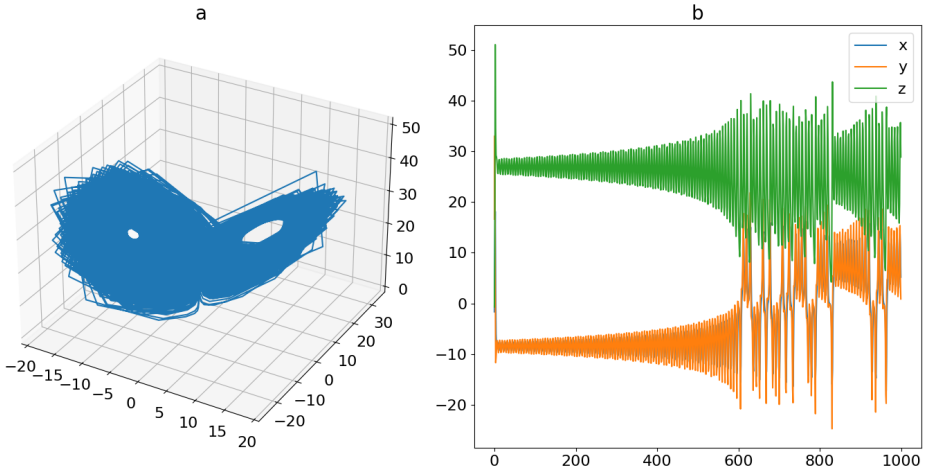

Figure 10. Trajectory of CHA_1: (a) phase-space plot, (b) time-domain plot.

making it a classic model for studying chaos. The governing equations are:

$$\begin{aligned} \dot{x} &= -y - z, \\ \dot{y} &= x + ay, \\ \dot{z} &= b + z(x - c). \end{aligned} \tag{3.7}$$

Parameters a, b, and c determine the system's dynamics: a affects feedback strength of y, b influences the baseline offset of z, and c controls nonlinear feedback of z. We set $a = 0.2$, $b = 0.2$, $c = 5.7$, defining CHA_2. Its trajectories are shown in Figure 11.

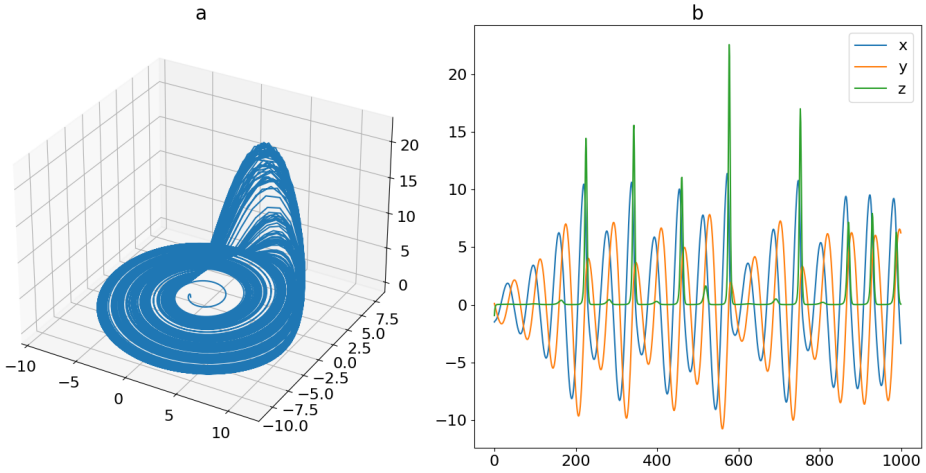

Figure 11. Trajectory of CHA_2: (a) phase-space plot, (b) time-domain plot.

The Rucklidge attractor, introduced by Ken Rucklidge during magnetohydrodynamics studies, exhibits complex behaviors including periodic windows and chaos. Its governing equations

are:

$$\begin{aligned} \dot{x} &= -kx + \lambda y - yz, \\ \dot{y} &= x, \\ \dot{z} &= -z + y^2. \end{aligned} \tag{3.8}$$

Parameter k acts as a damping coefficient for x, while λ controls linear coupling between x and y. We set $k = 2$, $\lambda = 6.7$, defining CHA_3. Its trajectories are shown in Figure 12.

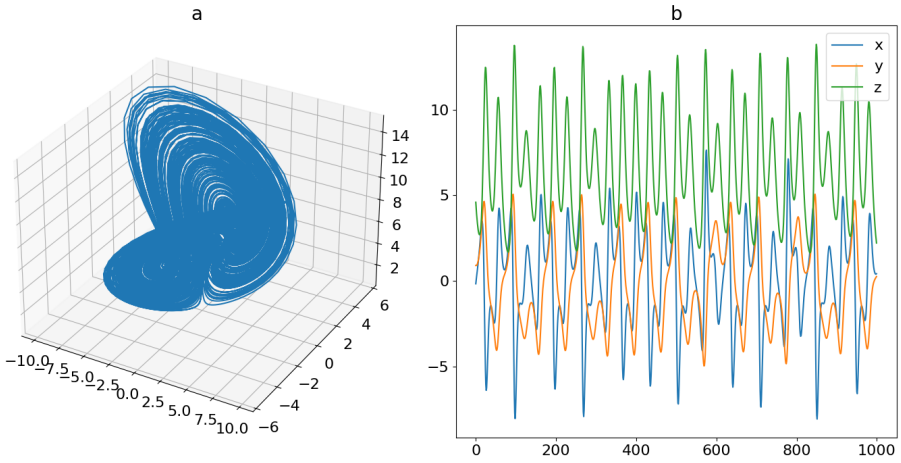

Figure 12. Trajectory of CHA_3: (a) phase-space plot, (b) time-domain plot.

We also consider pure random numbers, as both random and chaotic data can appear disordered and unpredictable, making them hard to distinguish. We focus on normally distributed random numbers, labeled Ran_1 and Ran_2, generated with different random seeds.

4. Experiments and results

In the subsequent phase, we will perform a series of experiments to ascertain the ability of the proposed method in accurately discriminating between signals exhibiting chaotic features. In this study, we compare the ability of HMHERT with other different methods to distinguish between chaotic and non-chaotic data. The non-chaotic data include periodic, quasi-periodic, non-periodic, and stochastic datasets.

The experimental data set is partitioned into three distinct subsets: The training set, the verification set, and the test set. The training set is employed for model training, the verification set aids in optimal model selection during the training process, and the test set assesses the model's accuracy. During the generalization ability test, we use the generalization test set to evaluate the generalization ability of all models. The generalization test set is also composed of periodic, quasi-periodic, non-periodic, random and chaotic data. Compared with the experimental data set, their governing equations are different.

In this work, we generate the experimental data set and the generalization test set using the Runge-Kutta method of simulation. The details of these diverse data types are presented in the third section, elaborating on the dataset's dynamic equations and parameters, along with their respective identifiers. The experimental data set consists of the chaotic data (CHA_2),

periodic data (DOSC), quasi-periodic data (QPS_1), non-periodic data (DS_1), and random data(Ran_1). Among them, 60% of the experimental dataset is classified as training set, 20% as validation set and 20% as testing set. Additional datasets, such as IOSC, QPS_2 and so on, serve as generalization test set to evaluate the generalization capabilities of the proposed model. We generated multiple sample sets, each containing 10,000 samples, for each dynamic system based on the information provided in Section 3. The initial conditions differ across sample sets. The number of features in each sample set corresponds to the number of variables in the respective dynamic system. In the experimental dataset, each dynamic system is represented by 20 sample sets, whereas the generalization test dataset comprises 200 sample sets for each dynamic system. This structured delineation not only provides clarity on the dataset's composition but also explains the methodology for testing and validating the model across diverse data scenarios. In the experiments, each sample set is partitioned into one-dimensional sequences of length 100 to serve as input to the neural network model. Because the number of features varies with the number of variables in each dynamic system, the resulting number of sequences differs across sample sets. The experimental data set comprises 20,000 samples, 30% of which consist of chaotic data, while the generalization test set contains 300,000 samples with chaotic instances accounting for 40% of the total collection.

To evaluate the performance of the proposed method, we consider accuracy as the assessment metric, defined as:

$$\text{Accuracy} = \frac{TC + TNC}{TC + TNC + FC + FNC}, \tag{4.1}$$

where TC is the number of chaotic samples correctly classified as chaotic, FC is the number of non-chaotic samples misclassified as chaotic, TNC is the number of non-chaotic samples correctly classified as non-chaotic, FNC is the number of chaotic samples misclassified as non-chaotic. This metric quantitatively measures the method's ability to accurately distinguish between chaotic and non-chaotic samples.

However, when confronted with distributional asymmetry between chaotic and non-chaotic classes in the dataset, employing classification accuracy as the primary performance metric becomes problematic because it tends to be expressed as classification performance of the model for overrepresented classes in the data. To mitigate this diagnostic limitation, we consider the F1-score and MCC respectively. Although the F1-score metric traditionally emphasizes classification performance for a single class, our experimental framework adopts a dual-perspective approach by evaluating separate F1-scores for both chaotic and non-chaotic categories, formally defined as follows:

$$\text{F1} = 2\frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{4.2}$$

where Precision represents the proportion of actually chaotic (or non-chaotic) samples predicted to be chaotic (or non-chaotic), Recall indicates the proportion predicted to be chaotic (or non-chaotic) in samples that are actually chaotic (or non-chaotic).

In contrast, MCC provides a balanced evaluation of classification performance by holistically incorporating predictive outcomes across both classes. Formally defined as:

$$\text{MCC} = \frac{TC * TNC - FC * FNC}{\sqrt{(TC + FC)(TC + FNC)(TNC + FC)(TNC + FNC)}}. \tag{4.3}$$

MCC yields values bounded within the interval $(-1, 1)$, where values approaching 1 indicate superior classification efficacy. In our experimental framework, model performance is rigorously

evaluated through a joint assessment of classification metrics derived from both the experimental data set and generalization test set, ensuring robustness validation across distinct data regimes.

4.1. Classification performance dependence on the effect of homotopy in multi-hierarchical multi-head attention mechanisms

In this section, we will evaluate the impact of the homotopy coefficient within the multi-hierarchical multi-head attention mechanism on the classification performance of the HMHERT. In order to more intuitively study impact of variations in the homotopy coefficient within the multi-hierarchical multi-head attention mechanism on model classification performance, we conducted an experiment where the homotopy coefficient was fixed prior to model training. For simplicity, the same homotopy coefficient was applied across all layers of the model, with the results presented in the Table 2.

During the training phase, we set the maximum number of training iterations to 8000 and the learning rate to 10^{-7}. The loss function is illustrated in Figure 13. We selected the best model based on its classification results on the validation set, and subsequently tested this model on the test set and the generalization test set.

As illustrated in Figure 13, the training loss function of the HMHERT decreases progressively with the number of training epochs, regardless of variations in the homotopy coefficients within the multi-hierarchical multi-head attention mechanism. In the later stages of training, the loss function continues to show a downward trend, indicating that the model still has potential for further optimization. A comparison of the training and validation loss function in the figure reveals that both decrease concurrently, indicating that the HMHERT performs well on both the training and validation datasets without exhibiting signs of overfitting.

Table 2. The classification results of the HMHERT under varying homotopy coefficients within the multi-hierarchical multi-head attention mechanism are presented, providing a detailed representation of its classification performance under different indicators.

(h_0, h_1)	(0.1,0.9)	(0.2,0.8)	(0.3,0.7)	(0.4,0.6)	(0.5,0.5)	(0.6,0.4)	(0.7,0.3)	(0.8,0.2)	(0.9,0.1)
$\text{Accuracy}_{\text{test}}$	97.83%	**98.00%**	97.75%	97.10%	97.75%	97.80%	97.18%	97.98%	96.48%
$F1_{\text{test, chaos}}$	0.9640	**0.9667**	0.9626	0.9515	0.9625	0.9633	0.9527	0.9665	0.9407
$F1_{\text{test, non-chaos}}$	0.9844	**0.9857**	0.9839	0.9793	0.9839	0.9843	0.9799	0.9855	0.9749
MCC_{test}	0.9485	**0.9525**	0.9465	0.9308	0.9465	0.9476	0.9326	0.9521	0.9157
$\text{Accuracy}_{\text{generalization}}$	95.02%	94.75%	**96.19%**	94.19%	95.92%	94.85%	95.62%	95.94%	95.30%
$F1_{\text{generalization, chaos}}$	0.9354	0.9320	**0.9517**	0.9250	0.9483	0.9337	0.9443	0.9482	0.9400
$F1_{\text{generalization, non-chaos}}$	0.9594	0.9573	**0.9685**	0.9526	0.9663	0.9579	0.9639	0.9665	0.9614
$MCC_{\text{generalization}}$	0.8966	0.8909	**0.9204**	0.8789	0.9148	0.8928	0.9085	0.9153	0.9020

As shown in the Table 2, regardless of the specific homotopy coefficient, the Transformer model with the homotopy multi-hierarchical multi-head attention mechanism consistently exhibited outstanding classification performance, with classification accuracy on the test set reaching approximately 98% in most cases, and generalization accuracy on the generalization test set exceeding 94%. MCC exhibited comparable trends to classification accuracy across experimental datasets. Through F1-score analysis, we observed that HMHERT demonstrated superior performance in classifying non-chaotic dynamics compared to chaotic dynamics. Parametric sensitivity studies revealed significant performance variations under different hyper-parameter configurations. Notably, on the test set, HMHERT achieved peak performance across all four classification metrics (accuracy, F1-score and MCC) at $h_0 = 0.2$, $h_1 = 0.8$, whereas optimal gen-

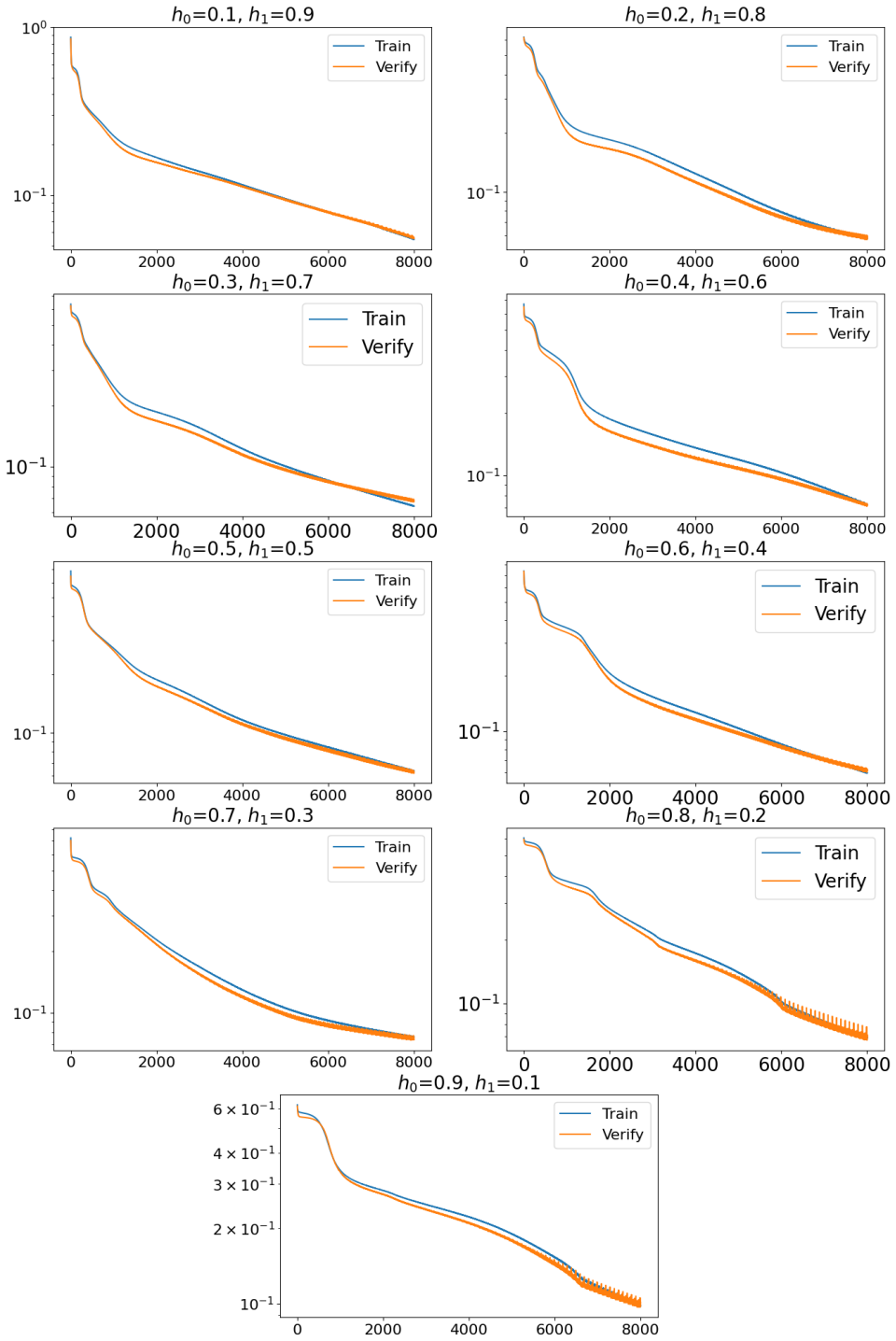

Figure 13. The training loss function and validation loss function results of the HMHERT's multi-hierarchical multi-head attention mechanism under different homotopy coefficients are presented. The x-axis represents the training epochs, while the y-axis denotes the value of the loss function. The y-axis is plotted on a logarithmic scale.

eralization performance on the generalization test set occurred at $h_0 = 0.3$, $h_1 = 0.7$. Conversely, configurations with $h_0 = 0.4$, $h_1 = 0.6$ and $h_0 = 0.9$, $h_1 = 0.1$ exhibited suboptimal classification efficacy in both test and generalization scenarios. These results suggest that the model is highly sensitive to variations in the homotopy coefficient within the multi-hierarchical multi-head attention mechanism, underscoring the importance of selecting an appropriate homotopy coefficient or optimization algorithm.

4.2. Comparison of neural networks

In the previous section, we examine the impact of homotopy coefficients within the multi-hierarchical multi-head attention mechanism on the HMHERT. In this section, we compare the neural networks mentioned in Section 3 with classical neural networks such as Time-Delayed RC, FCNN, LSTM, CNN by evaluating their classification performance in terms of Confusion Matrix, accuracy, F1-score and MCC. The classification results for the HMHERT are based on the outcomes in Table 2, specifically when $h_0 = 0.3$ and $h_1 = 0.7$. The training process follows the same methodology as outlined in the previous section. The results are presented in Table 3. The loss function is illustrated in Figure 14.

As shown in Figure 14, the training and validation loss functions for LSTM network, CNN, and the HMHERT decrease simultaneously. This indicating that these three neural networks perform well on both the training and validation sets without signs of overfitting. However, the training and validation loss functions for CNN exhibit oscillations in the later stages of training, suggesting that the learning rate for CNN might be slightly too high, and could benefit from a reduction in the later training stages. The loss function for LSTM network is higher than that of CNN and the HMHERT, implying that its classification accuracy is likely lower than CNN and the HMHERT, with CNN's accuracy expected to be slightly higher than the HMHERT's. For the FCNN, although both the training and validation loss functions decrease, there is a noticeable separation during the early and middle stages, with the validation loss being lower than the training loss. This suggests that the training set is more complex than the validation set for this model. In the later stages, the validation and training loss functions converge and stabilize, indicating that further training could lead to overfitting. Regarding the encoder of Transformer, although both the training and validation loss functions decrease, the training loss decreases at a much faster rate than the validation loss, suggesting the onset of overfitting.

As shown in Table 3, on the test dataset, the FCNN model exhibits the lowest classification accuracy at only 68.43% and MCC at only 0.0208. This means that FCNN's classification ability is not much different from random classification, suggesting that the model fails to learn meaningful discriminative patterns for the given task. In terms of the confusion matrix, FCNN misclassifies most chaotic signals as non-chaotic. Time-Delayed RC classifies all signals as non-chaotic. Although its accuracy is slightly higher than FCNN, its MCC is lower. The experimental results demonstrate that the LSTM network exhibits inferior classification performance compared to the CNN, the encoder of the Transformer, and the HMHERT model. Specifically, the classification accuracies of CNN, the encoder of Transformer, and HMHERT are 99.63%, 99.25%, and 97.75%, respectively, while their MCC values are 0.9911, 0.9823, and 0.9465, respectively. These metrics collectively indicate that CNN, the encoder of Transformer, and HMHERT achieve exceptional classification performance. Furthermore, the F1-score analysis reveals that these three models exhibit comparable classification efficacy in distinguishing between chaotic and non-chaotic categories.

However, on the generalization test dataset, only Transformer-based models perform well,

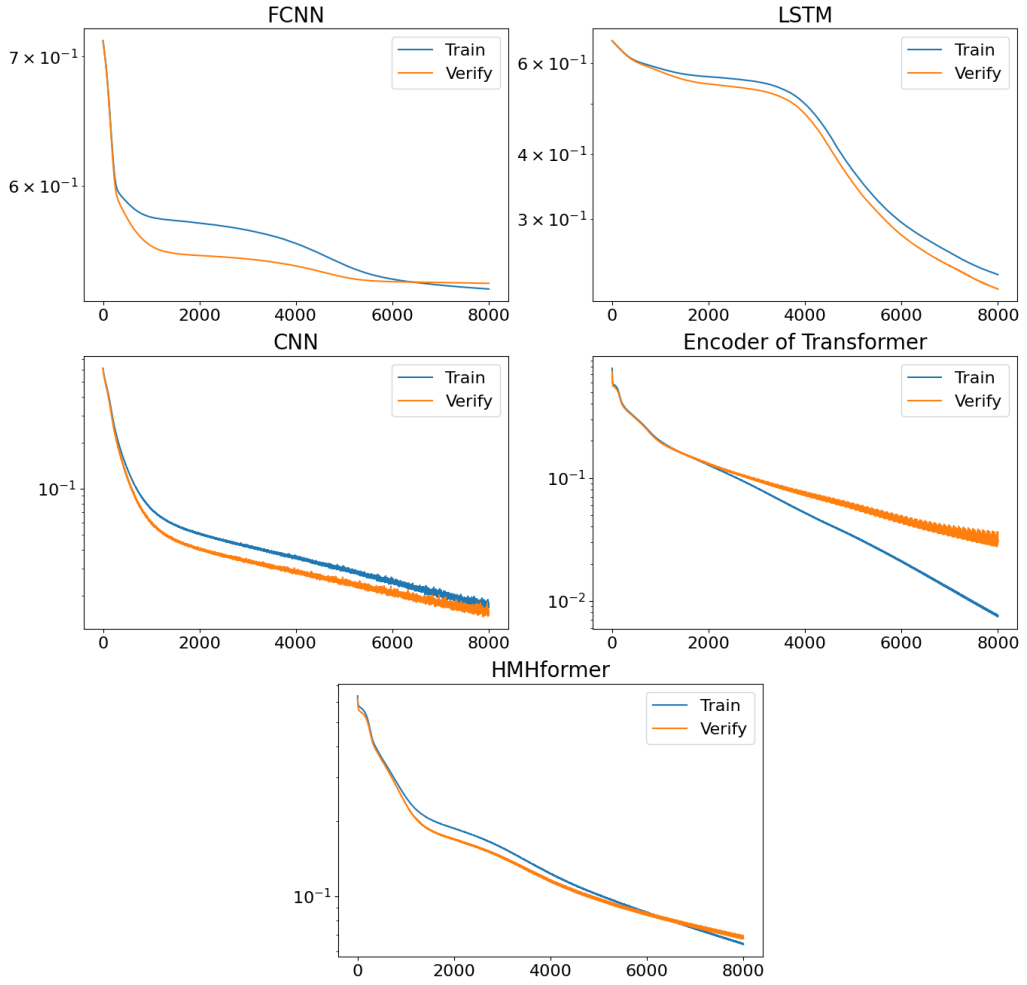

Figure 14. The training loss function and validation loss function results of different neural networks are presented (Time-Delayed RC is different from other neural network training methods, so there is no loss function diagram). The x-axis represents the training epochs, while the y-axis denotes the value of the loss function. The y-axis is plotted on a logarithmic scale.

Time-Delayed RC, FCNN, LSTM network, and CNN model achieves generalization classification accuracies no greater than 72%. Furthermore, their MCC values are below 0.42, with the Time-Delayed RC exhibiting an exceptionally low MCC of only 0.0000. These findings collectively demonstrate that these three models are inadequate for generalization classification tasks, as they fail to capture meaningful discriminative patterns required for robust performance.

In summary, among the FCNN, LSTM network, and CNN models, the CNN demonstrates good classification performance on the data it has been trained on but lacks generalization capability. The LSTM network has weaker classification performance compared to the CNN, and the FCNN performs poorly in both respects. We also confirmed that adjusting the learning rate between $1e^{-7}$ and $1e^{-4}$ does not improve the classification performance or generalization classification performance of the FCNN, LSTM network, and CNN models.

Both encoder of Transformer and HMHERT models show excellent classification performance on the test dataset, comparable to the CNN, but they significantly outperform the CNN in terms

Table 3. The classification results of different neural networks are presented, providing a detailed representation of its classification performance under different indicators.

	Time-Delayed RC	FCNN	LSTM	CNN	Encoder of Transformer	HMHERT
TNC_{test}	2800	2669	2725	2798	2773	2753
FC_{test}	0	131	75	2	27	47
TC_{test}	0	68	771	1187	1197	1157
FNC_{test}	1200	1132	429	13	3	43
$\text{Accuracy}_{\text{test}}$	70.00%	68.43%	87.40%	**99.63%**	99.25%	97.75%
$F1_{\text{test, chaos}}$	0.0000	0.0972	0.7537	**0.9937**	0.9876	0.9626
$F1_{\text{test, non-chaos}}$	0.8235	0.8087	0.9154	**0.9973**	0.9946	0.9839
MCC_{test}	0.0000	0.0208	0.6909	**0.9911**	0.9823	0.9465
$TNC_{\text{generalization}}$	180000	175649	134594	139474	178003	175732
$FC_{\text{generalization}}$	0	4351	45406	40526	1997	4268
$TC_{\text{generalization}}$	0	3278	79886	48188	100573	112828
$FNC_{\text{generalization}}$	120000	116722	40114	71812	19427	7172
$\text{Accuracy}_{\text{generalization}}$	60.00%	59.64%	71.49%	62.55%	92.86%	**96.19%**
$F1_{\text{generalization, chaos}}$	0.0000	0.0514	0.6514	0.4618	0.9037	**0.9517**
$F1_{\text{generalization, non-chaos}}$	0.7500	0.7437	0.7589	0.7129	0.9432	**0.9685**
$MCC_{\text{generalization}}$	0.0000	0.0098	0.4107	0.1894	0.8541	**0.9204**

of generalization capability, with accuracies of 92.86% and 96.19%, respectively. This indicates that these models are not only capable of accurately classifying training data but also possess strong generalization abilities. Between encoder of Transformer and HMHERT, the latter demonstrates superior generalization ability, mainly. The evaluation on the generalization test set demonstrates that HMHERT achieves a 3.33% improvement in classification accuracy and a 46.64% reduction in classification error rate compared to the encoder of the Transformer. Furthermore, HMHERT exhibits a 0.0663 increase in MCC relative to the Transformer encoder, highlighting its enhanced capability to maintain robust classification performance under cross-domain evaluation conditions. These findings suggest that the homotopy multi-hierarchical multi-head attention mechanism significantly enhances the generalization classification performance of the Transformer encoder.

4.3. Research on the relationship in the multi-hierarchical mechanism of Transformer

In neural networks, parameters are typically learned from the dataset during training. In previous experiments, when exploring the homotopy coefficients in the multi-hierarchical multi-head attention mechanism of the HMHERT, the homotopy parameters were directly specified. In the following experiments, we further investigate the multi-hierarchical multi-head attention mechanism of the HMHERT. In Experiment 1, the multi-hierarchical multi-head attention mechanism of the HMHERT remains based on homotopy, the coefficients are randomly assigned. In Experiment 2, the multi-hierarchical multi-head attention mechanism is still based on homotopy, but the homotopy coefficients are learned from the data. In Experiment 3, the multi-hierarchical multi-head attention mechanism is unrestricted, and the homotopy coefficients are fully learned from the data. The results of these three experiments are compared with those of the baseline experiment, in which the homotopy coefficients were directly specified. The experimental results are shown in Table 4, and the training and validation loss functions are depicted in Figure 15.

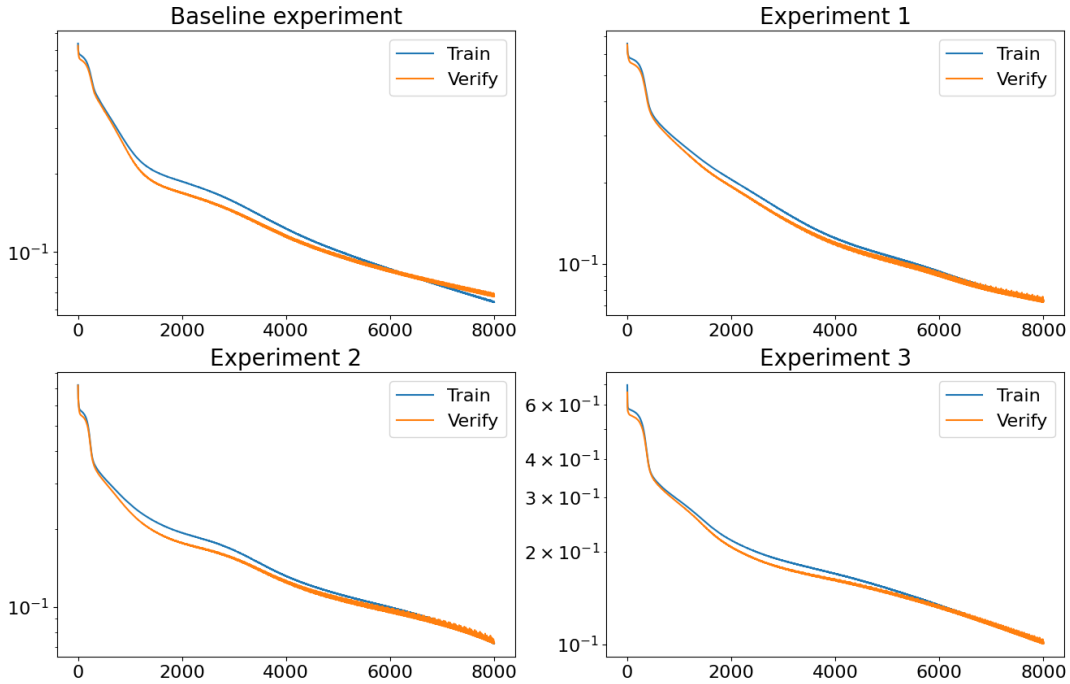

Figure 15. The results of the training and validation loss functions for the HMHERT under different training methods are presented. Additionally, the training and validation loss function results when the multi-hierarchical multi-head attention mechanism is unrestricted are also provided. The x-axis represents the training epochs, while the y-axis denotes the value of the loss function. The y-axis is plotted on a logarithmic scale.

As shown in Figure 15, the loss function curves for the four experiments are generally similar, with no evidence of overfitting. The primary difference lies in the rate of decrease in the loss functions. However, Experiment 3 shows the smallest decline, while Experiments 1 and 2 exhibit similar rates of decrease. The baseline experiment demonstrates the largest reduction in the loss function, indicating that it fits the training data well while maintaining strong performance on the validation set.

Table 4. The classification results of different experiments are presented, providing a detailed representation of its classification performance under different indicators.

	Baseline experiment	Experiment 1	Experiment 2	Experiment 3
$\text{Accuracy}_{\text{test}}$	**97.75**%	96.88%	96.78%	95.58%
$F1_{\text{test, chaos}}$	**0.9626**	0.9476	0.9455	0.9264
$F1_{\text{test, non-chaos}}$	**0.9839**	0.9777	0.9771	0.9684
MCC_{test}	**0.9465**	0.9254	0.9228	0.8948
$\text{Accuracy}_{\text{generalization}}$	**96.19**%	95.93%	95.78%	94.53%
$F1_{\text{generalization, chaos}}$	**0.9517**	0.9483	0.9457	0.9304
$F1_{\text{generalization, non-chaos}}$	**0.9685**	0.9664	0.9654	0.9549
$MCC_{\text{generalization}}$	**0.9204**	0.9150	0.9122	0.8857

As shown in 4, the baseline experiment achieves the highest classification accuracy and MCC on the test set, followed by Experiments 1 and 2, with Experiment 3 having the lowest accu-

racy and MCC, consistent with the observations from Figure 15. In the generalization test, the classification accuracy and MCC of the baseline experiment and Experiment 1 are comparable. However, the coefficients in the baseline experiment were determined through extensive testing, and as the number of attention layers in the HMHERT increases, directly specifying these homotopy coefficients becomes increasingly challenging. In Experiment 3, the classification accuracy on the test set falls below 96%, and its generalization accuracy is also lower than that of the other experiments. Notably, in both the test set and generalization test set, a result that stands in stark contrast to the performance of its counterparts in other experimental configurations. Additionally, Experiment 3 is more prone to encountering "NaN" results during training. These findings highlight the critical role that homotopy play in the multi-hierarchical multi-head attention mechanism.

5. Discussion

In this study, we introduced the HMHERT model, derived from the homotopy multi-hierarchical multi-head attention mechanism based on homotopy theory, to enhance the performance of deep learning models in complex classification tasks. By integrating the principles of homotopy theory into the model's architecture, we aimed to investigate its influence on model generalization and classification performance, particularly in the context of chaotic data distributions.

Through a series of three meticulously designed experiments, we demonstrated the superiority of the HMHERT model over traditional neural network architectures. Experimental results indicate that the generalization ability of the model in chaotic classification tasks is highly sensitive to the homotopy coefficient, achieving optimal performance when the coefficient is set to $(0.3, 0.7)$. When compared to traditional neural network models such as Time-Delayed RC, FCNN, LSTMs, CNNs, and Transformer encoders, the HMHERT model demonstrated slightly lower test classification performance than CNNs and Transformer encoders. However, HMHERT significantly outperformed all other models in terms of generalization classification performance. A variant of the HMHERT model, which retained its multi-hierarchical structure but removed the homotopy constraint, exhibited substantially weaker performance in classification chaos, underscoring the importance of both the homotopy constraint and the multi-hierarchical structure in enhancing the classification capabilities of the Transformer encoder for chaotic data.

Further investigation into the selection of homotopy parameters revealed that models using randomly assigned or data-trained parameters performed similarly, albeit slightly below the optimal model with directly specified parameters. This finding suggests new directions for training deeper models.

These experimental results demonstrate that homotopy theory provides a robust theoretical foundation for improving the design and generalization capabilities of deep learning models. Specifically, the HMHERT model's enhanced performance in generalization tasks highlights the practical utility of integrating homotopy principles into neural architectures. Future research could focus on optimizing the homotopy coefficient and adapting the Homotopy Multi-Hierarchical multi-head attention mechanism to diverse neural networks, broadening its applicability. Additionally, a deeper theoretical exploration into the mechanisms by which homotopy theory enhances model generalization could yield valuable insights to guide the development of next-generation neural networks.

In conclusion, this study underscores the potential of homotopy theory to advance deep learning research by offering a novel theoretical perspective for neural network design. By continuing

to investigate this approach, researchers may develop more robust and efficient models capable of addressing challenges in increasingly complex domains.

Acknowledgments

The authors extend their sincere gratitude to Professor Yong Li for his guidance and numerous invaluable suggestions.

Data availability statement

The datasets generated and analyzed during the current study, including both chaotic and non-chaotic data, are publicly available on Zenodo at https://zenodo.org/doi/10.5281/zenodo.14685513. This repository provides all necessary data and scripts to reproduce the findings of this study.

References

[1] S. J. Anagnostopoulos, J. D. Toscano, N. Stergiopulos and G. E. Karniadakis, *Residual-based attention in physics-informed neural networks*, Computer Methods in Applied Mechanics and Engineering, 2024, 421, 116805.

[2] A. Balestrino, A. Caiti and E. Crisostomi, *Generalised entropy of curves for the analysis and classification of dynamical systems*, Entropy, 2009, 11(2), 249–270.

[3] N. Boullé, V. Dallas, Y. Nakatsukasa and D. Samaddar, *Classification of chaotic time series with deep learning*, Physica D: Nonlinear Phenomena, 2020, 403, 132261.

[4] J. S. Cánovas, *Topological sequence entropy of interval maps*, Nonlinearity, 2003, 17(1), 49.

[5] T. L. Carroll, *Using reservoir computers to distinguish chaotic signals*, Physical Review E, 2018, 98(5), 052209.

[6] C. F. R. Chen, Q. Fan and R. Panda, *Crossvit: Cross-attention multi-scale vision transformer for image classification*, in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, 357–366.

[7] W. Chen and K. Shi, *Multi-scale attention convolutional neural network for time series classification*, Neural Networks, 2021, 136, 126–140.

[8] Z. Cui, Q. Li, Z. Cao and N. Liu, *Dense attention pyramid networks for multi-scale ship detection in sar images*, IEEE Transactions on Geoscience and Remote Sensing, 2019, 57(11), 8983–8997.

[9] J. Devlin, *Bert: Pre-training of deep bidirectional transformers for language understanding*, arXiv preprint, 2018. arXiv:1810.04805.

[10] H. Fan, B. Xiong, K. Mangalam, et al., *Multiscale vision transformers*, in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, 6824–6835.

[11] B. Feng and X. Zhou, *Energy-informed graph transformer model for solid mechanical analyses*, Communications in Nonlinear Science and Numerical Simulation, 2024, 108103.

[12] N. Geneva and N. Zabaras, *Transformers for modeling physical systems*, Neural Networks, 2022, 146, 272–289.

[13] D. Han, T. Ye, Y. Han, et al., *Agent attention: On the integration of softmax and linear attention*, in European Conference on Computer Vision, Springer, 2024, 124–140.

[14] J. Hao and W. Zhu, *Architecture self-attention mechanism: Nonlinear optimization for neural architecture search*, J. Nonlinear Var. Anal, 2021, 5, 119–140.

[15] A. Hemmasian and A. B. Farimani, *Multi-scale time-stepping of partial differential equations with transformers*, Computer Methods in Applied Mechanics and Engineering, 2024, 426, 116983.

[16] Y. Li and Y. Li, *A homotopy gated recurrent unit for predicting high dimensional hyperchaos*, Communications in Nonlinear Science and Numerical Simulation, 2022, 115, 106716.

[17] Y. Li and Y. Li, *Predicting chaotic time series and replicating chaotic attractors based on two novel echo state network models*, Neurocomputing, 2022, 491, 321–332.

[18] D. W. Liedji, J. H. Talla Mbé and G. Kenné, *Chaos recognition using a single nonlinear node delay-based reservoir computer*, The European Physical Journal B, 2022, 95(1), 18.

[19] D. Wenkack Liedji, J. H. Talla Mbé and G. Kenne, *Classification of hyperchaotic, chaotic, and regular signals using single nonlinear node delay-based reservoir computers*, Chaos: An Interdisciplinary Journal of Nonlinear Science, 2022, 32(12).

[20] B. Lim, S. Ö. Arık, N. Loeff and T. Pfister, *Temporal fusion transformers for interpretable multi-horizon time series forecasting*, International Journal of Forecasting, 2021, 37(4), 1748–1764.

[21] Z. Liu, Y. Lin, Y. Cao, et al., *Swin transformer: Hierarchical vision transformer using shifted windows*, in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, 10012–10022.

[22] E. N. Lorenz, *Deterministic nonperiodic flow*, Journal of Atmospheric Sciences, 1963, 20(2), 130–141.

[23] H. Miao, W. Zhu, Y. Dan and N. Yu, *Chaotic time series prediction based on multi-scale attention in a multi-agent environment*, Chaos, Solitons & Fractals, 2024, 183, 114875.

[24] S. Mukhopadhyay and S. Banerjee, *Learning dynamical systems in noise using convolutional neural networks*, Chaos: An Interdisciplinary Journal of Nonlinear Science, 2020, 30(10).

[25] Y. Pan, T. Yao, Y. Li and T. Mei, *X-linear attention networks for image captioning*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, 10971–10980.

[26] A. Radford, *Improving language understanding by generative pre-training*, 2018.

[27] A. Sinha and J. Dolz, *Multi-scale self-guided attention for medical image segmentation*, IEEE Journal of Biomedical and Health Informatics, 2020, 25(1), 121–130.

[28] J. Su, H. Li, R. Wang, et al., *A hybrid dual-branch model with recurrence plots and transposed transformer for stock trend prediction*, Chaos: An Interdisciplinary Journal of Nonlinear Science, 2025, 35(1).

[29] S. Sun, W. Ren, X. Gao, et al., *Restoring images in adverse weather conditions via histogram transformer*, in European Conference on Computer Vision, Springer, 2024, 111–129.

[30] A. Szczęsna, D. Augustyn, K. Harężlak, et al., *Datasets for learning of unknown characteristics of dynamical systems*, Scientific Data, 2023, 10(1), 79.

[31] A. Vaswani, N. Shazeer, N. Parmar, et al., *Attention is All You Need*, Advances in Neural Information Processing Systems, 2017.

[32] Y. Xiong, W. Yang, H. Liao, et al., *Soft variable selection combining partial least squares and attention mechanism for multivariable calibration*, Chemometrics and Intelligent Laboratory Systems, 2022, 223, 104532.

[33] S. Yun and Y. Ro, *Shvit: Single-head vision transformer with memory efficient macro design*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, 5756–5767.

[34] 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, 2023, 37, 11121–11128.

[35] C. Zhao, J. Ye, Z. Zhu and Y. Huang, *Flrnn-fga: Fractional-order lipschitz recurrent neural network with frequency-domain gated attention mechanism for time series forecasting*, Fractal and Fractional, 2024, 8(7), 433.

[36] L. Zhornyak, M. A. Hsieh and E. Forgoston, *Inferring bifurcation diagrams with transformers*, Chaos: An Interdisciplinary Journal of Nonlinear Science, 2024, 34(5).

[37] H. Zhou, S. Zhang, J. Peng, et al., *Informer: Beyond efficient transformer for long sequence time-series forecasting*, in Proceedings of the AAAI Conference on Artificial Intelligence, 2021, 35, 11106–11115.

[38] L. Zhu, X. Wang, Z. Ke, et al., *Biformer: Vision transformer with bi-level routing attention*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, 10323–10333.

Received March 2025; Accepted July 2025; Available online August 2025.