International Journal of Innovative Computing, Information and Control
Volume **21**, Number **6**, December **2025**

 ISSN 1349-4198
pp. **1557–1568**

A MULTIVARIATE TIME SERIES CLASSIFICATION APPROACH BASED ON FEATURE REPRESENTATION AND HIERARCHICAL NETWORK

Degang Wang[1,*], Jiani Tang[1], Wenyan Song[2] and Yuchen Jin[1]

[1]School of Control Science and Engineering
Dalian University of Technology
No. 2, Linggong Road, Ganjingzi District, Dalian 116024, P. R. China
tangjiani0913@163.com; 156152251@qq.com
*Corresponding author: wangdg@dlut.edu.cn

[2]School of Economics
Dongbei University of Finance and Economics
No. 217, Jianshan Street, Shahekou District, Dalian 116025, P. R. China
songwydufe@dufe.edu.cn

Received March 2025; revised July 2025

Abstract. *Multivariate time series classification has been widely utilized in modeling and analyzing of complex system. In this paper, a multivariate time series classification method based on feature representation and a hierarchical network is established. First, a representation model that integrates local, global and time-series-dependent features is constructed, and a joint loss function is designed to guide model optimization in extracting discriminative features with enhanced class separability and intrinsic specificity. Then, a multi-scale feature dataset is generated based on similarity metrics, and the extreme gradient boosting decision tree is employed to identify the importance of features according to different data scales. Further, a hierarchical network that integrated multi-level feature information is proposed for classification. The area under the receiver operating characteristic curve (AUROC) and the area under the precision-recall curve (AUPRC) are considered to measure the performance on a public dataset. Compared with some baseline models, the proposed method obtains superior performances in both AUROC and AUPRC.*

Keywords: Multivariate time series classification, Feature representation, Hierarchical network

1. **Introduction.** In the era of big data, multivariate time series data is widely available in many fields. By performing classification tasks on multivariate time series, the potential patterns embedded in the data can be revealed which can help to understand the evolution laws of complex systems.

Some researchers focus on learning representative subsequences which are important for time series classification. In [1,2], the Shapelet-based candidate feature representation method is proposed for handling the data classification. In order to obtain more distinguishable representation features, contrastive schemes are chosen to conduct representation learning of multivariate time series data. In [3], the temporal and contextual contrastive modules are considered to handle the features of time series. In [4], hierarchical contrastive method is utilized for learning feature. Due to the mixed information contained in time series data, some attention mechanisms are applied in capturing complex features. In [5], a time attention mechanism is applied to obtaining feature. In [6],

DOI: 10.24507/ijicic.21.06.1557

a self-attention model is provided to model time series. To synergize diverse network architectures, some attention modules are combined to capture the feature information. In [7], a kind of multi-channel model with time-aware attention embedding layers is designed to learn the temporal features. A bidirectional gated recurrent unit (GRU) architecture with multi-head self-attention module [8] is proposed to simultaneously extract local and global temporal patterns. Current research demonstrates that effective feature representation plays a critical role in multivariate time series classification. Due to the high feature dimensionality of multivariate time series and the highly coupled correlations within the data, how to fully utilize these multi-modal information and distribution characteristics to mine key features is still a question which needs further exploration.

Motivated by these facts, the aim of this paper is to establish an information fusion model to extract multi-scale features for improving the classification performance of time series. The main contributions of this paper are as follows.

1) A multi-modal feature fusion framework integrating residual dilated-convolution and GRU is proposed to simultaneously capture local, global, and temporal dependencies, effectively enhancing class separability while preserving intrinsic specificity of time series.

2) An ensemble hierarchical broad network with multi-scale feature fusion is proposed, which can improve classification precision by identifying feature importance in different data scales and integrating different scale prediction results.

The paper is organized as follows. In Section 2, a feature representation method for multivariate time series is established. In Section 3, an ensemble hierarchical broad network is constructed and the corresponding parameter learning method is designed. Some numerical examples are provided to demonstrate effectiveness of the model in Section 4. In Section 5, some conclusions are summarized.

2. Feature Representation Based on Multi-Modal Feature Fusion.

This section presents a temporal data representation model that incorporates multi-modal features.

2.1. Model construction by fusion of multi-modal features.

Given a collection of multivariate time series consisting of N samples $\boldsymbol{\mathcal{X}} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N\}$, where $\boldsymbol{X}_i = [\boldsymbol{x}_{i,1}, \ldots, \boldsymbol{x}_{i,T}]^T \in \mathbb{R}^{T\times L}$ is the i-th sample, $i = 1, \ldots, N$, T is the length of the time series and L is the number of features at each time step. $\boldsymbol{x}_{i,t} = [x_{i,t,1}, \ldots, x_{i,t,L}]$ is the feature vector at time step t, $t = 1, \ldots, T$, where $x_{i,t,l}$ is the l-th feature, $l = 1, \ldots, L$. The label dataset is $\boldsymbol{Y} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N\}$, where $\boldsymbol{y}_i = [y_{i,1}, \ldots, y_{i,C}]$ is the label data, and C is the number of categories.

The overall structure of the model is shown in Figure 1.

Firstly, the random cropping operation [4] is used to augment the data. For each multivariate time series sample $\boldsymbol{X}_i$, it is subjected to P random cropping operations along the time dimension to obtain P augmented samples with different starting and ending time points. The time window for cropping is defined as $[t_{S,1}, t_{E,1}], \ldots, [t_{S,P}, t_{E,P}]$ which satisfies $0 < t_{S,1} \leq \cdots \leq t_{S,P} \leq t_{E,1} \leq \cdots \leq t_{E,P} \leq T$, where the length of the j-th time window is $T_j = t_{E,j} - t_{S,j} + 1 \geq S_t$, $j = 1, \ldots, P$. The j-th augmented sample of $\boldsymbol{X}_i$ is denoted as $\boldsymbol{X}_{i,j} = [\boldsymbol{X}_{i,j,1}, \ldots, \boldsymbol{X}_{i,j,T_j}]^T$, where $\boldsymbol{X}_{i,j,t} = [x_{i,j,t,1}, \ldots, x_{i,j,t,L}]$. A set of augmented samples $\boldsymbol{\mathcal{X}}_P = \{\boldsymbol{X}_{1,1}, \ldots, \boldsymbol{X}_{1,P}, \ldots, \boldsymbol{X}_{N,1}, \ldots, \boldsymbol{X}_{N,P}\}$ consisting of $N \times P$ augmented samples can be obtained.

The first layer of the time-dependent feature-focused representation learning model (TD-RLM) is the input layer, where $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T]^T$ is the input value, T is the number of time steps, $\boldsymbol{x}_t$ is the feature vector at time t, and L is the number of features for each time step.

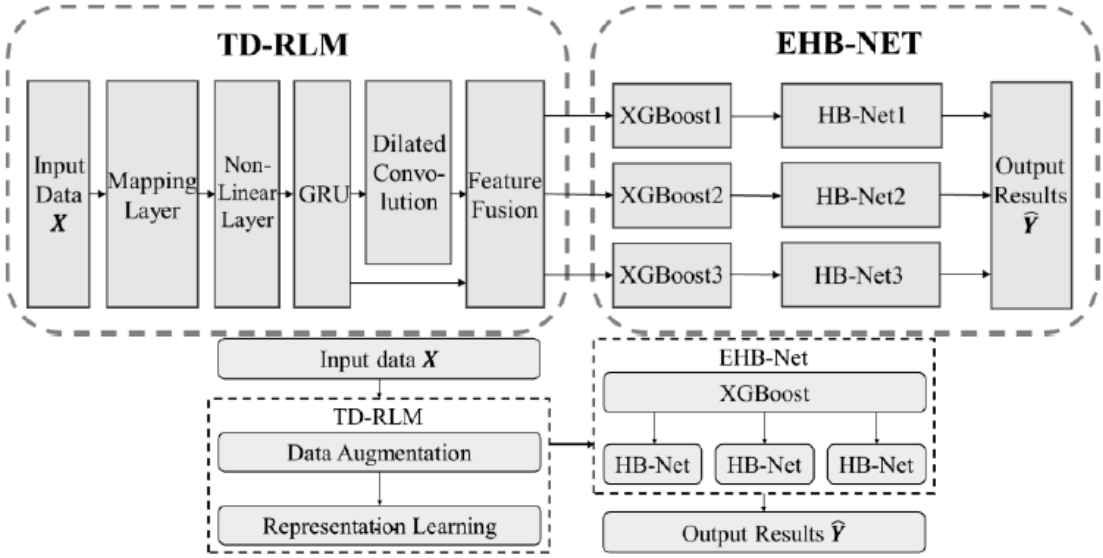

FIGURE 1. Model workflow diagram

The second layer is the mapping layer. The feature vector $\boldsymbol{x}_t$ is mapped onto the potential vector $\widetilde{\boldsymbol{x}}_t$, i.e., $\widetilde{\boldsymbol{x}}_t = \boldsymbol{x}_t \boldsymbol{W}_{I,1} + \boldsymbol{b}_{I,1}$, where $\boldsymbol{W}_{I,1}$ is the weight parameter and $\boldsymbol{b}_{I,1}$ is the bias vector. The output data of this layer is denoted as $\widetilde{\boldsymbol{X}} = [\widetilde{\boldsymbol{x}}_1, \ldots, \widetilde{\boldsymbol{x}}_T]^T$.

The third layer is the non-linear layer, where $\widehat{\boldsymbol{X}} = [\widehat{\boldsymbol{x}}_1, \ldots, \widehat{\boldsymbol{x}}_T]^T$ is obtained by non-linear transformation of $\widetilde{\boldsymbol{X}}$ using the GELU activation function: $\widehat{\boldsymbol{X}} = \phi_{GELU}\left(\widetilde{\boldsymbol{X}}\right)$, where $\phi_{GELU}(x) = 0.5x\left(1 + \phi_{\tanh}\left(\sqrt{2/\pi}\left(x + 0.044715x^3\right)\right)\right)$ and $\phi_{\tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$.

Further, $\widehat{\boldsymbol{X}}$ is fed into the GRU module to extract the time-dependent feature of the data, which consists of N_G GRU layers and $N_G - 1$ dropout layers.

In the first layer of the GRU module, there are two gated units, the reset gate $\boldsymbol{R}_t^{(1)}$ and the update gate $\boldsymbol{U}_t^{(1)}$. These two units are used to control how much of the previous moment's hidden state information is retained. They are computed by

$$\boldsymbol{R}_t^{(1)} = \phi_{sigmoid}\left(\widehat{\boldsymbol{x}}_t \boldsymbol{W}_{R,1}^{(1)} + \boldsymbol{h}_{t-1}^{(1)} \boldsymbol{W}_{R,2}^{(1)} + \boldsymbol{b}_{R,1}^{(1)}\right) \tag{1}$$

$$\boldsymbol{U}_t^{(1)} = \phi_{sigmoid}\left(\widehat{\boldsymbol{x}}_t \boldsymbol{W}_{U,1}^{(1)} + \boldsymbol{h}_{t-1}^{(1)} \boldsymbol{W}_{U,2}^{(1)} + \boldsymbol{b}_{U,1}^{(1)}\right) \tag{2}$$

where $\phi_{sigmoid}(x) = \frac{1}{1+e^{-x}}$, $\boldsymbol{W}_{R,1}^{(1)}$, $\boldsymbol{W}_{R,2}^{(1)}$, $\boldsymbol{W}_{U,1}^{(1)}$ and $\boldsymbol{W}_{U,2}^{(1)}$ are the weight parameters, $\boldsymbol{b}_{R,1}^{(1)}$ and $\boldsymbol{b}_{U,1}^{(1)}$ are the bias. The candidate hidden state is denoted by $\widehat{\boldsymbol{h}}_t^{(1)} = \phi_{\tanh}\Bigg(\widehat{\boldsymbol{x}}_t \boldsymbol{W}_{H,1}^{(1)} + \left(\boldsymbol{R}_t^{(1)} \odot \boldsymbol{h}_{t-1}^{(1)}\right)\boldsymbol{W}_{H,2}^{(1)} + \boldsymbol{b}_{H,1}^{(1)}\Bigg)$, where $\boldsymbol{W}_{H,1}^{(1)}$ and $\boldsymbol{W}_{H,2}^{(1)}$ are the weight parameters, $\boldsymbol{b}_{H,1}^{(1)}$ is the bias, and $\odot$ is the Hadamard product operation. The output hidden state $\boldsymbol{h}_t^{(1)}$ is computed by

$$\boldsymbol{h}_t^{(1)} = f_g\left(\widehat{\boldsymbol{x}}_t, \boldsymbol{h}_{t-1}^{(1)}, 1\right) = \boldsymbol{U}_t^{(1)} \odot \boldsymbol{h}_{t-1}^{(1)} + \left(1 - \boldsymbol{U}_t^{(1)}\right) \odot \widehat{\boldsymbol{h}}_t^{(1)} \tag{3}$$

Accordingly, the output of the first layer of the GRU module $\boldsymbol{H}^{(1)} = \left[\boldsymbol{h}_1^{(1)}, \ldots, \boldsymbol{h}_T^{(1)}\right]^T$ can be obtained. To reduce the risk of model overfitting, a dropout operation is applied to $\boldsymbol{h}_t^{(1)}$ at the dropout layer to obtain $\widetilde{\boldsymbol{h}}_t^{(1)}$, i.e., $\widetilde{\boldsymbol{h}}_t^{(1)} = \boldsymbol{h}_t^{(1)} \odot \boldsymbol{\delta}^{(1)}$, where $\boldsymbol{\delta}^{(1)}$ is a random

vector and the elements within the vector take the value 0 with probability d_r and take the value $1/(1-d_r)$ with probability $1-d_r$.

Similarly, the output at the t-th moment of the n_g-th GRU layer is $\boldsymbol{h}_t^{(n_g)} = f_g\Big(\widetilde{\boldsymbol{h}}_t^{(n_g-1)}, \boldsymbol{h}_{t-1}^{(n_g)}, n_g\Big)$, where $\widetilde{\boldsymbol{h}}_t^{(n_g-1)} = \boldsymbol{h}_t^{(n_g-1)} \odot \boldsymbol{\delta}^{(n_g-1)}$. Hence, the output of the GRU module $\boldsymbol{H}^{(N_G)} = \left[\boldsymbol{h}_1^{(N_G)}, \ldots, \boldsymbol{h}_T^{(N_G)}\right]^T \in \mathbb{R}^{T\times d_g^{(N_G)}}$ can be obtained. Further, in order to extract the local and global features of the samples, the residual dilation convolution module is applied in the model. The output of the first convolution block $\boldsymbol{S}^{(1)} \in \mathbb{R}^{T\times d_s^{(1)}}$ can be calculated as follows:

$$\begin{aligned}\boldsymbol{S}^{(1)} &= f_s\left(\boldsymbol{H}^{(N_G)}, 1\right)\\ &= f_c\left(\boldsymbol{H}^{(N_G)}, 1\right) + \left(\phi_{GELU}\left(\phi_{GELU}\left(\boldsymbol{H}^{(N_G)}\right) *_{d^{(1)}} \boldsymbol{W}_{S,1}^{(1)}\right) *_{d^{(1)}} \boldsymbol{W}_{S,2}^{(1)}\right)\end{aligned} \tag{4}$$

where $d_s^{(1)}$ is the number of feature dimensions, $f_s(\cdot)$ is the residual dilation convolution block calculation formula, $*_{d^{(1)}}$ is the dilation convolution operation with the dilation factor $d^{(1)}$, $\boldsymbol{W}_{S,1}^{(1)}$ and $\boldsymbol{W}_{S,2}^{(1)}$ are the weight parameters of the one-dimensional dilation convolution operation, and $f_c(\cdot)$ is the projection function that ensures dimension matching:

$$f_c\left(\boldsymbol{H}^{(N_G)}, 1\right) = \begin{cases} \boldsymbol{H}^{(N_G)} * \boldsymbol{W}_{S,3}^{(1)}, & \text{if } d_s^{(1)} \neq d_g^{(N_G)} \\ \boldsymbol{H}^{(N_G)}, & \text{if } d_s^{(1)} = d_g^{(N_G)} \end{cases} \tag{5}$$

where $*$ is the one-dimensional convolution operation, and $\boldsymbol{W}_{S,3}^{(1)}$ is the corresponding weight.

The input of the n_s-th convolution block is the output of the previous convolution block $\boldsymbol{S}^{(n_s-1)}$, $1 < n_s \leq N_S$, and its output $\boldsymbol{S}^{(n_s)}$ can be expressed as

$$\boldsymbol{S}^{(n_s)} = f_c\left(\boldsymbol{S}^{(n_s-1)}, n_s\right) + \left(\phi_{GELU}\left(\phi_{GELU}\left(\boldsymbol{S}^{(n_s-1)}\right) *_{d^{(n_s)}} \boldsymbol{W}_{S,1}^{(n_s)}\right) *_{d^{(n_s)}} \boldsymbol{W}_{S,2}^{(n_s)}\right) \tag{6}$$

The final output of the residual dilation convolution module $\boldsymbol{S}^{(N_S)} = \left[\boldsymbol{s}_1^{(N_S)}, \ldots, \boldsymbol{s}_T^{(N_S)}\right]^T$ can be obtained by performing layer-by-layer operations according to Equation (6). By using different dilation factors for different layers of convolution blocks, the model can extract features at various scales.

At the feature fusion layer, $\boldsymbol{S}^{(N_S)}$ and $\boldsymbol{H}^{(N_G)}$ are spliced along the feature dimension to obtain the final output representation $\boldsymbol{Z} = \left[\boldsymbol{S}^{(N_S)}, \boldsymbol{H}^{(N_G)}\right] \in \mathbb{R}^{T\times M}$, where $M = d_s^{(N_S)} + d_g^{(N_G)}$ is the feature dimension. In this way, the augmented representation dataset $\boldsymbol{\mathcal{Z}}_P = \{\boldsymbol{Z}_{1,1}, \ldots, \boldsymbol{Z}_{1,P}, \ldots, \boldsymbol{Z}_{N,1}, \ldots, \boldsymbol{Z}_{N,P}\}$ of $\boldsymbol{\mathcal{X}}_P$ can be obtained by the TD-RLM model, where the representation data of $\boldsymbol{X}_{i,j}$ is $\boldsymbol{Z}_{i,j} = [\boldsymbol{z}_{i,j,1}, \ldots, \boldsymbol{z}_{i,j,T_j}]^T$.

2.2. **Feature representation based on joint loss function.** In order to enable the TD-RLM model to fully exploit the inter-class separability, intra-class diversity and intrinsic specificity features of the data, a joint loss function $\mathcal{L}$ [9] consisting of hierarchical mixed-supervised contrastive loss L_{HSCL}, cross-entropy loss L_{CE} and reconstruction loss L_{RE} is chosen to optimize the parameters of the model, which is represented as

$$\mathcal{L} = \lambda_{L,1} \cdot L_{HSCL} + \lambda_{L,2} \cdot L_{CE} + \lambda_{L,3} \cdot L_{RE} \tag{7}$$

where $\lambda_{L,1}$, $\lambda_{L,2}$ and $\lambda_{L,3}$ are the weights corresponding to these loss modules, respectively.

In Equation (7), L_{HSCL} helps to improve the model's ability to perceive similarity and difference features between samples. Before computing L_{HSCL}, a cropping operation is performed on $\boldsymbol{\mathcal{Z}}_P$ along the time axis to preserve its data on the overlapping time intervals and

to obtain the new representation data $\widehat{\boldsymbol{\mathcal{Z}}}_P = \left\{\widehat{\boldsymbol{Z}}_{1,1}, \ldots, \widehat{\boldsymbol{Z}}_{1,P}, \ldots, \widehat{\boldsymbol{Z}}_{N,1}, \ldots, \widehat{\boldsymbol{Z}}_{N,P}\right\}$, where $\widehat{\boldsymbol{Z}}_{i,j} = \left[\widehat{\boldsymbol{z}}_{i,j,1}, \ldots, \widehat{\boldsymbol{z}}_{i,j,\widehat{T}}\right]^T$, and $\widehat{T}$ is the number of time steps in the overlapping time intervals. Furthermore, in order to exploit the information at different temporal granularities, a maximum pooling operation of scale 2 is recursively performed on $\widehat{\boldsymbol{\mathcal{Z}}}_P$ along the time axis to obtain a representation dataset with $N_E = \left\lfloor \log_2 \widehat{T} \right\rfloor + 1$ granularities, where $\lfloor \cdot \rfloor$ is the downward rounding sign. The representation data at the granularity n_e is denoted as $\widehat{\boldsymbol{\mathcal{Z}}}_P^{(n_e)} = \left\{\widehat{\boldsymbol{Z}}_{1,1}^{(n_e)}, \ldots, \widehat{\boldsymbol{Z}}_{1,P}^{(n_e)}, \ldots, \widehat{\boldsymbol{Z}}_{N,1}^{(n_e)}, \ldots, \widehat{\boldsymbol{Z}}_{N,P}^{(n_e)}\right\}$, where $\widehat{\boldsymbol{Z}}_{i,j}^{(n_e)} = \left[\widehat{\boldsymbol{z}}_{i,j,1}^{(n_e)}, \ldots, \widehat{\boldsymbol{z}}_{i,j,\widehat{T}^{(n_e)}}^{(n_e)}\right]^T$, and $\widehat{T}^{(n_e)}$ is the length of time at the corresponding granularity. L_{HSCL} is defined as

$$L_{HSCL} = \frac{1}{N_E \cdot N \cdot P \cdot \widehat{T}^{(n_e)}} \sum_{n_e=1}^{N_E} \sum_{i=1}^{N} \sum_{j=1}^{P} \sum_{t=1}^{\widehat{T}^{(n_e)}} \left(L_{intra}^{(i,j,t,(n_e))} + L_{inter}^{(i,j,t,(n_e))}\right) \tag{8}$$

$$L_{intra}^{(i,j,t,(n_e))} = \frac{-1}{|A^*(i,j)|} \times \sum_{a \in A^*(i,j)} \ln \frac{\exp\left(\widehat{\boldsymbol{z}}_{i,j,t}^{(n_e)} \cdot \widehat{\boldsymbol{z}}_{i,a,t}^{(n_e)}\right)}{\sum_{m \in \widehat{B}(i)} \sum_{n \in \widehat{A}(m)} \exp\left(\widehat{\boldsymbol{z}}_{i,j,t}^{(n_e)} \cdot \widehat{\boldsymbol{z}}_{m,n,t}^{(n_e)}\right) + \sum_{a \in A^*(i,j)} \exp\left(\widehat{\boldsymbol{z}}_{i,j,t}^{(n_e)} \cdot \widehat{\boldsymbol{z}}_{i,a,t}^{(n_e)}\right)} \tag{9}$$

$$L_{inter}^{(i,j,t,(n_e))} = \frac{-1}{|H^*(i,j)|} \times \left(\sum_{m \in \widehat{B}(i)} \sum_{n \in \widehat{A}(m)} \ln \frac{\exp\left(\widehat{\boldsymbol{z}}_{i,j,t}^{(n_e)} \cdot \widehat{\boldsymbol{b}}_{m,n,t}^{(n_e)}\right)}{f_{n_e}(i,j,t,(n_e))} + \sum_{a \in A^*(i,j)} \ln \frac{\exp\left(\widehat{\boldsymbol{z}}_{i,j,t}^{(n_e)} \cdot \widehat{\boldsymbol{z}}_{i,a,t}^{(n_e)}\right)}{f_{n_e}(i,j,t,(n_e))} \right) \tag{10}$$

$$f_{n_e}(i,j,t,(n_e)) = \sum_{p \in \widehat{C}(i)} \sum_{q \in \widehat{A}(p)} \exp\left(\widehat{\boldsymbol{z}}_{i,j,t}^{(n_e)} \cdot \widehat{\boldsymbol{z}}_{p,q,t}^{(n_e)}\right) + \sum_{a \in A^*(i,j)} \exp\left(\widehat{\boldsymbol{z}}_{i,j,t}^{(n_e)} \cdot \widehat{\boldsymbol{z}}_{i,a,t}^{(n_e)}\right) \tag{11}$$

where N_E is the number of granularities, N is the number of samples, P is the number of augmented samples, and $L_{intra}^{(i,j,t,(n_e))}$ and $L_{inter}^{(i,j,t,(n_e))}$ are the intra-class and inter-class contrastive losses, respectively, $A^*(i,j)$ is the representation set of augmented samples of $\boldsymbol{X}_i$ expect for $\widehat{\boldsymbol{Z}}_{i,j}^{(n_e)}$, and $|A^*(i,j)|$ is its cardinality. $\widehat{\boldsymbol{z}}_{i,j,t}^{(n_e)}$ is the representation vector of the j-th augmented sample of $\boldsymbol{X}_i$ at time step t under time granularity n_e. $\widehat{B}(i)$ is the set of indicators for all input samples belonging to the same class as $\boldsymbol{X}_i$ except for $\boldsymbol{X}_i$. $\widehat{A}(m)$ denotes the set of all augmented samples of sample $\boldsymbol{X}_m$. $H^*(i,j)$ is the set of all augmented samples belonging to the same class as $\boldsymbol{X}_i$ except for $\boldsymbol{X}_{i,j}$, and $|H^*(i,j)|$ is its cardinality. $\widehat{C}(i)$ is all input samples expect for $\boldsymbol{X}_i$.

L_{CE} guides model optimization by comparing the difference between the model's predicted outputs and the true labels. The maximum pooling and dropout operations are applied to obtaining the instance-level representation data $\boldsymbol{Z}_{i,j}^* = \left[\boldsymbol{z}_{i,j,1}^*, \ldots, \boldsymbol{z}_{i,j,M}^*\right] \in \mathbb{R}^{1 \times M}$, where $\boldsymbol{z}_{i,j,m}^* = \max_{1 \le t \le T_j} (z_{i,j,t,m}) \cdot \delta_m$, $m = 1, \ldots, M$, δ_m takes the value 0 with probability d_r and takes the value $1/(1-d_r)$ with probability $1-d_r$. The predicted output value $\widehat{\boldsymbol{Y}}_{i,j} = [\widehat{y}_{i,j,1}, \ldots, \widehat{y}_{i,j,C}]$ is obtained by using $\boldsymbol{Z}_{i,j}^*$ for category probability prediction:

$$\widehat{\boldsymbol{Y}}_{i,j} = \phi_{softmax}\left(\boldsymbol{Z}_{i,j}^* \boldsymbol{W}_{O,1} + \boldsymbol{b}_{O,1}\right) \tag{12}$$

where $\boldsymbol{W}_{O,1}$ and $\boldsymbol{b}_{O,1}$ are the corresponding weight parameters and bias, respectively, and $\phi_{softmax}(\cdot)$ is the softmax activation function. L_{CE} is defined as

$$L_{CE} = -\frac{1}{N \cdot A} \sum_{i=1}^{N} \sum_{j=1}^{A} \sum_{c=1}^{C} \widetilde{y}_{i,c} \ln\left(\widehat{y}_{i,j,c}\right) \tag{13}$$

$$\widetilde{y}_{i,c} = \begin{cases} 1-\varepsilon, & \text{if } y_{i,c} \neq 0 \\ \varepsilon/(C-1), & \text{if } y_{i,c} = 0 \end{cases} \tag{14}$$

where ε is the smoothing factor.

L_{RE} allows the model to better capture the intrinsic structure and specificity features of the sample. The representation data at each time step $\widehat{\boldsymbol{z}}_{i,j,t}$ is first mapped to the original data space to obtain the reconstructed data $\boldsymbol{x}^*_{i,j,t} = \left[x^*_{i,j,t,1}, x^*_{i,j,t,2}, \ldots, x^*_{i,j,t,L}\right]$, i.e.,

$$\boldsymbol{x}^*_{i,j,t} = \widehat{\boldsymbol{z}}_{i,j,t}\boldsymbol{W}_{O,2} + \boldsymbol{b}_{O,2} \tag{15}$$

where $\boldsymbol{W}_{O,2}$ and $\boldsymbol{b}_{O,2}$ are the corresponding weight parameters and bias, respectively. Then, L_{RE} can be calculated by the following equation:

$$L_{RE} = \frac{1}{N \cdot A \cdot \widehat{T} \cdot L} \sum_{i=1}^{N} \sum_{j=1}^{A} \sum_{t=1}^{\widehat{T}} \sum_{l=1}^{L} (x^*_{i,j,t,l} - x_{i,t,l})^2 \tag{16}$$

Based on the idea in [9], the weights of the joint loss function are dynamically updated. For the k-th iteration, the computation is expressed as

$$\lambda_{L,i}(k) = \frac{3 \cdot \exp(\Theta_{L_i}(k)/\tau)}{\sum_{j=1}^{3} \exp(\Theta_{L_j}(k)/\tau)}, \ i = 1, 2, 3 \tag{17}$$

where L_1, L_2 and L_3 are indicative of L_{HSCL}, L_{CE} and L_{RE}, respectively, τ is the temperature factor, and $\Theta_{L_i}(k)$ is the corresponding amount of loss fluctuation:

$$\Theta_{L_i}(k) = \frac{|L_i(k-1) - L_i(k-2)|}{\sum_{j=1}^{3} |L_i(k-1) - L_i(k-2)|}, \ i = 1, 2, 3 \tag{18}$$

where $L_i(k-1)$ is the loss value at the $(k-1)$-th iteration.

In the process of model training, the parameters to be optimized include weight parameters and bias of the mapping layer, the GRU module, the residual dilation convolution module and the loss function. Parameters $\boldsymbol{\mathcal{W}}^*$ are determined by solving the equation:

$$\boldsymbol{\mathcal{W}}^* = \arg\min_{\boldsymbol{\mathcal{W}}} \mathcal{L}(\boldsymbol{\mathcal{W}}) \tag{19}$$

where $\mathcal{L}(\cdot)$ is the joint loss function defined in Equation (7). The adaptive moment estimation with weight decay (AdamW) algorithm [10] is used to solve the optimization problem (19). Further, in order to alleviate the problem of parameters oscillating near the optimal solution during the model training process, the stochastic weighted averaging (SWA) method [11] is further employed to synthesize the local optimal solution in each iteration. $\boldsymbol{\mathcal{W}}^*$ can be calculated by $\boldsymbol{\mathcal{W}}^* = \frac{1}{N_D} \sum_{i=1}^{N_D} \boldsymbol{\mathcal{W}}_i$, where N_D is the number of iterations, and $\boldsymbol{\mathcal{W}}_i$ is the model parameter at the i-th iteration.

3. Classification Modeling Based on Ensemble Hierarchical Networks.

In this section, a hierarchical broad network with multi-level features is designed for data classification.

3.1. Construction of hierarchical broad network with multi-level features.

Firstly, extreme gradient boosting (XGBoost) [12] is chosen to calculate the importance of each feature, the corresponding feature importance vectors are denoted by $\boldsymbol{V} = [v_1, \ldots, v_M]$. The sorted data is expressed as $\widetilde{\boldsymbol{Z}} = [\widetilde{\boldsymbol{z}}_1, \ldots, \widetilde{\boldsymbol{z}}_N]^T$. By the implementation of feature importance metrics, the hierarchical broad network (HB-Net) based on multi-level feature fusion is designed to improve the processing capability of high-dimensional data. The basic structure is shown in Figure 2.

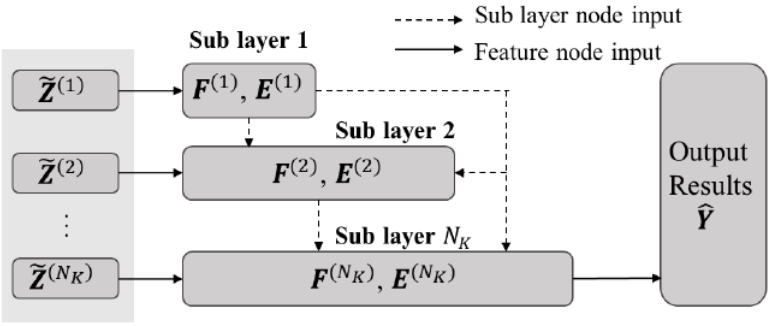

FIGURE 2. The architecture of HB-Net

In the construction of HB-Net with N_K layers, the input data $\widetilde{\boldsymbol{Z}}$ is first represented as a collection of N_K subsets:

$$\widetilde{\boldsymbol{Z}} = \left\{ \widetilde{\boldsymbol{Z}}^{(1)}, \ldots, \widetilde{\boldsymbol{Z}}^{(N_K)} \right\} = \begin{bmatrix} \tilde{z}_{1,1}^{(1)} & \cdots & \tilde{z}_{1,d_h}^{(1)} & \cdots & \tilde{z}_{1,1}^{(N_K)} & \cdots & \tilde{z}_{1,d_h}^{(N_K)} \\ \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \\ \tilde{z}_{N,1}^{(1)} & \cdots & \tilde{z}_{N,d_h}^{(1)} & \cdots & \tilde{z}_{N,1}^{(N_K)} & \cdots & \tilde{z}_{N,d_h}^{(N_K)} \end{bmatrix} \tag{20}$$

where $\widetilde{\boldsymbol{Z}}^{(N_K)}$ is the N_K-th subset of the input data, and $d_h = M/N_K$ is the number of features contained in each subset.

The input data for the first sub-layer is $\boldsymbol{X}_H^{(1)} = \widetilde{\boldsymbol{Z}}^{(1)}$. In the first sub-layer, feature mapping operations are first performed on $\boldsymbol{X}_H^{(1)}$ to obtain $N_Q^{(1)}$ groups of mapped features $\left\{ \boldsymbol{F}_i^{(1)} \middle| \left(i = 1, \ldots, N_Q^{(1)} \right) \right\}$. Each group of features contains $n_{q,i}^{(1)}$ mapped nodes, and the computation is represented by $\boldsymbol{F}_i^{(1)} = \phi\left(\boldsymbol{X}_H^{(1)} \boldsymbol{W}_{F,i}^{(1)} + \boldsymbol{\beta}_{F,i}^{(1)} \right)$, where $\phi(\cdot)$ is a nonlinear activation function, $\boldsymbol{W}_{F,i}^{(1)}$ and $\boldsymbol{\beta}_{F,i}^{(1)}$ are weight parameters and biases that are randomly generated. The total mapped features $\boldsymbol{F}^{(1)} = \left[\boldsymbol{F}_1^{(1)}, \ldots, \boldsymbol{F}_{N_Q^{(1)}}^{(1)} \right] \in \mathbb{R}^{N \times Q^{(1)}}$ can be obtained splicing all the mapped features yields, where $Q^{(1)} = \sum_{i=1}^{N_Q^{(1)}} n_{q,i}^{(1)}$. Supplemental learning of $\boldsymbol{F}^{(1)}$ by the enhancement mapping operation leads to $N_R^{(1)}$ groups of enhancement features $\left\{ \boldsymbol{E}_i^{(1)} \middle| \left(i = 1, \ldots, N_R^{(1)} \right) \right\}$, each containing $n_{r,i}^{(1)}$ enhancement nodes which is computed as $\boldsymbol{E}_i^{(1)} = \xi\left(\boldsymbol{F}^{(1)} \boldsymbol{W}_{E,i}^{(1)} + \boldsymbol{\beta}_{E,i}^{(1)} \right)$, where $\boldsymbol{W}_{E,i}^{(1)}$ and $\boldsymbol{\beta}_{E,i}^{(1)}$ are weight parameters and biases that are randomly generated. The total enhancement features $\boldsymbol{E}^{(1)} = \left[\boldsymbol{E}_1^{(1)}, \ldots, \boldsymbol{E}_{N_R^{(1)}}^{(1)} \right] \in \mathbb{R}^{N \times R^{(1)}}$ by splicing all the enhancement features, where $R^{(1)} = \sum_{i=1}^{N_R^{(1)}} n_{r,i}^{(1)}$.

The mapped feature $\boldsymbol{F}^{(1)}$ with the enhancement feature $\boldsymbol{E}^{(1)}$ is spliced to obtain the output feature node of the first sub-layer $\boldsymbol{A}^{(1)} = \left[\boldsymbol{F}^{(1)}, \boldsymbol{E}^{(1)} \right]$.

The input data of the n_k-th sub-layer consists of the output feature nodes of all previous sub-layers with a new subset of input features $\widetilde{\boldsymbol{Z}}^{(n_k)}$, $1 < n_k \leq N_K$. They are spliced to obtain the input data of the current layer $\boldsymbol{X}_H^{(n_k)} = \left[\boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(n_k-1)}, \widetilde{\boldsymbol{Z}}^{(n_k)} \right] \in \mathbb{R}^{N \times D^{(n_k)}}$, where $D^{(n_k)} = \sum_{i=1}^{n_k-1} \left(Q^{(i)} + R^{(i)} \right) + d_h$ is the feature dimension. The internal operation of this sub-layer is the same as the first sub-layer, the feature mapping operation is performed first to obtain $N_Q^{(n_k)}$ groups of mapped features $\left\{ \boldsymbol{F}_i^{(n_k)} \middle| \left(i = 1, \ldots, N_Q^{(n_k)} \right) \right\}$,

each containing $n_{q,i}^{(n_k)}$ mapped nodes: $\boldsymbol{F}_i^{(n_k)} = \phi\left(\boldsymbol{X}_H^{(n_k)}\boldsymbol{W}_{F,i}^{(n_k)} + \boldsymbol{\beta}_{F,i}^{(n_k)}\right)$, where $\boldsymbol{W}_{F,i}^{(n_k)}$ and $\boldsymbol{\beta}_{F,i}^{(n_k)}$ are weight parameters and biases that are randomly generated. All the mapped features are spliced to get the total mapped features of the current sub-layer $\boldsymbol{F}^{(n_k)} = \left[\boldsymbol{F}_1^{(n_k)}, \ldots, \boldsymbol{F}_{N_Q^{(n_k)}}^{(n_k)}\right]$.

Similar to the above process, $N_R^{(n_k)}$ groups of enhancement features $\boldsymbol{E}^{(n_k)}$ can be obtained. The output feature node of the current sub-layer is $\boldsymbol{A}^{(n_k)} = \left[\boldsymbol{F}^{(n_k)}, \boldsymbol{E}^{(n_k)}\right]$. The output feature of the last sub-layer is $\boldsymbol{A}^{(N_K)} = \left[\boldsymbol{F}^{(N_K)}, \boldsymbol{E}^{(N_K)}\right]$.

Further, based on the broad learning system [13], the predicted output of the model can be calculated as $\widehat{\boldsymbol{Y}} = \boldsymbol{A}^{(N_K)}\boldsymbol{W}_{O,3}$, where $\boldsymbol{W}_{O,3}$ can be determined by solving the following optimal problem:

$$\arg\min_{\boldsymbol{W}_{O,3}} J(\boldsymbol{W}_{O,3}) = \left\|\boldsymbol{A}^{(N_K)}\boldsymbol{W}_{O,3} - \boldsymbol{Y}\right\|_2^2 + \frac{\alpha}{2}\|\boldsymbol{W}_{O,3}\|_2^2 \tag{21}$$

The corresponding solution can be expressed as $\boldsymbol{W}_{O,3} = \left(\alpha\boldsymbol{I} + \boldsymbol{A}^{(N_K)}\boldsymbol{A}^{(N_K)^T}\right)^{-1}\boldsymbol{A}^{(N_K)^T}\boldsymbol{Y}$, where $\boldsymbol{I}$ is the unit matrix and α is the regularization factor.

3.2. **Ensemble hierarchical broad network construction for multi-scale information fusion.** In order to effectively utilize the representation information at different scales, an ensemble hierarchical broad network (EHB-Net) that fuses multi-scale information is further proposed.

First, the representation data at different scales are obtained by dividing $\boldsymbol{\mathcal{Z}}$ along the time dimension. Set the start moment of the time window for the first division as 1 and the end moment as T. Then, the start moment of the time window for each division is shifted back by the length Δ_d. The n_f-th division has a time window with a start moment of $1 + (n_f - 1)\Delta_d$, the end moment of T, and a time scale of $T - (n_f - 1)\Delta_d$. The division is stopped until the start moment exceeds the threshold S_d.

By this step, a multi-scale dataset consisting of N_F different scales representation data $\mathfrak{Z} = \left\{\mathcal{Z}^{(n_f)} \middle| (n_f = 1, \ldots, N_F)\right\}$ can be obtained, where $N_F = \left\lfloor \frac{T - S_d}{\Delta_d} \right\rfloor + 1$. By iterating each element in $\mathfrak{Z}$ and performing a global max-pooling operation on it along the time dimension, the dataset consisting of N_F instance-level representation data with different scales $\boldsymbol{\mathcal{Z}}^* = \left\{\boldsymbol{Z}^{*(n_f)} \middle| (n_f = 1, \ldots, N_F)\right\}$ can be obtained, where $\boldsymbol{Z}^{*(n_f)} = \left[\boldsymbol{z}_1^{*(n_f)}, \ldots, \boldsymbol{z}_N^{*(n_f)}\right]^T$. A similarity-based filtering strategy is developed to select multi-scale datasets for ensemble modeling. The similarity c_{n_f} between $\boldsymbol{Z}^{*(n_f)}$ and $\boldsymbol{Z}^{*(1)}$ is calculated by the following equation:

$$c_{n_f} = \frac{\sum_{i=1}^{N}\sum_{m=1}^{M} z_{i,m}^{*(n_f)} z_{i,m}^{*(1)}}{\sqrt{\sum_{i=1}^{N}\sum_{m=1}^{M}\left(z_{i,m}^{*(n_f)}\right)^2} \times \sqrt{\sum_{i=1}^{N}\sum_{m=1}^{M}\left(z_{i,m}^{*(1)}\right)^2}} \tag{22}$$

where $z_{i,m}^{*(n_f)}$ is the value of the m-th feature of the i-th sample in $\boldsymbol{Z}^{*(n_f)}$.

By calculating the similarity value of each element in $\boldsymbol{\mathcal{Z}}^*$ with $\boldsymbol{Z}^{*(1)}$, the similarity vector $\boldsymbol{C}^* = [c_1, \ldots, c_{N_F}]$ is obtained. Define the dataset $\mathcal{D}^* = \left\{\boldsymbol{Z}^{*(1)}\right\}$ for ensemble modeling, and denote the number of elements in this set as $|\mathcal{D}^*|$. Set the similarity threshold and difference thresholds as S_c and S_v, respectively. For any $i \in \{1, \ldots, |\mathcal{D}^*|\}$ and $j \in \{2, \ldots, N_F\}$, if $c_j > S_c$ and $|c_j - c_i| > S_v$, then $\boldsymbol{Z}^{*(j)}$ is added to $\mathcal{D}^*$; otherwise, $\mathcal{D}^*$ remains unchanged. Suppose that there are N_U elements in $\mathcal{D}^*$ after screening, the output $\widehat{\boldsymbol{Y}}$ of the EHB-Net model is computed by a weighted sum of N_U HB-Net base models,

i.e., $\widehat{\boldsymbol{Y}} = \sum_{n_u=1}^{N_U} w_{n_u} \widehat{\boldsymbol{Y}}_{n_u}$, where $\widehat{\boldsymbol{Y}}_{n_u}$ is the output of the n_u-th base model and w_{n_u} is the weight.

The main processes of EHB-Net are summarized as follows.

Step 1. Setting initial parameters of the EHB-Net model: the length Δ_d, the threshold S_d, the similarity threshold and difference thresholds as S_c and S_v.

Step 2. Based on Δ_d and S_d, determining the representation dataset $\mathfrak{Z}$ and using global max-pooling operation to obtain $\boldsymbol{\mathcal{Z}}^*$.

Step 3. Using similarity threshold S_c and difference thresholds S_v to obtain the multi-scale modeling dataset $\boldsymbol{\mathcal{D}}^*$.

Step 4. For each scale, using XGBoost to sort the features according to their importance.

Step 5. Computing the output $\widehat{Y}_{n_u}$ with different scale n_u.

Step 6. Computing the actual output $\widehat{\boldsymbol{Y}}$ by integrating the output $\widehat{Y}_{n_u}$ with different scale.

4. Simulation Experiment and Analysis.

4.1. **Description of dataset and data processing.** In this section, simulations are conducted using the following public dataset to evaluate the effectiveness of the proposed method. The Physionet 2012 dataset contains the anonymized records of 11,988 Intensive Care Unit (ICU) patients and aims to predict the risk of death during hospitalization based on observations recorded during the first 48 hours of a patient's ICU stay. 7671 samples, 1917 samples, and 2400 samples are respectively chosen as the training set, validation set, and testing set.

To ensure the reliability of the data, the data are filled using a combination of forward filling and normalized filling. All data are normalized by subtracting the mean of the training set and dividing by its standard deviation.

The area AUROC under the receiver operating characteristic (ROC) curve and the area AUPRC under the precision-recall (PRC) curve are used as evaluation metrics in the experiments. The high values of AUROC and AUPRC mean that the classification model achieves better performance.

4.2. **Numerical simulation.** For the Physionet 2012 dataset, the learning rate is set to 0.0005, the dropout rate is set to 0.2, and the temperature factor is set to 1.

The parameter settings of TD-RLM are as follows. The number of data enhancements $P = 3$. The threshold for subsequence cropping $S_t = 2^5$. The number of feature dimensions of the mapping layer $D = 64$. The number of layers of the residual dilation convolution block $N_S = 3$. The feature dimension of the residual dilation convolution block is set to 64. The dimension of the dilation factor is set to 2^{N_S-1}. The dimension of the convolution kernel is set to 3. The number of layers in the GRU module $N_G = 3$. The dimension of the hidden layer state in the GRU module is set to 128. The smoothing factor in cross-entropy loss $\varepsilon = 0.05$. The training epochs $N_A = 100$. The batch size $N_B = 8$. The weight decay factor is 3e-4.

The parameter settings of EHB-Net are as follows. The length of scale-divided nudges $\Delta_d = 2$. The starting moment threshold for scale division $S_d = 24$. The similarity threshold $S_c = 0.95$. The difference threshold $S_v = 0.01$. The number of enhancement feature groups $N_R = 1$. The regularization factor $\alpha = 0.08$.

In order to evaluate the effectiveness of the proposed method, comparisons are made with some baseline models on the dataset, including Retain [5], AdaCare [14], ConCare [7] and LGTRL-DE [8]. All baseline models are compared under the same preprocessing and variable selection conditions.

Table 1 reports the means and standard deviations of all models under three random seeds. Due to the fact that AUPRC is more sensitive to the imbalance of data, there are some differences between AUROC and AUPRC. Compared to the other methods, the proposed method still achieves better performances on these two indexes.

Table 1. Classification results on dataset

Model	Retain	AdaCare	ConCare	LGTRL-DE	**Proposed model**
AUROC	82.4 ± 0.2	86.1 ± 0.2	86.1 ± 0.3	86.2 ± 0.3	**86.7 ± 0.1**
AUPRC	45.3 ± 0.8	53.9 ± 0.4	50.6 ± 0.6	54.8 ± 0.1	**54.9 ± 0.2**

In order to verify the effectiveness of each feature extraction module and loss module in the TD-RLM model, as well as the effectiveness of the HB-Net base model and the multi-scale ensemble strategy in the EHB-Net, the corresponding ablation experiments are performed. Table 2 reports the means and standard deviations of all models based on three random seeds.

Table 2. Results of ablation experiments

Model	AUROC
Remove convolution module	85.9 ± 0.3
Remove GRU module	86.5 ± 0.1
Remove L_{HSCL}	85.7 ± 0.3
Remove L_{CE}	84.6 ± 1.3
Remove L_{RE}	86.3 ± 0.2
Remove hierarchical structure	86.6 ± 0.1
Feature reverse sorting	86.3 ± 0.5
Remove ensemble strategy	86.5 ± 0.1
Proposed model	**86.7 ± 0.1**

The experimental results show that the proposed method achieves optimal performance on all performance evaluation metrics. These results also illustrate that the individual modules in the representation and classification models play a role in improving the performance. Meanwhile, it can be seen that the performance of the variant model is reduced with the removal of L_{CE}, L_{HSCL} and L_{RE}. These facts mean that the joint function is beneficial for improving model accuracy.

Simulation experiment under different feature importance analysis methods is conducted. The results are shown in Table 3, which demonstrate that the EHB-Net model with XGBoost can capture better feature information.

Table 3. Experimental results of different feature importance identification methods

Model	Random Forest	Logistic Regression	Lasso	Pearson	**XGBoost**
AUROC	86.5 ± 0.1	86.6 ± 0.1	**86.7 ± 0.1**	86.5 ± 0.1	**86.7 ± 0.1**

The experimental results with two feature fusion strategies are shown in Table 4. It can be seen that simple concatenation still obtains satisfactory performance.

Simulation results under different numbers of random cropping operations P are shown in Table 5. When $P = 3$, the model can achieve better classification accuracy.

TABLE 4. Experimental results of different fusion methods

Fusion methods	Weighted fusion	Simple concatenation
AUROC	86.1	**86.2**

TABLE 5. Experimental results with different numbers of random cropping operations

Different P	2	3	4
AUROC	85.4	**86.2**	86.0

5. **Conclusion.** In this paper, a feature representation model with a joint loss function is considered to obtain the multi-scale information. The scale selection principle is formulated to represent multi-scale information for ensemble modeling. A hierarchical-broad network with multi-level feature fusion is designed for handling time series data classification. Although the proposed method achieves good classification performance on the public dataset, the calculation amount of this model is relatively large. Therefore, in future work, we will focus on the lightweight learning method to reduce the computational complexity and optimize the number of parameters.

Acknowledgment. This work was partially supported by the Humanities and Social Science project for Ministry of Education of China (21YJA630079). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] J. Grabocka, N. Schilling, M. Wistuba et al., Learning time-series shapelets, *Proc. of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, USA, pp.392-401, 2014.

[2] L. Ye and E. Keogh, Time series shapelets: A new primitive for data mining, *Proc. of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Paris, France, pp.947-956, 2009.

[3] E. Eldele, M. Ragab, Z. Chen et al., Time-series representation learning via temporal and contextual contrasting, *Proc. of the 30th International Joint Conference on Artificial Intelligence*, Canada, pp.2352-2359, 2021.

[4] Z. Yue, Y. Wang, J. Duan et al., TS2Vec: Towards universal representation of time series, *Proc. of the AAAI Conference on Artificial Intelligence*, pp.8980-8987, 2022.

[5] E. Choi, M. T. Bahadori, J. Sun et al., RETAIN: An interpretable predictive model for healthcare using reverse time attention mechanism, *Advances in Neural Information Processing Systems*, vol.29, pp.3512-3520, 2016.

[6] H. Song, D. Rajan, J. Thiagarajan et al., Attend and diagnose: Clinical time series analysis using attention models, *Proc. of the AAAI Conference on Artificial Intelligence*, New Orleans, USA, pp.4901-4908, 2018.

[7] L. Ma, C. Zhang, Y. Wang et al., ConCare: Personalized clinical feature embedding via capturing the healthcare context, *Proc. of the AAAI Conference on ArtZificial Intelligence*, New York, USA, pp.833-840, 2020.

[8] M. Zou, Y. An, H. Kuang et al., LGTRL-DE: Local and global temporal representation learning with demographic embedding for in-hospital mortality prediction, *Journal of Biomedical Informatics*, vol.143, 104408, 2023.

[9] J. Tang and J. D. Wang, Multivariate time series classification based on multi-channel representation model and broad learning system, *2023 International Conference on New Trends in Computational Intelligence*, Qingdao, China, pp.402-406, 2023.

[10] I. Loshchilov and F. Hutter, Decoupled weight decay regularization, *The 7th International Conference on Learning Representations*, New Orleans, USA, pp.1-18, 2019.

[11] P. Izmailov, D. Podoprikhin, T. Garipov et al., Averaging weights leads to wider optima and better generalization, *The 34th Conference on Uncertainty in Artificial Intelligence*, Monterey, USA, pp.876-885, 2018.
[12] T. Chen and C. Guestrin, XGBoost: A scalable tree boosting system, *Proc. of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, USA, pp.785-794, 2016.
[13] C. Chen and Z. Liu, Broad learning system: An effective and efficient incremental learning system without the need for deep architecture, *IEEE Transactions on Neural Networks and Learning Systems*, vol.29, no.1. pp.10-24, 2018.
[14] L. Ma, J. Gao, Y. Wang et al., AdaCare: Explainable clinical health status representation learning via scale-adaptive feature extraction and recalibration, *Proc. of the AAAI Conference on Artificial Intelligence*, New York, USA, pp.825-832, 2020.

Author Biography

Degang Wang received the B.Sc. degree in Mathematics from Liaoning Normal University, China, in 2002; the M.Sc. degree in Mathematics from Liaoning Normal University, China, in 2005; the Ph.D. degree in Applied Mathematics from Beijing Normal University, China, in 2008. Dr. Wang is currently a full-time associate professor at the School of Control Science and Engineering, Dalian University of Technology, China. His research interests include fuzzy system modelling and applications, machine learning and intelligent control.

Jiani Tang received her Bachelor's degree in Automation from Nanjing Normal University, China, in 2021, and subsequently obtained her Master's degree in Electronic Information from Dalian University of Technology, China, in 2024. Her research interests include time series classification and representation learning.

Wenyan Song received the B.Sc. degree in Mathematics from Beijing Normal University, China, 2002; the Ph.D. degree in Applied Mathematics from Beijing Normal University, China, 2007. Dr. Song is currently a full-time associate professor at the School of Economics, Dongbei University of Finance and Economics, China. Her research interests include fuzzy system modelling, neural network and management optimization.

Yuchen Jin received his Bachelor's degree in Mechanical Manufacturing and Automation from Taiyuan University of Technology, China, in 2022, and subsequently obtained his Master's degree in Electronic Information from Dalian University of Technology, China, in 2025. His research interests include graph neural networks and remote sensing time series classification.