SCIENCE CHINA
Information Sciences

• **Supplementary File** •

Prediction of Budd-Chiari syndrome based on attention mechanisms of high-risk factors in multi-hop graph learning

Lei Wang[1,2†], Sheng-Li Li[1,3†], Xiao-Rui Su[4], Zheng-Wei Li[1], Meng-Meng Wei[1], Bo-Wei Zhao[5], Mao-Heng Zu[6], Qing-Qiao Zhang[6] & Zhu-Hong You[2]

[1]*School of Computer Science and Technology, China University of Mining and Technology, Xuzhou* 221116*, China;*
[2]*Guangxi Key Lab of Human-Machine Interaction and Intelligent Decision, Guangxi Academy of Sciences, Nanning* 530007*, China;*
[3]*Clinical Research Institute, The Affiliated Hospital of Xuzhou Medical University, Xuzhou* 221006*, China;*
[4]*University of Chinese Academy of Sciences, Beijing* 100049*, China;*
[5]*Xinjiang Technical Institute of Physics and Chemistry, Chinese Academy of Sciences, Urumqi* 830011*, China;*
[6]*Department of Interventional Radiology, The Affiliated Hospital of Xuzhou Medical University, Xuzhou* 221006*, China*

Appendix A Data collection and organization

The data used in this work were obtained from 754 BCS patients who received interventional treatment at Affiliated Hospital of Xuzhou Medical University from January 2015 to June 2022, and were approved by the Ethics Committee of Xuzhou Medical University Affiliated Hospital (XYFY2023-KL188-01). The diagnostic criteria for BCS were based on the 2021 Asia-Pacific Association for the Study of Liver (APASL) consensus guidelines and confirmed by angiography, Computed Tomography and color Doppler ultrasound. The following situations were excluded during data collection: 1. Obstruction caused by pericardial disease, sinusoidal obstruction syndrome or cardiac disease; 2. Secondary BCS caused by tumors, parasites, abscesses, postoperative venous injury, or cysts compressing the lumen; 3. Pregnancy or lactation; 4. Severe organ failure precluding interventional treatment; 5. History of iodine contrast agent allergy; 6. Incomplete case data. All patients were followed up every 6 months for 3 years. Patients with lumen diameter stenosis of more than 50% compared to postoperative luminal diameter or complete occlusion, or with new thrombosis of hepatic vein (HV), inferior vena cava (IVC), or collateral vessels, were considered to have recurred, resulting in 243 positive samples and 511 negative samples.

Patient data included 32 BCS-related factors such as age, sex, gender, Albumin (ALB), neutrophil count (NEU), etc. For quantitative data, we conducted normality tests. If the data followed a normal distribution, they were described using the mean and standard deviation, and then the independent samples t-test was used to determine the difference between the recurrent and non-recurrent sample groups; if the data did not follow a normal distribution, they were described using the median and interquartile spacing, and then the Mann-Whitney U test was used to determine the difference between the different sample groups. All tests had a P-value of less than 0.05, which was considered statistically significant, thus obtaining standardized data.

We construct a network-based data representation using the normalized data described above, and follow an experimentally validated and effective method [1–3] for calculating connection weights using the Pearson correlation coefficient.

$$PCC_{u_1,u_2} = \frac{n\sum u_1u_2 - \sum u_1 \sum u_2}{\sqrt{n\sum u_1^2 - (\sum u_1)^2}\ \sqrt{n\sum u_2^2 - (\sum u_2)^2}}, \tag{A1}$$

where u_1 and u_2 denote standardized BCS sample factors and n is the number of factors. Since not all samples are correlated with each other, we experimentally set greater than a threshold of 0.5 as having a positive correlation, and less than a threshold of 0.5 as having a negative correlation, thus constructing the BCS sample correlation network.

Appendix B ARM-BCS algorithm

Appendix B.1 Attention mechanism feature extraction

After standardization and network representation of patient sample data, we use a multi-hop graph neural network [4–6] with attention mechanism for deep feature extraction in anticipation of mining more representative features [7–10]. The specific procedures of the scheme are as follows:

Construct graph structure. For a given graph, it is mathematically described by $G = (U, E)$, where U denotes the set of nodes S_n and $E \subseteq U \times U$ denotes the set of edges S_e. Every node u in the node set U conforms to its type mapping function $\Theta : U \to \mathcal{A}$ and every edge e in the edge set E conforms to its type mapping function $\Phi : E \to \mathcal{R}$. The node embedding is labeled $V \in (\mathbb{R})^{S_n \times d_n}$, where d_r denotes the dimension of the edge type and $S_r = |\mathcal{R}|$. The embedding of node u_i $(1 \leqslant i \leqslant S_n)$ can be represented by the row vector v_i of node embedding V and the embedding of relation r_j $(1 \leqslant j \leqslant S_r)$ can be represented by the row vector r_i of edge embedding R.

* Corresponding author (email: leiwang@cumt.edu.cn, sjc19960612@gmail.com, zhuhongyou@nwpu.edu.cn)
† Wang L and Li S L have the same contribution to this work.

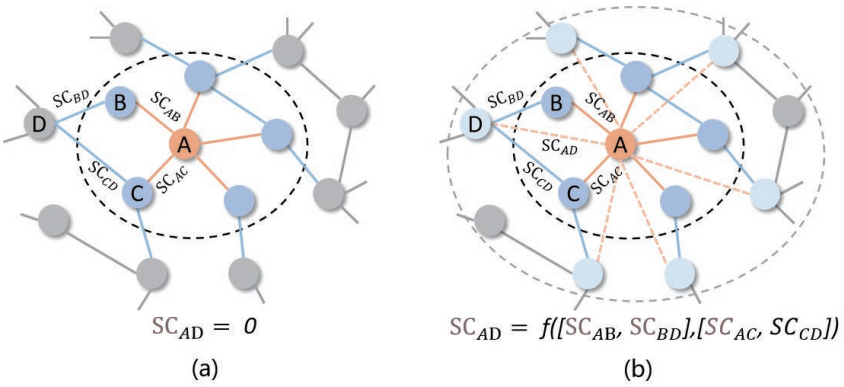

Figure B1 The two-hop attention schematic. Taking node A as an example, its neighboring nodes are shown within the dotted line in Figure (a), and the attention score SC_{AD} between it and node D is 0. The two-hop neighbor nodes of node A are within the two dashed lines shown in Figure (b), and the attention score SC_{AD} between it and node D is computed by combining the scores of the relevant neighboring nodes.

Calculation of edge attention. The attention scores of all edges in the graph structure is computed first. In each layer l of the model, the attention diffusion factor $(\ u_i,\ r_k,\ u_j\ )$ a set of triples, where u_i and u_j represent nodes and r_k represents edge types. Merge all the information from triplets to u_j and update u_j^{l+1} , thus obtaining the representation of u_j at the $(l+1)\,th$ layer. With this, the edge attention score SC^l can be derived as below.

$$SC_{i,k,j}^{l} = \varphi\left(\ u_a^l tanh\left(\ W_h^l h_i^l \parallel W_t^l h_j^l \parallel W_r^l r_k\ \right)\right), \tag{B1}$$

where $W_h^l, W_t^l \in (\mathbb{R})^{d^l \times d^l}$ and $W_r^l \in (\mathbb{R})^{d^l \times d_r}$ are the shared weight of $l\ th$ layer, $h_i^l \in (\mathbb{R})^{d^l}$ indicates the embedding of node i on $l\ th$ layer and $h_i^0 = x_i$, r_k $(\ 1 \leqslant k \leqslant N_r\ )$ indicates the relational embedding of the $k\ th$ relational type, $u_a^l \in (\mathbb{R})^{1\times 3d^l}$, and $||$ stands for the concatenation of embedded vectors.

The attention score matrix SC^l for each edge in graph G can be calculated as follows based on the above definition:

$$SC_{i,j}^{l} = \begin{cases} SC_{i,k,j}^{l},\ \text{if}\ (u_i, r_k, u_j)\ \text{in}\ G \\ -\infty,\ \text{otherwise} \end{cases}, \tag{B2}$$

As a result, when node u_i gives information to node u_j in $l\ th$ layer, the attention matrix AM^l for that layer can be obtained by performing a row-wise sorting by the following equation:

$$AM^l = softmax\left(\ SC_{i,j}^{l}\ \right), \tag{B3}$$

Diffusion of attention in multi-hop nodes. The attention scores obtained above are the scores between nodes with direct connections in the graph, in order to expand the attention scope, we use graph diffusion to compute the attention scores without direct connections, that is, multi-hop nodes, using the following formula:

$$\mathcal{A} = \sum_{i=0}^{\infty} \alpha_i A M^i, \tag{B4}$$

where α stands for the attentional attenuation factor, which has a value greater than 0 and decreases as i increases, but sums to 1.

The feature aggregation Att, based on the diffusion of attention of graph G, can be obtained by computing the attention matrix $\mathcal{A}$ from node u_j to u_i.

$$Att\left(G, H^l, \Phi\right) = \mathcal{A} H^l, \tag{B5}$$

where Φ stands for the set of parameters for attention computation and H^l stands for the embedding of the $l\ th$ level node.

This approach not only increases the receptive domain of attention by increasing the number of node-relational paths, but also sets different attention weights based on the distance of node-relational paths. More importantly, this approach considers both the attention scores between nodes in the previous layer and the path relationships of the nodes, so as to establish attentional associations between nodes that are not directly related. The schematic diagram is shown in Figure B1.

Approximate calculation. Since calculating the power of the attention matrix based on a complex graph structure takes a lot of time and resources, and its accuracy is not that important for us, it can be obtained by approximate calculation. Given that the input to $0th$ layer is $H^0 = X$, its input at lth layer can be denoted as H^l, with an attention decay factor α_i of $\sigma\,(1-\sigma)^i$, $\sigma \in (0, 1]$. Since the model only calculates the aggregation of AH^l, we can define a sequence Z^K to approximate its value, which is $\lim_{K\to\infty} Z^K = AH^l$. This sequence can be used to approximate the calculation of attention diffusion.

$$Z^0 = H^l, \quad Z^{(K+1)} = (1-\sigma)AZ^K + \sigma Z^0, \quad 0 \leqslant k \leqslant K, \tag{B6}$$

Assuming that the BCS graph contains E edges, using the above approximation, a total of $O(|E|)$ messages are communicated, and the constant factor corresponds to the number of hops K, whose computational complexity is $O(K|E|)$. In the experiment, we train the model in a computer carrying an Intel Core i7-6500 CPU and 64G RAM. The amount of data fed into the model per fold is 600, its dimension is 96, and the optimal hop count K is 3. The runtime for a single fold is roughly 72 seconds, and the runtime

for the five-fold cross-validation is about 360 seconds. In clinical practice, since the model has optimized the parameters using the training data, it only needs to operate on the new data, and the decision result is given in about 2 seconds for a single piece of data.

Architecture of multi-hop attention. After calculating the approximate score, we perform deep aggregation by integrating multi-headed attention. Multi head attention simultaneously focuses on features in different search spaces from different perspectives, and the attention diffusion of its head can be calculated as follows:

$$\hat{H}^l = MulHead(G, \tilde{H}^l) = \left(\|_{i=1}^{M} head_i\right) W_0, \tag{B7}$$

$$head_i = Att(G, \tilde{H}^l, \Phi_i)\tilde{H}^l = LayerNorm(H^l), \tag{B8}$$

where W_0 is the parameter matrix, $\|_{i=1}^{M}$ is the multiple connections, denotes the parallel operation of multiple heads from $i = 1$ to M, and Φ_i is the parameter set for the attention computation. Additionally, the two-layer fully-connected feedforward network sublayers are included in the model, with layer normalization and residual connectivity in the middle of the sublayers to capture more expressive aggregations.

$$\hat{H}^{(l+1)} = \hat{H}^l + H^l, \tag{B9}$$

$$H^{(l+1)} = W_2^l \, ReLU\left(W_1^l \, LayerNorm(\hat{H}^{(l+1)})\right) + \hat{H}^{(l+1)}, \tag{B10}$$

The time complexity of the proposed model is concentrated in multi-head attention and graph neural networks, which are determined by factors such as the size of the graph (number of nodes N and edges E), feature dimension D, and the number of layers L. The core of the computational efficiency of the multi-attention mechanism is to compute for each node its attention weight with respect to its neighboring nodes and to aggregate the information. For all nodes, the time complexity of computing the attention score and weighted aggregation is $O(E \cdot D) + O(E \cdot D) = O(E \cdot D)$. The core of the computational efficiency of graph neural networks is the aggregation of neighbor information and node updates through a message passing mechanism with time complexity $O(E \cdot D)$ and $O(N \cdot D^2)$, respectively. Therefore, the total time complexity of the proposed model is $O(L(E \cdot D + N \cdot D^2))$. In the experiment, we used MATLAB 2016a to execute the program in the server, which configuration is: Intel Core i7-6500 CPU and 64G RAM. During cross-validation, the average training time is about 50 seconds per fold of data.

Appendix B.2 Principal component feature extraction

Apart from the attention features, we extracted the principal components of the data to enhance the representativeness of the features. In the experiment, we utilize the Singular Value Decomposition (SVD) method, which can reveal the most essential transformation of the data, to obtain the principal component features of the data [11–13]. For the standardized BCS data matrix A, we can always find a set of unit-orthogonal bases such that the set of vectors obtained after A is transformed to it is still orthogonal, with the following decomposition:

$$A \approx A_d^k = U_d^k \sum_d^k (V_d^k)^T = H^K Z^K, \tag{B11}$$

$$H^K = U_d^k \left(\sum_d^k\right)^{\frac{1}{2}}, \quad Z^K = \left(\sum_d^k\right)^{\frac{1}{2}} V_d^k, \tag{B12}$$

Here, U_d^k is a unitary matrix composed of the eigenvectors of the symmetric matrix AA^T, $\sum$ is a diagonal matrix composed of the square root of the eigenvalues of the symmetric matrix AA^T or A^TA, $(V_d^k)^T$ is a unitary matrix composed of the eigenvectors of the symmetric matrix A^TA, k represents the k-order relationship information of nodes and d represents the dimension of the feature vectors. With the above decomposition, we geometrically realize the operation of rotating, scaling and then rotating the matrix to achieve the purpose of downscaling and extracting the main features from the BCS data.

Appendix B.3 Principal component feature extraction

In this work, we construct the attention feature F_Att and the principal component feature F_Prc. In order to obtain more discriminative features that help the model better understand the built-in structure and regularity of the data, and to improve the performance and generalization of the model, we fuse these two features [14–16]. Since these two features are obtained through different methods and their dimensions are inconsistent, we use feature concatenation for the same samples based on the addition rule to fuse them and give them different weights, which is represented by the following formula:

$$F_{\text{Fus}} = \alpha F_{\text{Att}} \cup (1 - \alpha) F_{\text{Prc}}, \tag{B13}$$

Here, F_{Fus} denotes the fused features, F_Att denotes the attention features, F_Prc denotes the principal component features, and α is the weight factor. We optimized the weight factor in the experiment, and the optimal value after the experiment was 0.6. The specific measurement method is described in the section 'Feature fusion optimization'.

Appendix B.4 Classification prediction

In the prediction of BCS, we experimentally selected the most suitable rotation forest (RF) classifier to perform this task, and the specific experimental procedure is described in section 'Hyper-parameter optimization'. RF is capable of constructing multiple subclassifies with differential and robust properties simultaneously and integrating these results through ensemble approach, resulting in high prediction accuracies [17–19].

Provide the sample set $D(X, Y)$ for BCS, where $X = (x_1, x_2, \ldots, x_n)^T$ is the factor of the sample and $= (y_1, y_2, \ldots, y_n)^T$ is the sample label. The RF classifier constructs a series of decision trees $T_1, T_2, \ldots, T_L$ based on the given hyperparameters L, which results in the generation of L subclassifiers. Each decision tree undergoes the following training: The hyper parameter K is first set based on the dimensionality of the sample D. The value should be a factor of the sample dimensions, dividing the sample into independent subsets on average. Subsequently a new training set $X'_{(i,j)}$ is constructed by randomly selecting 75% of the samples

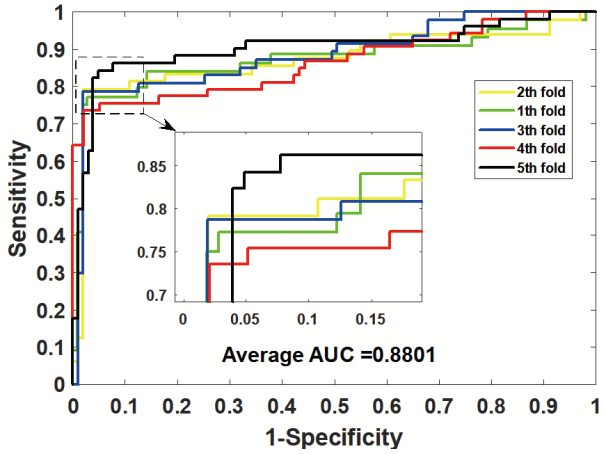

Figure C1 ROC curves of 5-CV obtained by ARM-BCS on BCS dataset.

Table C1 Experimental results of 5-CV obtained by ARM-BCS on BCS dataset

Testing set	Acc. (%)	Sen. (%)	Spe. (%)	F1 (%)	MCC (%)	AUC
1	91.33	77.27	97.17	83.95	78.62	0.8744
2	90.67	75.00	98.04	83.72	78.34	0.8711
3	92.00	78.72	98.06	86.05	81.20	0.8854
4	89.33	73.58	97.94	82.98	76.71	0.8656
5	91.56	82.35	96.12	86.60	80.69	0.9041
Average	90.98±1.04	77.39±3.41	97.46±0.84	84.66±1.57	79.11±1.83	0.8801±0.0152

from the training set via the bootstrap algorithm and assigning them one by one to each of the K decision trees. Then use principal component analysis to generate a coefficient matrix $M_(i,j)$, and rotate it to generate a sparse rotation matrix R_i in the following form:

$$R_i = \begin{bmatrix} r_{i,1}^{(1)}, \ldots, r_{i,1}^{(C_1)} & 0 & \cdots & 0 \\ 0 & r_{i,2}^{(1)}, \cdots, r_{i,2}^{(C_2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & r_{i,k}^{(1)}, \ldots, r_{i,k}^{(C_k)} \end{bmatrix}, \tag{B14}$$

where $r_{(i,k)}^{(C_k)}$ denotes the coefficients of the rotation matrix R_i. Each decision tree T_i makes independent judgments based on the reordered matrix R_i^r when predicting the category of a new sample X, and calculates its score $S_j(x)$ using the embedded learning strategy described below:

$$S_j(x) = \frac{1}{L} \sum_{i=1}^{L} d_{i,j} \left(X R_i^r \right), \tag{B15}$$

Eventually the RF classifier determines the sample category based on the calculated score. In this experiment, we determined the optimal values of hyperparameters K and L through grid search method to be 32 and 15, respectively.

Appendix C Results

Appendix C.1 Model Evaluation

The 5-CV experimental results obtained by ARM-BCS in the clinical dataset are listed in Table C1. From the table, it can be seen that the accuracy achieved in the ARM-BCS five-fold experiment is higher than 89%, with the highest reaching 92%. The average value is 90.98%, while the standard deviation is only 1.04%. In the assessment metrics Sen. and Spe., which reflect the rate of leakage and misdiagnosis in clinical testing, ARM-BCS yielded results of 77.39% and 97.46%, respectively. In the metrics F1, MCC and AUC, which reflect the overall performance of the model, the average results of ARM-BCS are 84.66%, 79.11% and 0.8801, with the standard deviation of 1.57%, 1.83% and 0.0152. The ROC curve acquired by ARM-BCS is presented in Figure C1.

In traditional neural networks, the decision-making process of the model is often black-boxed, and we have no way of understanding how the model arrives at its decisions. In contrast, our proposed ARM-BCS model utilizes the attention mechanism to improve the model's ability to process BCS clinical data by simulating human attentional processes. The model can clearly indicate which features it focuses on during the decision-making process and assign different weights to them. By analyzing the attention weights, we can understand which features of the BCS clinical data the model relies on when making decisions, leading to a more reasonable interpretation. We observed in our modeling runs that the ARM-BCS assigns higher weights to the Age, Neutrophil Count (NEU), Platelet (PLT), Prothrombin Time (PT), Albumin (ALB), Glucose (GLU) and Alpha-Fetoprotein (AFP) factors in its attention assessment of all influencing factors. This suggests that ARM-BCS considers these factors to be at high risk of contributing to the development of BCS.

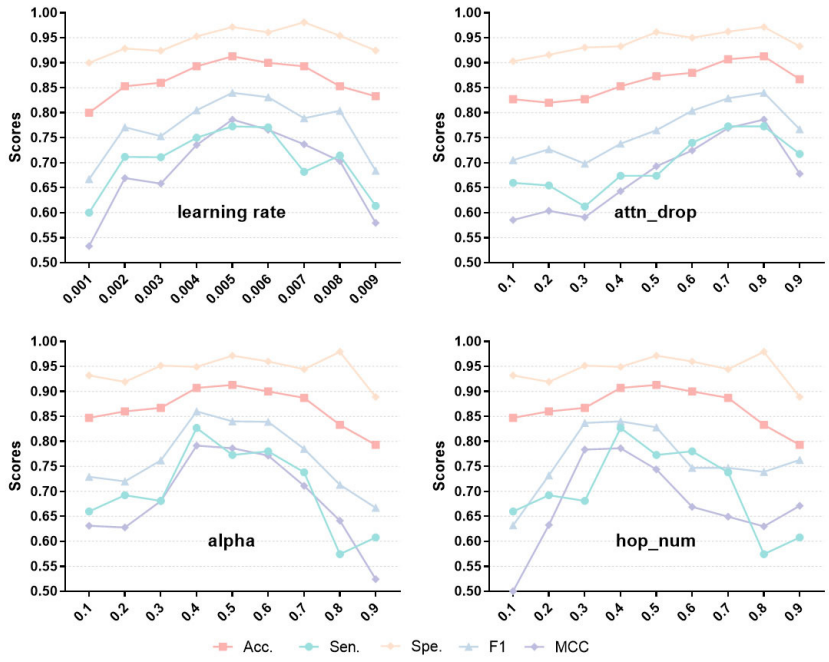

Figure C2 Hyper-parametric optimization of attentional multi-hop for ARM-BCS.

For age, BCS can occur in any age group, but it is more common in younger patients, especially those aged 20-40, and may be associated with hereditary hypercoagulability. And elderly patients may be associated with an acquired hypercoagulable state or anatomical abnormalities. For Neutrophil Count, which may be associated with an inflammatory response to BCS or secondary infection, inflammatory factors may promote thrombus formation by activating endothelial cells and the coagulation system. For Platelet, its elevation may be associated with myeloproliferative tumors, while its decrease may be associated with splenic hyperfunction or liver failure. For Prothrombin Time, it is an important indicator for evaluating liver function impairment and is correlated with disease severity and prognosis. Its prolongation suggests impaired liver synthesis function, which may be associated with chronic liver disease or acute liver failure. For Albumin, it is an important indicator for assessing liver functional reserve and prognosis, and is associated with complications such as ascites and hepatic encephalopathy. Decreased albumin suggests hepatic hyposynthesis and may be associated with chronic liver disease or malnutrition. For Glucose, its abnormalities are associated with liver function impairment or secondary metabolic disorders, indicating metabolic syndrome or liver function impairment, which requires monitoring and intervention to improve prognosis. For Alpha-Fetoprotein, an important marker for monitoring hepatocyte regeneration and HCC, its elevation may be associated with hepatocyte regeneration or hepatocellular carcinoma. By combining the comprehensive analysis of the above indicators, the etiology, pathophysiological mechanisms and prognosis of BCS can be assessed more comprehensively, providing a basis for clinical decision-making and scientific research design.

From a clinical treatment perspective, there are indeed significant differences in the features that ARM-BCS focuses on between BCS recurrent patients and non-recurrent patients. Especially, clinicians can consider them as high-risk factors for BCS based on the focused features given by the model, and thus individualize the diagnosis and treatment of patients in a more targeted manner. From this experimental result, we can see that ARM-BCS not only predicts BCS with high accuracy, but also identifies high-risk factors that may lead to BCS occurrence.

Appendix C.2 Hyper-parameter optimization

The ARM-BCS utilizes attention multi-hop mechanism to capture factors with high-risk effects, thereby improving the prediction rate of the model. The different hyper-parameter settings of the method have a large impact on the model performance, and hyper-parameter optimization experiments are implemented in order for the model to extract the most representative features. We select hyper-parameters of multi-hop graph neural network including learning rate lr, attention drop $attn\_drop$, Laplacian eigenvalues α and number of hops hop_num that have a large impact on model performance for the experiment, and the results are shown in Figure C2. Taking learning rate lr as an example, the hyperparameter is tested from 0.001 to 0.009 with a step size of 0.001. Among the five metrics, their experimental results increase with the increase of learning rate and peaked when it was 0.005 and then gradually decreased. Although the trends exhibited by different metrics are not completely consistent, they generally follow this pattern. Therefore, we combine all the metrics and consider 0.005 as the optimal value for learning rate. Similar analyses are conducted on other hyper-parameters, and the final optimized optimal values of $attn\_drop$ and hop_num are set to 0.8, 0.5, and 4, respectively.

Beyond the attentional multi-hop hyper-parametric, the hyper-parametric of the classifier also play a key role in predicting the outcome. Since the classifier is primarily responsible for the accuracy of classification, we optimize it using accuracy as the evaluation metric. Specifically for rotation forest, the main hyper-parameters are the number of decision trees L and the decomposition factor K of the features. For factor K, we sequentially verify all factors of feature dimension 64, and for number L, we start with 3 decision trees and gradually increase the step size from 3 to 21. The accuracy of the model predictions is illustrated in Figure C3. From the figure, it can be seen that the accuracy on the K-axis first increases with the increase of the factor, and gradually decreases when it reaches 15; On the L-axis, the accuracy roughly increases with the increase of the number of decision trees. Therefore, according to the experimental results, we set the parameters of the rotation forest to 32 and 15, respectively.

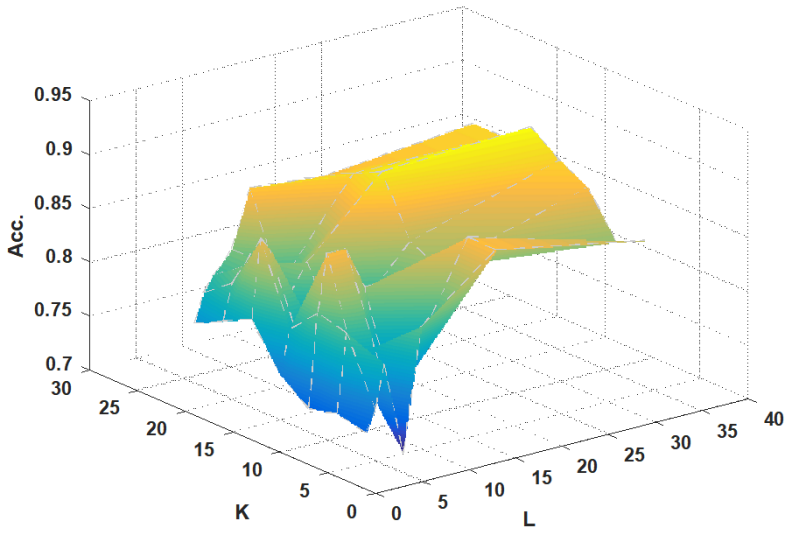

Figure C3 Hyper-parametric optimization of classification prediction for ARM-BCS.

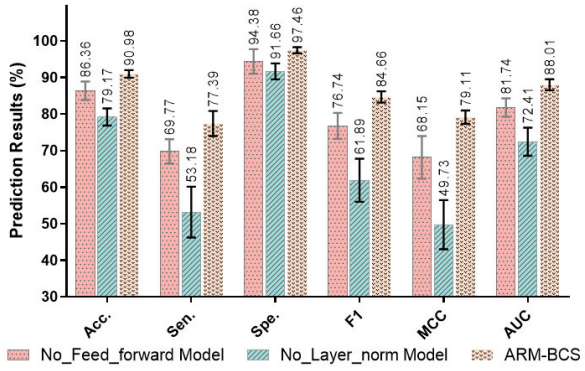

Figure C4 Comparison of experimental results of ablation study.

Appendix C.3 Ablation study

When extracting attentional features ARM-BCS utilizes feed forward and layer normalization to capture deep features. To verify whether these two approaches can improve the model performance, we implemented an ablation study. Specifically, we maintain the overall ARM-BCS framework constant, and only remove the feed forward and layer normalization in the attention feature extraction, thus constructing the No_Feed_forward and No_Layer_norm models, respectively. We then implemented 5-CV validation in the clinical dataset using these two models and compared the results they generated with those of ARM-BCS, and the visualized comparisons are listed in Figure C4. As can be seen from the figure, ARM-BCS achieves optimal results in all evaluation metrics, and the No_Feed_forward model outperforms the No_Layer_norm model. This result indicates that feed forward and layer normalization do contribute to improving the overall performance of the model, with the former providing greater assistance to the model.

Appendix C.4 Feature fusion optimization

In this work, we extracted attention features and principal component features, and fused them to improve the recognition ability of the model. To achieve the optimal effect of the fused features, we validated different weight ratios. More specifically, we stipulate that the sum of weights is 1, and then gradually increase the attention features while gradually decrease the principal component features in steps of 0.1, and the experimental results are presented in Figure C5. As can be seen from the figure, when the ratio of attention features and principal component features is 6:4, its generated radargram can reach the outermost periphery, which means that the evaluation parameters are optimized. The results of this experiment indicates that the fused features represent the essential attributes of the data more effectively and that the attention features are more expressive. Therefore, following the experimental results, we set the feature fusion in the model at this ratio.

Appendix C.5 Comparison with inattentive features

To validate whether the attention mechanism has an enhancing effect on the model performance, we compared it with the inattentive feature model. We experimentally employ graph neural networks without attentional mechanisms to extract BCS features and validate them on the same dataset. The specific comparison results are summarized in Table C2 and Figure C6. As can be seen

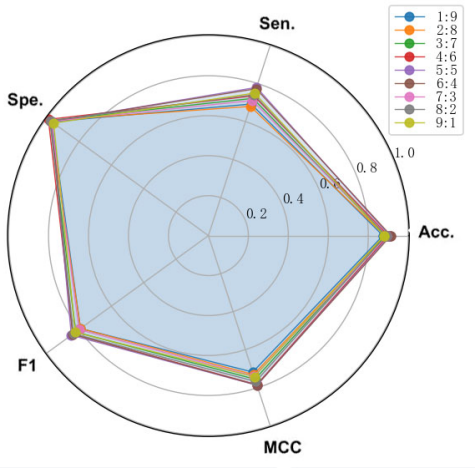

Figure C5 Experimental results generated by fusing features of different proportions.

from the table, the inattentive feature model achieves inferior results to ARM-BCS in all evaluation metrics, and the gap is large. It is 9.54%, 20.70%, 4.12%, 18.38%, 23.36%, and 0.1477% lower than ARM-BCS in accuracy, sensitivity, specificity, F1 Score, MCC and AUC, respectively. Besides, from the perspective of standard deviation, the inattentive feature model is also worse than ARM-BCS. This experimental result indicates that the attention mechanism used in ARM-BCS not only contributes to the improvement of model performance, but also has a significant effect.

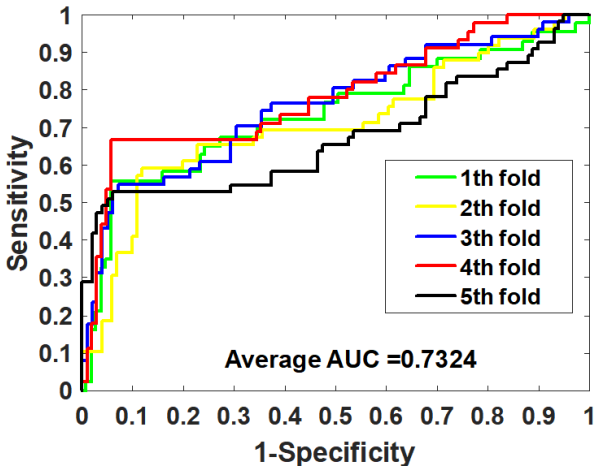

Figure C6 ROC curves of 5-CV gained by inattentive features model.

Appendix C.6 Comparison of various classifier models

We utilize rotation forest in the experiments to classify the predicted BCS results. To estimate the applicability of this classifier to the proposed model, we compare it with different classifier models including random forest (RaF), support vector machines (SVM), NaiveBayes (NB) and K-Nearest Neighbor (KNN), AdaBoost and XGBoost, respectively. In the comparison we only replaced the classifier while keeping the overall framework of the model constant, and the results of the 5-CV comparison are summarized in Table C3. From the table, we can see that ARM-BCS achieves the optimal results in all the evaluated metrics, where the accuracy is screwed by 2.80% than the second highest RaF model, while it is 12.77% higher than the average of the other classifier models. On the AUC, which reflects the overall performance of the model, ARM-BCS exhibits significant competitiveness, which is 0.1663 higher than the average of the other models. This result suggests that the rotation forest classifier we picked can be better applied to the proposed model, which helps to improve the accuracy of the prediction.

Appendix C.7 Comparison of previous methods

To verify the performance of ARM-BCS, we compared it with previous methods. The specific comparison results are summarized in Table C4, where MKSVRB is the method proposed by Xue et al., MKSVRB-RF, MKSVRB XGBoost, and MKSVRB-KNN are variants of this method, representing classifiers based on random forest, XGBoost, and KNN, respectively. AutoML is a method proposed by Yu et al., while AutoML-LR and AutoML-DL are variants of this method, representing logistic regression and deep learning classifiers, respectively. As seen in the table, ARM-BCS achieved the highest value of accuracy among all the methods, which was 15.36 higher than the average score, and the AUC achieved the second highest score, which was only 0.0079 lower than the highest score and higher than the average score of 0.1236. This experimental result shows that ARM-BCS exhibits strong competitiveness compared to other excellent methods.

Table C2 Experimental results of 5-CV obtained by inattentive feature model

Testing set	Acc. (%)	Sen. (%)	Spe. (%)	F1 (%)	MCC (%)	AUC
1	83.33	55.81	94.39	65.75	56.76	0.7359
2	78.67	57.14	89.11	63.64	49.45	0.7086
3	80.00	52.94	93.94	64.29	53.61	0.7590
4	86.00	66.67	94.29	74.07	65.40	0.7841
5	79.22	50.91	94.95	63.64	53.55	0.6742
Average	81.44±3.13	56.69±6.08	93.34±2.39	66.28±4.44	55.75±5.98	0.7324±0.0429
ARM-BCS	90.98±1.04	77.39±3.41	97.46±0.84	84.66±1.57	79.11±1.83	0.8801±0.0152

Table C3 Results of 5-CV gained by various classifier models

Model	Testing set	Acc. (%)	Sen.(%)	Spe.(%)	F1 (%)	MCC (%)	AUC
RaF Model	1	87.33	65.91	96.23	75.32	68.29	0.7993
	2	86.67	68.75	95.10	76.74	68.48	0.8270
	3	90.67	74.47	98.06	83.33	78.04	0.8467
	4	85.33	62.26	97.94	75.00	68.04	0.8323
	5	90.91	76.47	98.06	84.78	79.36	0.8759
	Average	88.18±2.49	69.57±5.90	97.08±1.35	79.04±4.66	72.44±5.73	0.8363±0.0280
SVM Model	1	74.00	40.91	87.74	48.00	32.21	0.5933
	2	72.67	39.58	88.24	48.10	32.05	0.6291
	3	71.33	42.55	84.47	48.19	29.35	0.6457
	4	73.33	41.51	90.72	52.38	38.05	0.6985
	5	77.27	47.06	92.23	57.83	45.58	0.7379
	Average	73.72±2.22	42.32±2.86	88.68±2.98	50.90±4.30	35.45±6.49	0.6609±0.0573
NB Model	1	81.33	61.36	89.62	65.85	53.37	0.7022
	2	73.33	68.75	75.49	62.26	42.38	0.7202
	3	81.33	74.47	84.47	71.43	57.71	0.7943
	4	79.33	64.15	87.63	68.69	53.68	0.7724
	5	82.47	74.51	86.41	73.79	60.62	0.8517
	Average	79.56±3.66	68.65±5.95	84.72±5.49	68.40±4.54	53.55±6.93	0.7681±5.99
KNN Model	1	76.00	29.55	95.28	41.94	34.79	0.5633
	2	71.33	22.92	94.12	33.85	25.07	0.5801
	3	73.33	27.66	94.17	39.39	30.45	0.6232
	4	70.67	37.74	88.66	47.62	31.16	0.6901
	5	75.32	33.33	96.12	47.22	40.39	0.6920
	Average	73.33±2.36	30.24±5.62	93.67±2.92	42.00±5.75	32.37±5.67	0.6297±0.0601
AdaBoost Model	1	79.33	36.36	97.17	50.79	45.90	0.6023
	2	73.33	27.08	95.10	39.39	31.84	0.6350
	3	74.00	25.53	96.12	38.10	32.53	0.6348
	4	71.33	26.42	95.88	39.44	32.79	0.6610
	5	75.97	35.29	96.12	49.32	42.25	0.7085
	Average	74.79±3.03	30.14±5.24	96.08±0.74	43.41±6.11	37.06±6.54	0.6483±0.0396
XGBoost Model	1	72.00	43.18	83.96	47.50	28.94	0.6010
	2	78.67	60.42	87.25	64.44	49.53	0.7443
	3	83.33	70.21	89.32	72.53	60.65	0.7941
	4	82.67	60.38	94.85	71.11	61.24	0.7821
	5	81.82	64.71	90.29	70.21	57.70	0.7777
	Average	79.70±4.66	59.78±10.12	89.13±4.01	65.16±10.34	51.61±13.51	0.7398±0.0798
ARM-BCS		90.98±1.04	77.39±3.41	97.46±0.84	84.66±1.57	79.11±1.83	0.8801±0.0152

Table C4 Scores of Acc and AUC obtained by different methods

Model	ARM-BCS	MKSVRB	MKSVRB-RF	MKSVRB-XGBoost	MKSVRB-KNN	AutoML	AutoML-LR	AutoML-DL
Acc (%)	90.98	78.00	62.00	59.74	61.43	85.70	82.10	85.00
AUC	0.8801	0.8310	0.6610	0.6330	0.6560	0.8860	0.6730	0.830

References

1 Bi X A, Huang Y, Yang Z, et al. Structure Mapping Generative Adversarial Network for Multi-View Information Mapping Pattern Mining. IEEE transactions on pattern analysis and machine intelligence, 2024, 46: 2252-2266.
2 Wang L, Li Z W, Hu J, et al. A PiRNA-disease association model incorporating sequence multi-source information with graph convolutional networks. Applied Soft Computing, 2024, 157: 111523.
3 Ayana, Wang Z, Xu L, et al. Topic-sensitive neural headline generation. Science China Information Sciences, 2020, 63: 1-16.
4 Liang Z, Rong Y, Li C, et al. Unsupervised large-scale social network alignment via cross network embedding. In:Proceedings of the 30th ACM International Conference on Information & Knowledge Management, Queensland, 2021. 1008-1017.
5 Li C, Liu X, Wang C, et al. Gtp-4o: Modality-prompted heterogeneous graph learning for omni-modal biomedical representation. In: European Conference on Computer Vision, London, 2025. 168-187.
6 Li W, Liu X, Yuan Y. Sigma: Semantic-complete graph matching for domain adaptive object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, 2022. 5291-5300.
7 Song L, Wang Z, Yu M, et al. Evidence integration for multi-hop reading comprehension with graph neural networks. IEEE Transactions on Knowledge and Data Engineering, 2020, 34: 631-639.
8 Wang L, Li Z W, You Z-H, et al. MAGCDA: A Multi-hop Attention Graph Neural Networks Method for CircRNA-disease Association Prediction. IEEE journal of biomedical and health informatics, 2023, 1-10.
9 Wu J, Sun C, Yang C. On the size generalizibility of graph neural networks for learning resource allocation. Science China Information Sciences, 2024, 67: 142301.
10 Sun L, Li C, Ding X, et al. Few-shot medical image segmentation using a global correlation network with discriminative embedding. Computers in biology and medicine, 2022, 140: 105067.
11 Gao Q, Xia W, Wan Z, et al. Tensor-SVD based graph learning for multi-view subspace clustering. In: Proceedings of the AAAI Conference on Artificial Intelligence, New York, 2020. 3930-3937.
12 Wei M M, Wang L, Li Y, et al. BioKG-CMI: a multi-source feature fusion model based on biological knowledge graph for predicting circRNA-miRNA interactions. Science China Information Sciences, 2024, 67: 1-2.
13 Shi Q, Cheung Y M, Lou J. Robust tensor SVD and recovery with rank estimation. IEEE Transactions on Cybernetics, 2021, 52: 10667-10682.
14 Zhu W, Zhang H, Eastwood J, et al. Concrete crack detection using lightweight attention feature fusion single shot multibox detector. Knowledge-Based Systems, 2023, 261: 110216.
15 Zhou Q, Qu Z, Wang S Y, et al. A method of potentially promising network for crack detection with enhanced convolution and dynamic feature fusion. IEEE Transactions on Intelligent Transportation Systems, 2022, 23: 18736-18745.
16 Guo L X, Wang L, You Z H, et al. Likelihood-based feature representation learning combined with neighborhood information for predicting circRNA–miRNA associations. Briefings in Bioinformatics, 2024, 25: bbae020.
17 Ganaie M A, Tanveer M, Suganthan P N, et al. Oblique and rotation double random forest. Neural Networks, 2022, 153: 496-517.
18 Wang L, Wong L, You Z H, et al. AMDECDA: Attention Mechanism Combined with Data Ensemble Strategy for Predicting CircRNA-Disease Association. IEEE Transactions on Big Data, 2023, 1-11.
19 Wong L, Wang L, You Z H, et al. GKLOMLI: a link prediction model for inferring miRNA–lncRNA interactions by using Gaussian kernel-based method on network profile and linear optimization algorithm. BMC bioinformatics, 2023, 24: 188.