International Journal for Housing Science and Its Applications
Publish August 6, 2025. Volume 46, Issue 3 Pages 8426-8444

https,//doi.org/10.70517/ijhsa463721

A Multi-Task Financial Fraud Detection Framework Based on Deep Reinforcement Learning Heterogeneous Graph Neural Networks and Variational Autoencoders

Xiaohan Du[1,*]
[1] School of Smart Finance and Economics, Anhui Vocational and Technical College, Hefei, Anhui, 230011, China
Corresponding authors: (e-mail: duxh@uta.edu.cn).

Abstract Financial fraud, as a global problem in the financial industry, brings huge economic losses to financial institutions and customers. In this paper, a multi-task financial fraud detection model is constructed based on heterogeneous graph neural network with deep reinforcement learning, combined with variational self-encoder. In this model, the variational self-encoder is combined with graph convolutional network to construct the node input representation coding module, as a way to enhance the multi-task financial fraud data and better mine the structured features of different nodes. The attention mechanism is then introduced to build the relation-aware attention, which deeply mines the input node features, further acquires the neighbor-generated features of different nodes in the network, and combines the mutual information to measure the nonlinear correlation between different random nodes. Then the financial fraud node representation is mapped into the high-dimensional space by the multilayer perceptron, and then the financial fraud prediction confidence of the model is obtained, and different types of loss functions are set to ensure the detection efficiency of the model. The results show that the F1-macro and AUC values of the financial fraud detection model on the self-constructed FFD dataset are 0.749 and 0.925, respectively.Relying on the heterogeneous graphical neural network and the variational autocoder, a multi-task financial fraud detection model can be constructed, which provides a new idea for solving the suspected fraud and money laundering cases that may exist in the field of finance and economy.

Index Terms heterogeneous graph neural networks, variational self-encoder, attention mechanism, financial fraud detection

I. Introduction

Financial fraud, refers to the use of fictitious facts or concealment of the truth for the purpose of illegal possession, to defraud public and private property or financial institutions of credit, and to undermine the order of financial management [1]. Its impact has been far-reaching in the financial industry, government, business sector and ordinary consumers. Over the past decades, financial fraud has caused alarming losses to the global economy and threatened the efficiency and stability of capital markets, with credit card fraud alone causing hundreds of millions of dollars in lost revenue each year [2], [3]. The relevant security measures of most financial institutions are insufficient, especially for small and medium-sized banks, etc., which causes them to be overstretched when facing fraud risks, which may affect the user experience, or suffer from serious hazards such as loss of funds and leakage of business information [4]-[6].

With the deep integration of financial business and deep learning algorithms, the technical means used by criminal groups and criminals engaged in financial fraud crimes have also been upgraded and iterated, reflecting to the performance of the financial business process, which implies fraud clues and fraud patterns of the detected data, and its covertness has also been increasingly enhanced [7]-[9]. Traditional detection techniques based on expert systems or machine learning are overdue to expose significant limitations due to the need for manual screening and processing of features, which require excessive quality of raw data [10], [11]. In addition, due to the camouflage of the fraudster's behavioral pattern, the financial fraud graph is characterized by non-homogeneity, i.e., the fraudulent nodes will tend to associate with a large number of dissimilar normal nodes in order to hide themselves among the normal users, which violates the assumption of homogeneity of the graph neural network, and leads to the difficulty of existing graph neural networks to deal with the graph data under the financial fraud scenarios, and the accuracy of the detection is unsatisfactory [12]-[16]. Therefore, how to propose fraud detection methods with higher accuracy and robustness for financial frauds with stronger motives, more insidious forms, and more complex data has become a key problem to be solved [17], [18].

Rule-based fraud detection methods aim to utilize expert knowledge to build a detection system that evaluates the likelihood of a potential fraud risk using rules that have been created in advance by experts with relevant knowledge.Ahmed, M. et al. proposed an Intimate Rule-Based (IRB) Financial Fraud Detection and Deterrence Model, which is based on a rich domain knowledge base and ontology-based rule reasoning to classify the severity of a digital fraud level to form a defense against fraud attempts [19]. del Mar Roldán-García, M. et al. established an anti-fraud rule dataset based on anti-fraud rule ontology and semantic rules, which effectively discovers and categorizes semantic conflicts in the execution of the tasks of the underlying expert system, and plays an important role in the field of anti-fraud applications [20]. Hajek, P. developed an automatic fraud detection system that integrates a feature selection component and a rule extraction component, filtering out strongly relevant attributes through feature selection and performing fraud detection based on fuzzy rules, which enhances the interpretability of the model while realizing a high-precision detection target [21]. Yang, F. et al. combined an integrated learning framework with a confidence rule base to establish an online transaction fraud detection model, which effectively handles the highly unbalanced classification tasks, improve the detection performance of the model, and provide good interpretability [22]. However, the expert system can only reduce the inspection burden of inspectors through screening is suitable for simplifying the manual inspection of fraud samples process, but can not replace the inspectors to determine the potential fraud samples.

Many researchers have considered the application of machine learning in fraud detection scenarios, aiming to automate the detection of fraudulent samples through machine learning methods.Ashtiani, M. N. et al. investigated the application of machine learning and data mining techniques in financial statement fraud detection, which is capable of fast processing of large amounts of financial statement data, including textual and audio data to be realized in the future, improving the efficiency of fraud detection [23]. Alghofaili, Y. et al. developed a deep learning fraud detection model for credit card fraud, which introduces a long and short-term memory network to detect fraud threats in order to improve high detection accuracy in a big data environment [24]. Huang, Z. et al. showed that a financial fraud detection model incorporating K-means clustering approach has high detection flexibility and accuracy, which helps to optimize enterprise resource allocation, monitor and prevent high-risk areas, and create a safe and reliable environment for enterprise financial transactions [25]. Bin Sulaiman, R. et al. showed that machine learning techniques show greater potential for application in large-scale fraud detection and prevention tasks, and by embedding neural networks in a federated learning framework to solve the credit card fraud detection privacy preservation and accuracy problem, which is considered as an effective method [26]. Chen, Y. et al. found that integrated learning algorithms have better performance for corporate financial fraud detection, so they established a financial reporting fraud identification model based on stacking algorithm, which reduces the dependence on non-financial data during the detection process, and provides effective decision-making support for managers [27].

Compared with traditional machine learning methods that usually ignore the complex relationships between users because they pay too much attention to their statistical features, graph neural networks pay attention to the relationships between nodes, which can better identify and extract the complex interactions in financial transactions, and play an important role in the field of fraud detection.Innan, N. et al. constructed a financial fraud detection model based on quantum graph neural networks, which utilizes the highly efficient quantum computing power to process graph-structured data, which greatly enhances the detection performance of the financial fraud detection model [28]. Cheng, D. et al. use spatio-temporal attention-based graph network to detect fraud in transaction records, which shows outstanding advantages in discovering problems and mining hotspots compared to other fraud detection models [29]. Tong, G. et al. address the complexity of the transaction data and the hidden nature of fraudulent entities, proposed to introduce a self-attention module in the fraud detection model based on graph neural network, by distinguishing isomorphic and heteromorphic connections between fraudulent nodes, in order to fully utilize the hidden information of the transaction data, and to improve the accuracy of the fraud detection [30]. Zhang, G. et al. examined the role played by a competitive graph neural network model in fraud detection, and the The established fraud detection system can effectively distinguish between fraudulent behavior and normal behavior, and can identify emerging fraudulent behavior and improve it in time [31]. However, some scholars believe that the complexity and heterogeneity of social networks will seriously affect the relationship perception of graph neural network fraud detection model on node information, and the ability of traditional graph neural network to capture node information from multiple perspectives is limited, which in turn faces the problem of reduced detection accuracy [32], [33]. Therefore, based on the existing work experience, this paper proposes a heterogeneous graph neural network based on deep reinforcement learning to construct a multi-task financial detection model to accurately identify fraudsters by improving the performance of abnormal node detection.

Financial fraud has brought many negative impacts to the society, and a variety of artificial intelligence and financial anti-fraud algorithms have been proposed and applied to practical anti-fraud business scenarios in response to financial fraud. In order to more effectively mine and utilize the correlation information between

individuals, this paper proposes a multi-task financial fraud detection model that integrates deep reinforcement learning heterogeneous graph neural network and variational self-encoder. The model mainly includes node input representation encoding, relation-aware attention, and anomaly score discrimination modules. In terms of node representation, this paper utilizes variational self-encoder and graph convolutional network to mine user structured features in nodes, and combines relationship-aware attention to obtain relationship-specific neighbor-generated messages in heterogeneous graph networks. In terms of the anomaly score discriminative model, this paper adopts a multilayer perceptron to map the fused node representation into a high-dimensional space, in order to obtain the model's financial fraud prediction confidence, and then realize the financial fraud detection for multi-task and multi-perspective.

II. Multi-task financial fraud detection modeling

Driven by the wave of digitization, FinTech has risen and reshaped the financial industry at an unprecedented rate. As a key force leading this change, deep learning technology plays a pivotal role in the fintech sector. With excellent data processing capabilities, accurate analysis and optimization of the decision-making process, AI technology has greatly improved the efficiency of financial services, enhanced the customer experience, and injected new vitality into the financial industry. As a global problem in the financial industry, financial fraud has brought huge economic losses to financial institutions and customers. In recent years, deep learning technology has played an important role in financial fraud detection and achieved significant practical results.

II. A.Deep Learning Related Technologies

Through the use of advanced deep learning algorithms, the intelligent system is able to analyze a large amount of transaction data in real time and accurately identify abnormal transaction patterns, so that fraud can be detected and stopped in a timely manner. At the same time, deep learning technology can also be combined with big data analysis technology to verify the identity information of customers, effectively preventing identity impersonation and other fraudulent means. In practical application, various large financial institutions have already constructed intelligent identity verification systems through artificial intelligence and deep learning technology, successfully reducing the risk of fraud and improving customer satisfaction.

II. A. 1) Variational self-encoders

Variational Auto-Encoder (VAE) is a generative network structure based on variational Bayesian inference.VAE models two probability density distributions based on a neural network, whose specific structure is shown in Fig. 1 [34]. One is called the inference network and is used to generate variational probability distributions of hidden variables from the input data. The other is called generative network, which is used to generate approximate probability distributions close to the original input data from the obtained probability distributions of the hidden variables.

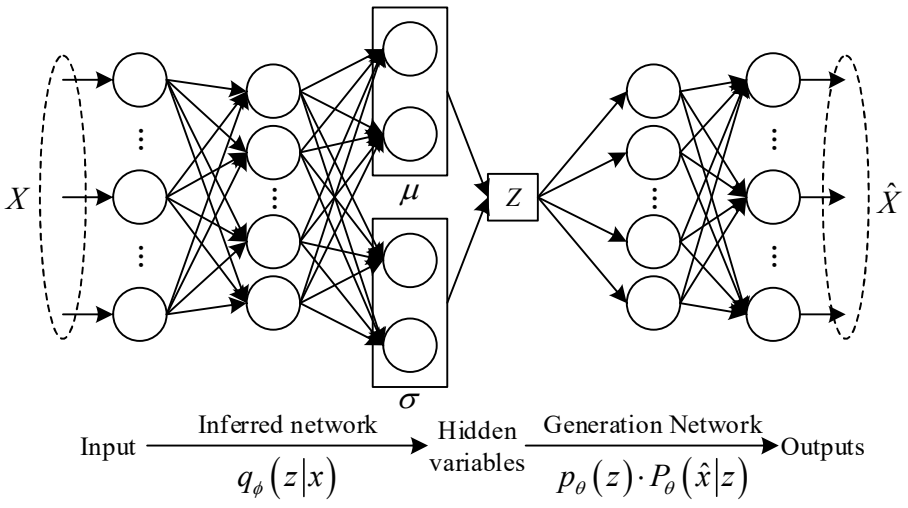


Figure 1: Variable divided self-code structure diagram

where Z is the hidden variable, μ and σ are its mean and standard deviation, X is the original data sample independent of each other, and $\hat{X}$ is the generated data sample. Z is an unobservable random variable that cannot be solved by the maximum iteration algorithm, which is usually assumed z to obey the standard normal distribution, i.e., $P_\theta(z) \sim N(0, I)$. Therefore, $q_\phi(z \mid x)$ is introduced into the inference network in place of the unknown true posterior distribution $P_\theta(z \mid x)$, and $q_\phi(z \mid x)$ is assumed to be known. Then $q_\phi(z \mid x)$ becomes the inference network of the VAE for inferring the approximate distribution of the hidden variables z from the input data x. At the same time, the conditional distribution $P_\theta(\hat{x} \mid z)$ can then be used as a generative network for reconstructing the generation $\hat{x}$ from the hidden variables z. i.e., the training of the VAE model can be divided into two phases, where first, the inferential network infers an approximation of the posterior distribution of z, $q_\phi(z \mid x)$. Then, the generative network generates conditional distributions $P_\theta(z) \cdot P_\theta(\hat{x} \mid z)$ of the variables $\hat{X}$.

The training objective of VAE is to minimize the discrepancy between the input data and the generated data and to optimize the distribution of the hidden variables to approach the a priori distribution. Specifically, the reconstruction ability of the model is improved by minimizing the reconstruction loss, while the distribution of the hidden variables is ensured to be close to the a priori distribution by minimizing the KL dispersion so that the model can effectively learn the data generation process.The VAE loss function is as follows:

$$L_{VAE}(x, \theta, \phi) = -E_{z \sim q_\phi(z \mid x)}\left[\log p_\theta(x \mid z)\right] + KL\left(q_\phi(z \mid x) \| p_\theta(z)\right) \tag{1}$$

The first of these is the reconstruction loss, which measures the difference between the generated data and the real data. The second term is the KL scatter, which measures the difference between the posterior distribution $q_\phi(z \mid x)$ and the prior distribution $p_\theta(z)$ of the inferred network output.

In VAE, both the encoding process and the decoding process are controlled by a parameterized probability density distribution. This approach introduces randomness and uncertainty so that the encoding process is no longer a simple mapping of data, but rather a modeling of the underlying structure of the input data through probability distributions. As a result, VAE not only learns the representation of the data, but also the distributional information of the data, making the generated data richer and more diverse.

II. A. 2) Graph Convolutional Networks

Graph convolutional network is a deep learning model for processing non-Euclidean data, which extends the idea of convolution to graph data by using the local information modeling ability of convolution, and is generally divided into two research directions: spectral domain and null domain. Spectral domain graph convolution is based on filters to process spectral domain signals, and null domain graph convolution is based on convolution kernels to process spatial graphs [35].

The core idea of null domain graph convolution is to use the information about the edges connecting nodes to nodes to aggregate the nodes and thus update the feature information of the nodes. Given an unweighted graph with Q node, where $Q = 1, 2, 3 \cdots q$. matrix X denotes the features of all nodes, and matrix A denotes the adjacency matrix with values 0 or 1.

Assuming that each node information can be inferred from neighboring nodes, the aggregating node is i and the aggregated node is j, the current node information can be simply calculated by summing the neighboring nodes as:

$$aggregate\left(X_i\right) = \sum_{j \in neighbor(i)} A_{ij} X_j \tag{2}$$

where, X_i denotes the characteristics of the i nd node, X_j denotes the characteristics of the j th node, and $j \in neighbor(i)$ denotes that node j is adjacent to node i. If node i is not adjacent to node j, then $A_{ij} = 0$ and hence $\sum_{j \in neighbor(i)} A_{ij} = 0$, then the above equation can be rewritten as:

$$aggregate\left(X_i\right) = \sum_{j \in Q} A_{ij} X_j \tag{3}$$

Consider that most graphs in practical applications are entitled graphs, i.e., A_{ij} can be any value. Therefore it can be expressed in the form of matrix aggregation. Then:

$$aggregate(X) = AX \tag{4}$$

The above equation adjacency matrix A has a diagonal of 0 and only considers the relationship between the node and its neighboring nodes, so the unit matrix I is added to complement the characteristics of the node itself. Namely:

$$aggregate(X) = (A + I)X \tag{5}$$

Simply summing up the features of neighboring nodes will cause some problems of weakness of features of nodes with smaller degree or smaller weights with neighboring nodes, normalized weighted average method can solve such problems. To wit:

$$aggregate\ (X) = \hat{D}^{-1}\hat{A}X \tag{6}$$

where $\hat{A} = A + I$, $\hat{D} = \sum_j \hat{A}_{ij}$ are the degree matrices of $\hat{A}$. However, considering the case where there is a large gap between two neighboring nodes, in addition to the degree $\hat{D}_{ii}$ of node i, the degree $\hat{D}_{ij}$ of node j is equally important. Emphasizing such gaps with symmetrically normalized Laplace matrices, Eqs:

$$aggregate(X) = \hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{\frac{1}{2}}X = \sum_{j=1}^{Q}\left(\hat{A}_{ij} / \sqrt{\hat{D}_{ij}\hat{D}_{ij}}\right)X_j \tag{7}$$

Adding the training parameters and activation function, the final output is:

$$X_{out} = \sigma\left(\hat{D}^{\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}XV\right) \tag{8}$$

where X_{out} is the output feature, σ is the activation function, and W is the weight parameter.

II. A. 3) Heterogeneous graph neural networks

Heterogeneous graph neural networks (H-GNNs) are a neural network architecture specialized in processing heterogeneous graphs, which are a special class of graphs that contain not only multiple types of nodes but also multiple types of edges [36]. In a heterogeneous graph, each type of node and edge can represent different entities and relationships in the real world. It is particularly suitable for dealing with complex data covering multiple types of relationships and multiple types of entities. When modeling multimodal data and its relationships using H-GNNs, it is necessary to determine the characteristics of the multimodal data and map them into a graph structure that contains different types of nodes and edges to represent the diversity of entities and their complex relationships with each other. Therefore, H-GNNs need to model different types of entities and relationships and capture their rich structured information.

When constructing H-GNNs, graph modeling is required first to abstract the multimodal data and its relationships into a graph structure. This step involves determining node types and edge types. Feature extraction is then performed, and for each type of node, the corresponding feature representation is extracted based on its modal properties. For text nodes word embeddings may be used and for image nodes features may be extracted using convolutional neural networks.

In heterogeneous graph neural networks, there are different node fusion strategies and mechanisms need to be introduced to fuse information from different types of nodes. This usually involves designing specific fusion functions f for integrating information between different nodes, i.e.,:

$$h_i' = f\left(h_i, \left\{h_j \mid j \in \mathrm{N}(i)\right\}\right) \tag{9}$$

where h_i denotes the feature representation of node i and $\mathrm{N}(i)$ denotes the set of neighboring nodes of node i in the graph. The fusion function f may be a simple weighted summation, a pooling operation, or a more complex neural network model, where this neighbor aggregation mechanism is to aggregate the information of the node's neighbors into the representation of the current node. A common approach is to refer to the idea of GAT, which calculates the importance of neighboring nodes to the current node and aggregates them. For heterogeneous graphs, this may involve using different attention weights to distinguish between different types of edges as follows:

$$h_i' = \sigma\left(\sum_{j \in \mathrm{N}(i)} \alpha_{ij} W h_j\right) \tag{10}$$

Here α_{ij} is the attention coefficient, which expresses the relative importance of node j to node i, while W is a weight matrix that can be learned to linearly transform the features of the nodes and σ is an activation function.

Multi-step message passing is then required, i.e., multi-step message passing and feature fusion through multiple layers of H-GNNs to capture long distance dependencies between nodes. Each layer can be viewed as a messaging step, allowing node information to span longer distances in the graph. The output layer is designed to process the aggregated node representations for tasks such as node-level classification, link prediction, and graph-level classification, depending on the task.

II. A. 4) Attention mechanisms

The attention mechanism is a method that dynamically focuses on key parts of the input data according to the current task requirements, helping the model to learn and emphasize important features [37]. Its core components include query vectors (Q), key vectors (K) and value vectors (V). The query vector usually represents the current information to be processed and is an important reference for determining the allocation of attention. The key vector is matched with the query vector to calculate the attention score. The value vectors contain the actual information content, and after determining the attention weights, these weights are multiplied with the value vectors to obtain the weighted feature vectors carrying relevance information. Typically, Q is used as an external query, while K and V are computed using the same vectors.

First, the attention score S_i is computed, which reflects how much attention the query vector pays to the key vector. The commonly used calculation function $F(Q,K)$ is shown in the following equation:

$$F(Q,K) = \begin{cases} Q \cdot K \\ W[Q;K] \end{cases} \tag{11}$$

The attention scores S_i are then normalized by the Softmax function, which converts the scores into weight probability distributions α_i. This process, shown in the following equation, ensures that each weight is justified and that they sum to 1. Then:

$$\alpha_i = soft\max(S_i) = \frac{\exp(S_i)}{\sum_j \exp(S_j)} \tag{12}$$

Finally, the normalized weights α_i and Value are weighted and summed to obtain a feature vector carrying key information.

II. B.Financial Fraud Detection Models

The rapid development of mobile Internet technology has led to the emergence of digitalized financial product forms such as third-party payments, online lending and consumer loans, which, while enriching and facilitating people's daily lives, have also brought about greater risks of fraud. The ever-changing fraudulent behavior of financial transactions has brought huge losses to both financial institutions and consumers. Traditional statistical and rule-based risk control methods cannot effectively detect the endless fraud patterns, and deep learning technology provides new ideas for financial fraud detection.

II. B. 1) Model framework design

First, we usually represent a multi-relational graph in the following form, i.e:

$$G = (V, E, A, X, C) \tag{13}$$

where, $V = \{v_1, \ldots, v_N\}$ denotes the set of nodes, N denotes the number of nodes, $E = \{E_1, \ldots, E_R\}$ denotes the set of edges under each of the R relationship graphs, and $A = \{A_1, \ldots, A_R\}$ denotes the corresponding adjacency matrix under each of the R relationship graphs for representing the connectivity between nodes. $X = \{x_1, \ldots, x_n\}$ denotes the set of feature vectors of a node, and $C = \{c_1, \ldots, c_N\}$ denotes the set of labels of a node. For each

node $v_i \in V$, $x_i \in X$ denotes the input feature vector corresponding to that node, and $c_i \in C$ denotes the label corresponding to that node (the fraud label is usually 1 and the normal label is 0).

In addition, we measure the non-homogeneity in the graph using the metric β_r^{fraud}, which is formulated as follows:

$$\beta_r^{fraud} = \frac{1}{\left|\{v_i \mid c_i = 1\}\right|} \sum_{v_i \in fraudsters} \frac{\left|\{v_j \mid v_j \in N_r(i) \,\&\, c_i == c_j\}\right|}{\left|\{v_j \mid v_j \in N_r(i)\}\right|} \tag{14}$$

This metric indicates the average label similarity between fraudsters and their single-hop neighbors under relation r. When β_r^{fraud} is relatively low or even tends to 0, it indicates that the fraudster has successfully camouflaged in this relation graph. A large number of their one-hop neighbor nodes are normal nodes with different labels, i.e., the relationship graph reflects a high degree of non-homogeneous characteristics. On the contrary, if β_r^{fraud} is relatively large or even tends to 1, it indicates that at this point the fraudster is densely connected to his associates and there may be a multi-node operation.

In the financial fraud detection task based on graph convolution, each graph node represents a user, which may be a fraudulent node or a normal node. The purpose of the financial fraud detection model is to find out the fraudulent node among all graph user nodes by judging the attribute information of the nodes and the relationship information of the edges. Combining the H-GNNs and VAE given in the previous section, and then introducing the attention mechanism to construct a financial fraud detection model applied to multi-tasking, its specific structure is shown in Figure 2.

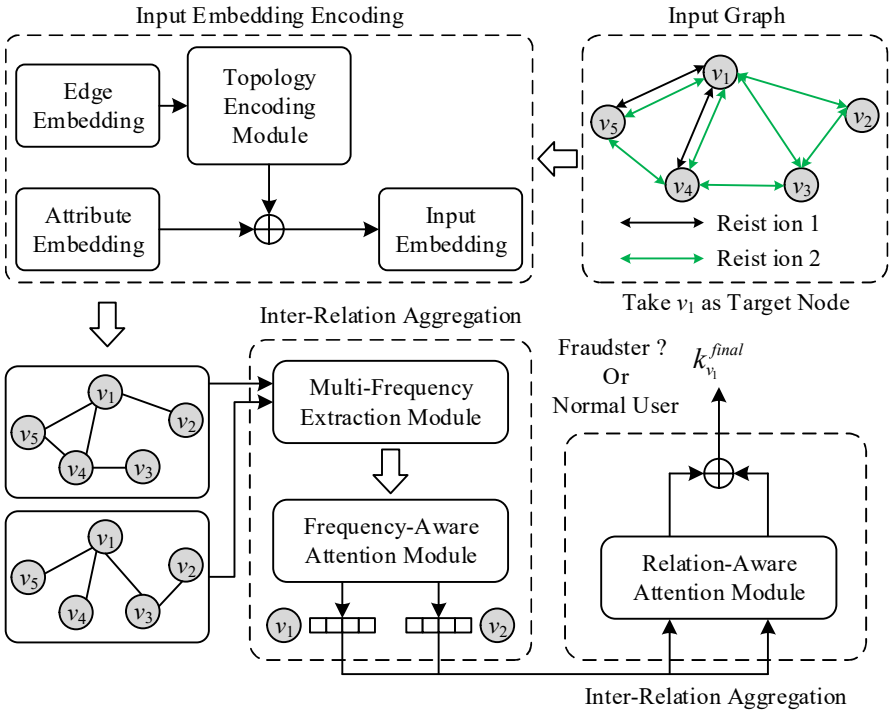


Figure 2: Multi-task financial fraud detection model

The model is mainly divided into three main parts, i.e., node input representation encoding, node representation aggregation within a single relationship, and node representation aggregation between multiple relationships. Specifically, the node input representation encoding process takes the constructed edge representations through the variational graph self-encoding module of the node structural information to obtain the structural information representation of each node, and then splices this structural information representation with the node's original input features as the node's final input representation. The intra-relational node representation aggregation process splits the input multi-relational graph into single-relational graphs for processing, and within each relational graph, the information of different frequencies of neighboring nodes is extracted by the multi-frequency information extraction module. Then the useful frequency information in different frequencies of neighbor nodes is adaptively aggregated by frequency-aware attention module. After this process, each node learns to get its specific representation in each different relationship graph. The inter-relationship node representation process adaptively aggregates the node's

representations in different relationship graphs to get the final representation of the node by calculating the contribution of different relationships to the current node by the relationship-aware attention module. This node representation will be used to determine whether the node is a fraudulent node or not through the final anomaly score discrimination module.

II. B. 2) Variogram self-encoding

In node input representation coding, VAE is combined with graph convolutional network to construct variogram self-encoder as a way to enhance the input data and provide diverse data for the subsequent modules to recognize and sense financial fraud nodes. In the variogram self-encoder, it mainly consists of an encoder for inferring the model and a decoder for generating the model.

We first define an undirected, weightless and acyclic graph $\mathrm{G}=(\mathrm{V},\mathrm{E})$, where V is the set of vertices and E is the set of edges between vertices. Further, define A as the adjacency matrix of G (the corresponding element of A is set to 1 if there exists an edge between two nodes and 0 otherwise). The matrix X introducing $N\times D$ represents the node features, where $N=|\mathrm{V}|$ denotes the number of feature nodes and D is the dimension of the original feature. The random variables z_i of the latent space are obtained by sampling the inferential model and integrated into the matrix Z of $N\times L$, where L is the dimension of the latent features. The inferential model is used to fit an intractable posterior distribution to generate the latent random variables z_i, specifically:

$$q(Z\mid X,A)=\prod_{i=1}^{N}q\left(z_i\mid X,A\right) \tag{15}$$

$$q\left(z_i\mid X,A\right)=\mathrm{N}\left(z_i\mid \mu_i,diag\left(\sigma_i^2\right)\right) \tag{16}$$

Assuming that the posterior distribution q obeys a multivariate mixed Gaussian distribution, μ and σ are regarded as parameters learned by an encoder consisting of a graph-convolution network, and thus $\mu=GCN_{\mu}(X,A)$ and $\log\sigma=GCN_{\sigma}(X,A)$ can be learned and acquired.Similarly, graph-convolution networks can be replaced by graph-attention networks (GATs).

If z_i is sampled directly from q, the problem of calculating derivatives in backpropagation is encountered. As a result, this paper introduces the reparameterization trick, i.e.,:

$$\varepsilon\sim\mathrm{N}\ (0,I)Z=\mu+\varepsilon\square\ \sigma \tag{17}$$

The generative model is mainly used to obtain the reconstruction loss by reconstructing the adjacency matrix, the inner product decoder is used to accomplish this task and is calculated using the following equation. I.e:

$$p(A\mid Z)=\prod_{i=1}^{N}\prod_{j=1}^{N}p\left(A_{ij}\mid z_i,z_j\right) \tag{18}$$

$$p\left(A_{ij}=1\mid z_i,z_j\right)=\sigma\left(z_i^{T}z_j\right) \tag{19}$$

where the adjacency matrix A consists of its elements A_{ij} and σ is the logistic sigmoid function.

Ultimately, the loss function can be described as:

$$\mathrm{L}=E_{z\sim q(Z|X,A)}\left[\log p(A\mid Z)\right]-KL[q(Z\mid X,A)\|p(Z)\Big] \tag{20}$$

where L is often referred to as the lower bound of evidence and consists of two parts: the previous reconstruction loss and the latter regularization constraint. The regularization constraint uses the KL scatter which measures the proximity between $q(\cdot)$ and $p(\cdot)$.

Financial fraud data augmentation is achieved by the variational graph self-encoder in the node input representation coding, which better ensures that the model has access to more diverse training data when carrying out financial fraud detection, thus enhancing the detection accuracy of financial fraud.

II. B. 3) Relationship-aware attention

In the Relationship-Aware Attention module, the following interactions are required for each entity in each node to be able to generate messages from neighbors with specific relationships. Then:

$$m_{(v,n,r)} = \phi\left(h_{v,n}, h_r, \theta_r\right) \tag{21}$$

$$\theta_r = W_r = diag\left(w_r\right) \tag{22}$$

where $\phi\left(e_u, e_r, \theta\right) = \left(\theta_r \cdot e_u\right) - e_r$, $m_{(v,n,r)}$ denote messages aggregated in the n rd node from a neighbor v with relation r, and $\theta_r \in \square^d$ denotes the relation-specific projection matrix, which is restricted to a diagonal matrix for efficiency. To better capture the correlation between nodes and nodes in each component space, an attention mechanism is also utilized to infer the importance of each neighbor in the aggregation, i.e.,:

$$\alpha^k_{(u,v,r)} = soft\max\left(\left(e^k_{u,r}\right)^T \cdot e^k_{v,r}\right) = \frac{\exp\left(\left(e^k_{u,r}\right)^T \cdot e^k_{v,r}\right)}{\sum_{(v',r)\in\hat{N}(u)} \exp^k\left(\left(e^k_{(u,r)}\right)^T \cdot e^k_{(v',r)}\right)} \tag{23}$$

where $e^k_{u,r} = c_{u,k} \circ \theta_r$, denotes the n rd relationship-aware node representation of entity u in a particular relationship-aware subspace. $\hat{N}(u)$ denotes neighboring entities $N(u)$, including u itself. After obtaining the attention scores, the next step is to aggregate the representations of the neighbors in each node separately, and let $h^{l+1}_{u,n}$ denote the n th component representation of entity u obtained after layer l as:

$$h^{l+1}_{u,n} = \sigma\left(\sum_{(v,r)\in\hat{N}(u)} a^n_{(u,v,r)} \phi\left(h^l_{v,n}, h^l_r, \theta_r\right)\right) \tag{24}$$

where h^l_r denotes the representation of the relation r in layer n and is updated by a layer-by-layer linear transformation with parameter W^l_{rel}. Then:

$$h^{l+1}_r = W^l_{rel} \cdot h^l_r \tag{25}$$

In addition, in this paper, mutual information is introduced to measure the nonlinear correlation between different random nodes in the relationship-aware attention module as a way to ensure that different nodes are sufficiently independent from each other, thus helping to realize the full decoupling of entities.

II. B. 4) Abnormal score discrimination

For the node embedding vectors fused by the relation-aware attention module, this paper employs a multilayer perceptron (MLP) to map the fused node representations into a high-dimensional space and ultimately obtain the model's financial fraud prediction confidence. This mapping goes beyond simply integrating data, but transforms information from different perspectives into more expressive features, enabling the model to perform financial fraud detection in a richer and higher-dimensional space.

The MLP achieves a complex mapping from input to output by passing signals between different layers, where each neuron is connected to all neurons in the previous layer and each connection has a weight.The input layer of the MLP receives the fused embedding vectors from the relationally-aware attention module and passes them to the first hidden layer. Each neuron in the hidden layer performs the following operations, viz:

$$h^{(1)}_j = \sigma\left(\sum_{i=1}^{n} w^{(1)}_{ij} \hat{z}_i + b^{(1)}_j\right) \tag{26}$$

where, $h^{(1)}_j$ is the output of the j nd neuron in the first hidden layer, $w^{(1)}_{ij}$ is the weights connecting the input layer to the first hidden layer $b^{(1)}_j$ is the bias of the j th neuron in the first hidden layer and σ is the activation function.

The output of the hidden layer is used as the input of the next layer, and after multiple hidden layers, the output of the output layer is finally obtained. The output of the output layer $\hat{y}$ represents the prediction result of the model, which is calculated as:

$$\hat{y} = \sigma\left(\sum_{j=1}^{m} w_{kj}^{(2)} h_j^{(2)} + b_k^{(2)}\right) \tag{27}$$

where $h_j^{(2)}$ is the output of the j nd neuron in the last hidden layer, $w_{kj}^{(2)}$ is the weight connecting the last hidden layer and the output layer, and $b_k^{(2)}$ is the bias of the k th neuron in the output layer.

With this approach, the complex relationship between multi-view and multi-task information can be captured more accurately and the financial fraud nodes can be recognized and understood effectively in a higher dimensional space, which improves the ability to detect anomalies, and provides a more powerful tool for further deeper understanding of fraud behavior patterns in FinTech.

II. B. 5) Model Loss Functions

In order to better improve the model's detection effect on financial fraud nodes, this paper adopts a strategy that combines multiple loss functions to constrain the training process of the financial fraud detection model, including reconstruction loss, classification loss and regularization constraints. This strategy not only achieves end-to-end training of the model and simultaneous optimization of parameters, but also obtains a more informative embedding representation and neighborhood graph reconstruction. The loss function is specifically defined as follows:

For classification, a cross-entropy loss function is used to measure the difference between the true label Y and the predicted label $\hat{Y}$. The loss function is defined as:

$$Loss_1 = -\sum_{i=1}^{M}\left[Y_i \log\left(\hat{Y}_i\right) + \left(1 - Y_i\right)\log\left(1 - \hat{Y}_i\right)\right] \tag{28}$$

In order to reconstruct the feature data closer to the original data, KL scatter is used to optimize the model by constant iteration of the parameters. The loss function of the node data expression matrix is defined as:

$$Loss_2 = KL\left(soft\max(X), soft\max(\hat{X})\right) \tag{29}$$

where X is the initial node expression matrix of the cell, $\overline{X}$ is the reconstructed node expression matrix, and KL scatter to measure the gap between X and $\hat{X}$.

The loss function for reconstructing the node adjacency matrix $\hat{G}$ is defined as:

$$Loss_3 = MSE(G, \hat{G}) \tag{30}$$

To prevent overfitting, this paper imposes regularization constraints on the parameters, $\varnothing_i$ denotes each parameter in the model, and the regularization term is defined as:

$$Loss_4 = \sum \left\|\varphi_i\right\|_2 \tag{31}$$

Thus, the total model loss loss function is a combination of gene expression reconstruction loss, classification loss, gene adjacency matrix reconstruction loss, and regularization term:

$$Loss = \gamma Loss_1 + \alpha Loss_2 + (1-\alpha) Loss_3 + \theta Loss_4 \tag{32}$$

The financial fraud detection model constructed by heterogeneous graphical neural networks incorporating variational selfencoders is a new end-to-end hypothesis-free framework that employs different integration methods based on different data types and utilizes the integrated node data to construct heterogeneous networks. Finally, multi-task financial fraud detection behaviors such as classification and clustering are performed based on the node representations obtained from the model applied to financial fraud detection analysis.

III. Multi-task financial fraud detection model validation

Deep learning technology to help the development of financial technology industry has brought positive impact to many aspects of society, enterprises can be more efficient organization and management of the capital chain, individuals can more accurately complete the personal financial investment and property management, but the endless financial fraud has also brought huge economic losses for enterprises and people. In order to cope with the rampant financial fraud offensive, a variety of deep learning techniques and financial anti-fraud algorithms have been applied to actual anti-fraud business scenarios and achieved good results.

III. A. Data sources and processing

III. A. 1) Data sources

This paper uses data from the month of May 2023, an international payment company fraud detection screened out the user's data, the total sample has 357,260 account data, fraudulent behavior caused by loss of a total of 8,848 accounts. The purpose is to analyze the special performance found at the beginning of the establishment of the account, analyze the time of account registration, account name, payment performance, payment card similarity, transaction amount range, receipt of goods and other factors for comparison, so as to achieve early discovery, rapid screening, timely processing. Preliminary analysis of all newly registered account information to understand the overall customer registration, whether there is a group of machine registration behavior, whether the first six digits of the added financial information is handled by the same bank, whether the credit card and bank card corresponds to the country of registration, the payment transaction amount of the special amount of money, the total amount of payment, the average monthly transaction amount. Whether there is any short-term quick registration, quick consumption, quick account closure behavior, whether there is any change of account information, whether there is any change of login area. Whether there is a change of account password, whether there is a loss due to card fraud and card refund after payment, and whether there is a manual audit to find out the problems and deal with them. After summarizing the above factors, we analyze the data trend performance by observing the data performance, specific value, and whether there is any specific performance.

The quantity and quality of the data samples are very important for the analysis approach adopted and the final results, in order to apply the data research in the empirical stage, the original data need to be pre-processed and screened. By labeling all user data in the form of [0,1] as a distinction, dealing with redundant data and invalid information appearing in the data, and categorizing the registered accounts as a group sample. Taking time as a reference, factors such as email time, name, address information, first/last addition of the first 6 digits of credit card, first 6 digits of bank card, monthly transaction amount, average daily transaction amount and so on according to the customer's registered email time, name, address information, first/last addition of the first 6 digits of credit card, first 6 digits of bank card, monthly transaction amount, average daily transaction amount and so on will be used as the object of this paper. The financial fraud data produced in this paper is defined as FFD dataset, and in order to ensure the validity of the financial fraud model, this paper additionally selects several types of mature datasets (Elliptic, DGraph-Fin, SCC, CollegeMsg, Worldline) to carry out comparative analysis.

III. A. 2) Oversampling processing

SMOTE algorithm is a random oversampling algorithm to improve the oversampling algorithm, random oversampling method is the idea of direct repetitive sampling of a few classes of samples, although the idea is simple, but it is easy to cause overfitting problems. In order to solve the problem of random oversampling, the SMOTE algorithm proposes the idea of stochastic linear interpolation, which generates new minority samples based on the minority samples.The specific steps of the SMOTE algorithm are as follows:

Step1 Assume that there are n minority class samples in the dataset, denoted as $X_1, X_2, \cdots, X_n$ respectively, and for each sample there are m features, denoted as $X_i = (x_{i1}, x_{i2}, \cdots, x_{im})$, $i = 1, 2, \cdots, n$.

Step2 Obtain the k nearest neighboring similar samples of each minority class sample by calculating the Euclidean distance, for example, the Euclidean distance from minority class sample X_j to X_h is:

$$S_{jh} = \sqrt{\sum_{i=1}^{m} \left(x_{ji} - x_{hi} \right)^2}, j, h = 1, 2, \cdots, n \tag{33}$$

Step3 Set the multiplicity of minority samples N according to the ratio of majority samples to minority samples, and randomly select a number of similar samples from the k immediate neighbors of each minority sample X_i.

Step4 For each randomly selected k near-neighbor minority class sample $\bar{X}_i$, calculate the formula:

$$W_i = X_i + rand(0,1) \times \left| X_i - \bar{X}_i \right| \tag{34}$$

as a new minority class of sample points obtained by random linear interpolation between X_i and $\bar{X}_i$.

The class imbalance of data is a relatively common problem in reality, especially in the process of financial fraud detection, often labeled as fraudulent sample number is much smaller than the number of samples labeled as non-fraudulent. The sample imbalance often has a certain impact on the results of the empirical evidence, by solving the problem of sample imbalance to improve the effect of fraud detection in the traditional financial anti-fraud

methods are more common. In this paper, the SMOTE oversampling algorithm is mainly used in dealing with the class imbalance of data in the FFD dataset.

III. A. 3) Data sampling effects

In this paper, when the SMOTE algorithm is utilized for oversampling financial fraud data, the writing and compilation are implemented in PyCharm, and the experimental environment is AMD Ryzen 7 5800H with Radeon Graphics 2.40 GHz, 32.0 GB RAM, 64-bit operating system, and Windows 10 Flagship Edition.

The experiment starts with 30 ten-fold cross-validation for each dataset, totaling 300 training results, and finally the average of all results is taken. 30% of the real samples are taken as the validation set when performing cross-validation, and the remaining samples are oversampled using a variety of oversampling methods. The oversampling algorithms used include SMOTE algorithm, random oversampling (Random), VTO, Borderline, ASYDMN, and the comprehensive sampling algorithm Tomek as will be mentioned in this paper, and after resampling, the decision tree is used to train the classifier to calculate the classification result index. Choosing F-score and PR-AUC as evaluation metrics, this paper conducts experiments on different unbalanced public datasets and obtains a comparison of the unbalanced processing effect of SMOTE algorithm and other oversampling algorithms as shown in Table 1 and Table 2.

Compared with the Origin dataset without sampling operation, the F-score of the SMOTE algorithm improves by 0.33%, 1.12%, 1.26%, 8.28%, 5.24%, and 0.98% on the six datasets, which proves the effectiveness of the oversampling method proposed in this paper in dealing with the class imbalance of financial fraud data. Compared with other oversampling methods, SMOTE algorithm also has relatively the best oversampling performance, achieving the best classification results on the four datasets with an average ranking of 1.24, which is significantly higher than the remaining four oversampling methods. In addition, in the PR-AUC comparison results of different algorithms, the SMOTE algorithm has higher data grooming effect than all other methods on five datasets, which further proves that the SMOTE algorithm in this paper is able to significantly improve the oversampling effect of data when it is used in the class imbalance processing of financial fraud data.

Table 1: Comparison of the F-score in different data sets

Dataset	Origin	SMOTE	Random	VTO	Borderline	ASYDMN	Tomek
FFD	0.915	**0.918**	0.901	0.911	0.915	0.912	0.914
Elliptic	0.536	**0.542**	0.503	0.528	0.516	0.534	0.525
DGraph-Fin	0.714	**0.723**	0.707	0.714	0.701	0.689	0.698
SCC	0.338	0.366	0.313	0.352	**0.369**	0.351	0.349
CollegeMsg	0.706	0.743	**0.752**	0.739	0.721	0.716	0.724
Worldline	0.915	**0.924**	0.885	0.909	0.905	0.883	0.907
Average ranking	-	1.24	4.13	2.53	4.34	4.17	4.35

Table 2: Comparison of the PR-AUC in different data sets

Dataset	Origin	SMOTE	Random	VTO	Borderline	ASYDMN	Tomek
FFD	0.924	**0.937**	0.931	0.934	0.935	0.935	0.931
Elliptic	0.641	0.651	0.642	0.648	0.629	0.645	**0.659**
DGraph-Fin	0.768	**0.791**	0.775	0.778	0.769	0.752	0.724
SCC	0.549	**0.567**	0.554	0.552	0.558	0.559	0.557
CollegeMsg	0.738	**0.785**	0.762	0.764	0.753	0.749	0.717
Worldline	0.985	**0.996**	0.988	0.982	0.973	0.991	0.994
Average ranking	-	1.35	3.52	3.34	4.17	3.03	4.09

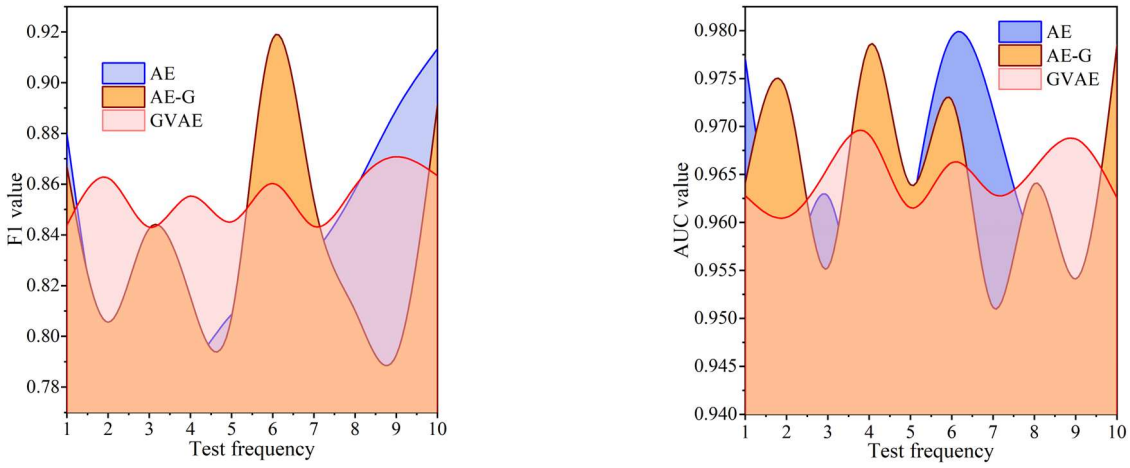
III. B. Effectiveness of financial fraud detection

III. B. 1) Self-Encoder Performance

The financial fraud detection model established in this paper mainly combines the variational self-encoder and heterogeneous graph neural network, in which the variational self-encoder is moreover introduced into the graph convolutional network optimization to obtain the variational graphic self-encoder (GVAE), for the effectiveness of this encoder in carrying out the coding of the node input representations, this paper designs the comparison experiment. Self-encoder (AE) and generative self-encoder (AE-G) are chosen as comparisons with the aim of verifying the effect of model performance enhancement from the regular encoding of self-encoder to GVAE.

We randomly sample 20,000 normal trading samples from the self-constructed FDD training dataset, and use sparse autocoder for the autocoder part, with 60 hidden layer feature dimensions, sparsity constraints are added to the hidden layer units by L1 regularization, and the optimizer uses Adam, with MSE to measure the deviation of the predicted value from the true value, and the value of the Batch Size is set to 256. The training termination does not use a fixed number of epochs, but uses early termination. Instead of using a fixed number of Epochs, early termination is used to stop training when the verification loss fails to decrease in 30 consecutive epochs. The F1 value and AUC value changes during the training process are selected as the evaluation indexes, and the ten-fold cross-validation method is used to observe the training performance changes of different self-encoders as shown in Fig. 3, in which Fig. 3(a)~(b) shows the F1 value and AUC value changes during the training process, respectively. Table 3 shows the evaluation indexes of the detection results of different selfencoders.

It can be seen that the two models, AE and AE-G, have large fluctuations in F1 and AUC values during the training process, while GVAE is relatively more stable, verifying that AE-G uses the original generation of adversarial networks to train the generated data to approximate the original data hidden variables resulting in the model is not easy to converge the training results fluctuate more, and GVAE is mainly the combination of the graphical convolutional network and the variational autocoder, which is designed to generate a different value from the normal GVAE mainly combines graph convolutional network with variational self-encoder, which aims to generate different values from the normal sample hidden variables and the training results are relatively smooth. In financial fraud detection, we are more concerned about the effective identification of fraudulent transactions. Based on the results in the table, it can be seen that the detection effect of GVAE is significantly better than the other two kinds of self-encoders, especially in the more important evaluation indexes of financial fraud detection, recall and F1 value, GVAE performs the best, and the increase of recall will sacrifice the precision rate to some extent. However, for the financial fraud detection problem, recall is more important than precision, because the value of correctly identifying fraudulent transactions is greater than the value of correctly identifying normal transactions. It is worth noting that in practice, during the training process, the average value of the precision of GVAE is slightly lower than that of AE-G, but the GVAE model performs more robustly in the detection phase, which shows that GVAE can better improve the recognition rate of fraudulent transaction samples due to the fact that the generator generates fraudulent transaction samples, and thus the discriminator learns the boundaries with more confidence. Comparing the AE-G and AE models, the self-encoder traditional anomaly detection method only learns normal samples, and the model has a tendency to discriminate the fraudulent samples as normal samples, so the recall and AUC values in the experimental results are low, but the accuracy value is relatively high. The GVAE proposed in this paper integrates the variational self-encoder with the graph convolutional network, using the network to generate “fraudulent samples” to assist in classification training can alleviate the above problems to a certain extent, and the discriminator can improve the identification of fraudulent transactions through the differentiation between the hidden variables of the original data and the generator's approximation of the variables generated by the learning. The experimental results of various indicators GVAE are better than the self-encoder, proving that the performance of GVAE has a greater improvement than the self-encoder.



(a) F1 (b) AUC

Figure 3: Different self-code training performance changes

Table 3: Different self-code detection results

Index	AE	AE-G	GVAE
F1 value	0.756	0.823	0.869
AUC	0.925	0.951	0.972
Recall	0.783	0.771	0.805
Precision	0.719	0.947	0.924
Accuracy	0.991	0.995	0.999

III. B. 2) Model performance comparison

In order to verify the effectiveness of this paper's model in financial fraud detection, GCN, MLP, Graphs AGE, CARE-GNN, PC-GNN, SAGN, SIGN, DAGNN, and AO-GNN models are selected for comparison. Taking the self-constructed FFD dataset of this paper as well as the publicly available dataset DGraph-Fin as the data source, we choose to use AUC and F1-macro values as the model performance evaluation metrics. Table 4 shows the detection performance of this paper's model with all the compared models. It can be seen that the F1-macro and AUC values of this paper's model on the self-constructed FFD dataset are 0.749 and 0.925, respectively, which outperforms all the comparative methods and proves its effectiveness. In addition, through experiments we draw the following conclusions:

(1) Compared with existing algorithmic models such as MLP, GCN and Graphs AGE, this paper's model has a significant improvement in financial fraud detection, which is mainly due to the fact that these models do not take into account the complexity and diversity of fraudulent behaviors in practice. The main reason is that these models do not take into account the complexity and diversity of fraudulent behaviors in practice, in which MLP is a relatively simple neural network structure, its ability to deal with complex graph structure data is limited, and the transaction behavior network relationship is complex, MLP may not be able to adequately capture these complex patterns.GCN model is prone to oversmoothing problems in feature aggregation, making the nodes in the graph converge to the similarity of the feature information, which affects the subsequent classification and detection tasks. Constructing node neighborhoods will be inappropriately sampled according to a fixed number, which may lead to a further increase in the imbalance of node classes, or even a situation where a few classes are completely filtered, thus leading to worse performance in financial fraud detection.

(2) For the algorithmic models for multi-relational graphs such as CARE-GNN and PC-GNN, considering the complexity of the graph structure, their performance is slightly higher than the classical methods, but weaker than the models proposed in this paper. The CARE-GNN algorithm selectively filters the nodes taking into account the possible presence of artifacts and noise in the data, and the PCGNN samples the nodes taking into account the class imbalance in the fraudulent data. In this paper, the model introduces the variogram self-encoder structure on the basis of the above methods, mining node features from a multi-perspective multi-task, further enhancing the information value of the aggregated features, and thus improving the model performance.

(3) The SIGN method avoids the need of graph sampling by different sizes of graph convolution filters, which improves the training efficiency of the model on large graphs.The SAGN method, which can adaptively collect neighborhood information between different jumps, employs a structure-aware attention mechanism instead of the crosstalk operation in SIGN. Both methods are higher than the classical methods in performance, but weaker than the method in this paper. The model in this paper is able to deeply mine the node features in specific fraud scenarios through the relation-aware attention mechanism, enrich the node features through the multi-relational graph feature integration layer, and improve the overall performance of the model.

(4) The AO-GNN method introduces the label distribution insensitive maximization AUC method to deal with the label imbalance problem in the dataset, while the DAGNN method takes into account the camouflage behavior and noise problem that may exist in the real data, and optimizes the GNN model by reducing the noise interference and expanding the channels, etc., which have been improved compared with the previous methods, but all of them are weaker than the model of the present paper. The GNN model is optimized by reducing noise interference and expanding channels. The model in this paper explores the close relationship between different nodes through the anomaly score matching model, which can avoid the negative impact of disguise and noise to a certain extent, thus further improving the accuracy of the model in financial fraud detection.

Table 4: Financial fraud detection performance of different models

Model	FFD datasets		DGraph-Fin datasets	
	F1-macro	AUC	F1-macro	AUC
GCN	0.443	0.568	0.412	0.532
MLP	0.498	0.761	0.436	0.579
Graphs AGE	0.452	0.752	0.515	0.653
CARE-GNN	0.573	0.743	0.493	0.656
PC-GNN	0.642	0.839	0.521	0.669
SAGN	0.686	0.828	0.506	0.652
SIGN	0.704	0.832	0.514	0.664
DAGNN	0.715	0.881	0.509	0.689
AO-GNN	0.723	0.894	0.532	0.693
Ours	**0.749**	**0.925**	**0.575**	**0.716**

III. B. 3) Model ablation experiments

This subsection verifies the effectiveness of three key modules in this paper's model, namely variational graph self-encoder, relation-aware attention, and anomaly score discrimination, through ablation experiments. We construct 3 variants of the method as follows:

A - In the full model, only the variogram self-encoder module is used without adding the relation-aware attention and anomaly score discrimination modules. b - In the full model, only the relation-aware attention module is used without adding the variogram self-encoder and anomaly score Discriminative module.C - In the full model, only the anomaly score discriminative module is used without adding the variogram self-encoder and relation-aware attention modules.

The evaluation metrics chosen are mainly AUC, F1-marco and G-mean, then the results of the model's ablation experiments on different datasets are shown in Table 5.

First of all, from the table, it can be seen that since Model A only uses the variogram self-encoder module without adding the relation-aware attention and anomaly score discrimination modules, which leads to a large gap between the fraudulent and benign class samples during the model training process, the model is difficult to learn the minority class features well. This suggests that the self-encoder module with variogram is necessary in the financial fraud detection task.

Second, the experimental results show that all three performance evaluation metrics of Model B are reduced on both FFD and DGraph-Fin datasets compared to the full model. This is mainly due to the fact that in real financial fraud scenarios, fraudsters usually create many cases of noisy information on purpose in order to avoid detection by the detector, and these noisy information are also one of the reasons for the unsatisfactory detection performance.

Again, from the experimental results, it can also be seen that since Model C only uses the anomaly score discrimination module without adding the variogram self-encoder and the relation-aware attention module, it results in the features between the fraudulent and benign class samples that the model finally learns are not clearly distinguished from each other. The main reason is that the information that can be obtained about the features of the minority classes is quite limited due to the fact that there are too few connections between the minority classes. This shows that the anomaly score discrimination module can effectively improve the detection performance.

Table 5: The experimental results of the experiment

Model	FFD datasets			DGraph-Fin datasets		
	F1-macro	AUC	G-mean	F1-macro	AUC	G-mean
A	0.702	0.872	0.795	0.532	0.683	0.902
B	0.726	0.903	0.816	0.561	0.705	0.921
C	0.713	0.889	0.805	0.554	0.692	0.913
Ours	**0.749**	**0.925**	**0.824**	**0.575**	**0.716**	**0.934**

III. B. 4) Parameter sensitivity

(1) Validation of different model parameters

In order to verify the robustness of the financial fraud detection model constructed in this paper, its performance under different experimental parameters is compared on the self-constructed FFD dataset, and parameter sensitivity experiments are conducted. For the convenience of presentation, only 120 local financial fraud features are used for the experiments. Figure 4 shows the results of different loss function weights on the model performance, and

the shaded area indicates the standard deviation. As can be seen from the figure, the Recall of the model reaches the highest (0.621) when the weight assigned to normal nodes is 0.27 and the weight of fraud nodes is 0.73. The Precision of the node reaches the highest (0.613) when the weight assigned to the normal node is 0.38 and the weight of the fraudulent node is 0.62. And the F1 score of the model reaches the highest (0.595) when the weight assigned to normal nodes is 0.31 and the weight of fraudulent nodes is 0.69. Due to the label imbalance in the dataset, the number of fraudulent nodes is much lower than the number of normal nodes, which will lead to the model leaning more towards the detection of normal nodes during the training process, resulting in the underfitting of the fraudulent nodes, which reduces the model's generalization ability. Therefore, this problem can be mitigated by assigning higher weights to fraudulent nodes in the loss function when dealing with label imbalance data.

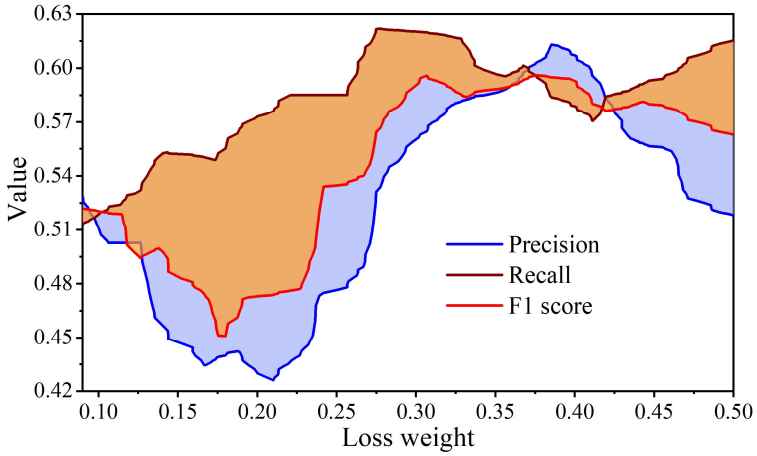


Figure 4: The impact of different loss function weights on model performance

Since the number of updates of the weight parameters of the heterogeneous graph neural network increases with the increase of the number of Epochs, the model effect will become overfitting from underfitting, in order to find out the optimal number of Epochs, experiments on the parameters of the Epochs were carried out. Figure 5 shows the effect of different number of Epochs on the final results, and the shaded area indicates the standard deviation. From the figure, it can be seen that when the number of iterations is 70, 70 and 40 respectively, the F1 score, Precision and Recall of the model have achieved the maximum value of 0.586, 0.573, 0.644 respectively.In summary, in this paper, when the financial fraud detection model training, the number of iterations is selected to be about 60 times, so as to ensure that the F1 score of the model, Precision and Recall are balanced to better improve the financial fraud detection effect.

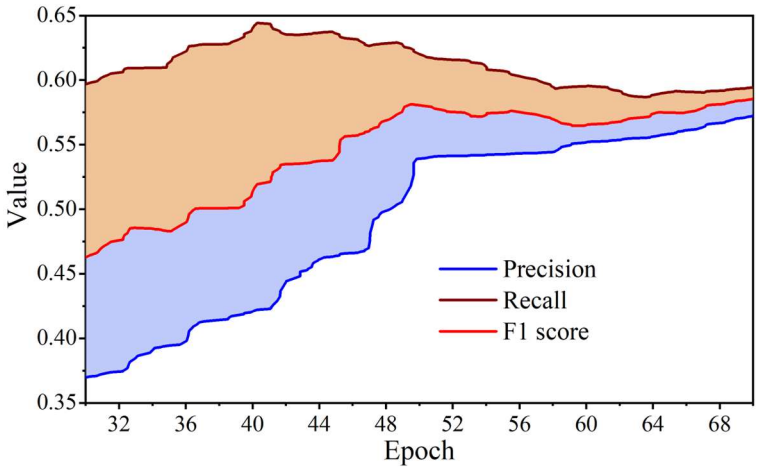
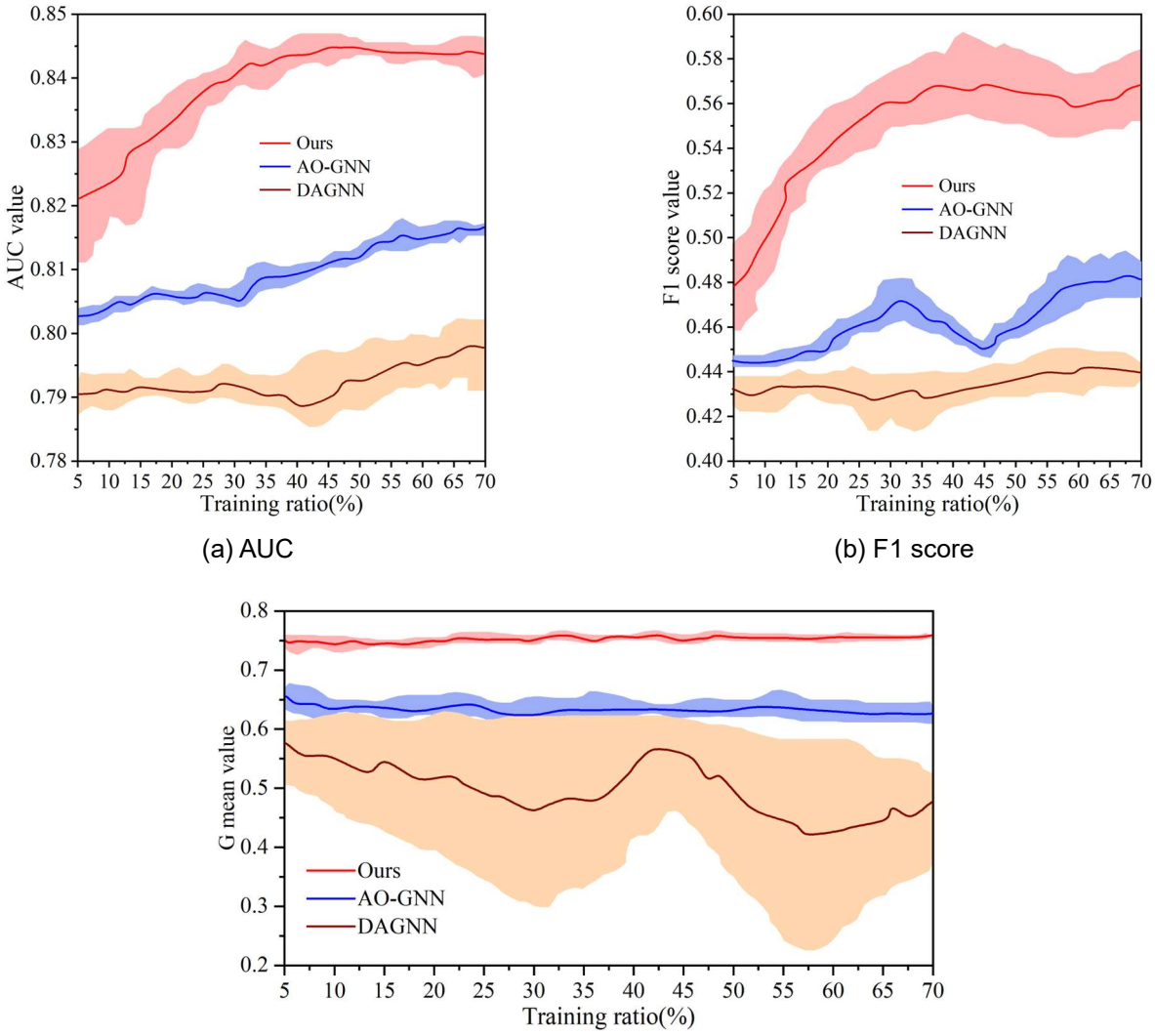


Figure 5: The impact of different number of epochs on model performance

(2) Different training set ratios

In order to further test the financial fraud detection performance of the financial fraud detection model established in this paper under different training set ratios, this paper sets the training set ratio to vary from 5% to 70%, and based on this, compares the model proposed in this paper with the DAGNN and AO-GNN models that have relatively better performance, and its experimental results are shown in Figure 6. Where Fig. 6(a)~(c) shows the experimental results of AUC, F1 score, and G-mean under different training set ratios, respectively, the solid line indicates the average of the performance scores of 10 runs, and the shaded part indicates the standard deviation.

The financial fraud detection model designed in this paper always achieves the best performance under different training set ratios. The AUC scores of this paper's model far exceed those of the DAGNN and AO-GNN models, and steadily improve with the change of training set ratio. In terms of AP and G-mean scores, this paper's model outperforms DAGNN and AO-GNN models by a great advantage, and DAGNN also shows very obvious fluctuations in G-mean scores. Therefore, the financial fraud detection model constructed by combining variational self-encoder and heterogeneous graph neural network in this paper is robust to the change in the proportion of the training set and always leads the DAGNN and AO-GNN models by a great advantage. This also indicates that the model designed in this paper can realize effective financial fraud detection and provide technical guarantee to ensure the high-quality and safe development of fintech.



(a) AUC

(b) F1 score

(c) G-mean

Figure 6: Experiment results under different training set ratios

IV. Conclusion

In the new era of the rise of financial technology, digital technology is the core driving force for the future development of the financial industry, and financial fraud detection based on digital technology has become a new

research hotspot. In this paper, a multi-task financial fraud detection model based on deep learning heterogeneous graph neural network combined with variational self-encoder is proposed, and the validation analysis is carried out by self-constructed FFD dataset. The F1-macro and AUC values of the financial fraud detection model combined with deep learning techniques on the self-constructed FFD dataset are 0.749 and 0.925, respectively, and its performance is significantly better than that of all the compared methods. This indicates that the combination of heterogeneous graph neural network and variational autoencoder can realize effective financial fraud detection and better ensure the healthy development of finance in the digital era.

Funding

This work was supported by the following projects: Research Project on Integration and Application of “Science and Technology + Green Finance” to Enable the Development of New Quality Productivity (2024AH052674); Innovation and Entrepreneurship Ecosystem of Universities and Colleges in the Context of the Construction of the “Three Places and One Region” in Anhui Province to Promote the Development of Regional Economy and High-Quality Economy (SK2021A1026) Path and Countermeasures Research on Economic Development in High Quality Li Wisdom College of Finance and Economics (SK2021A1026).

References

[1] Sun, G., Li, T., Ai, Y., & Li, Q. (2023). Digital finance and corporate financial fraud. International Review of Financial Analysis, 87, 102566.
[2] Hashim, H. A., Salleh, Z., Shuhaimi, I., & Ismail, N. A. N. (2020). The risk of financial fraud: a management perspective. Journal of Financial Crime, 27(4), 1143-1159.
[3] Podkolzina, I. M., Belousov, A. I., Uzdenova, F. M., Romanko, L. V., & Chernikova, O. A. (2019, October). Forms of financial fraud and ways to minimize risks. In Institute of Scientific Communications Conference (pp. 2197-2205). Cham: Springer International Publishing.
[4] Hilal, W., Gadsden, S. A., & Yawney, J. (2022). Financial fraud: a review of anomaly detection techniques and recent advances. Expert systems With applications, 193, 116429.
[5] Zhu, X., Ao, X., Qin, Z., Chang, Y., Liu, Y., He, Q., & Li, J. (2021). Intelligent financial fraud detection practices in post-pandemic era. The Innovation, 2(4).
[6] Tanvir Rahman, A., Md Sultanul Arefin, S., & Md Shakil, I. (2024). Investigating Innovative Approaches to Identify Financial Fraud in Real-Time. American Journal of Economics and Business Management, 7(11), 1262-1265.
[7] Ali, A., Abd Razak, S., Othman, S. H., Eisa, T. A. E., Al-Dhaqm, A., Nasser, M., ... & Saif, A. (2022). Financial fraud detection based on machine learning: a systematic literature review. Applied Sciences, 12(19), 9637.
[8] Hernandez Aros, L., Bustamante Molano, L. X., Gutierrez-Portela, F., Moreno Hernandez, J. J., & Rodríguez Barrero, M. S. (2024). Financial fraud detection through the application of machine learning techniques: a literature review. Humanities and Social Sciences Communications, 11(1), 1-22.
[9] Eswar Prasad, G., Hemanth Kumar, G., Venkata Nagesh, B., Manikanth, S., & Kiran, P. (2023). Enhancing Performance of Financial Fraud Detection Through Machine Learning Model. J Contemp Edu Theo Artific Intel: JCETAI-101.
[10] Minastireanu, E. A., & Mesnita, G. (2019). An Analysis of the Most Used Machine Learning Algorithms for Online Fraud Detection. Informatica Economica, 23(1).
[11] Islam, S., Haque, M. M., & Karim, A. N. M. R. (2024). A rule-based machine learning model for financial fraud detection. International Journal of Electrical & Computer Engineering (2088-8708), 14(1).
[12] Motie, S., & Raahemi, B. (2024). Financial fraud detection using graph neural networks: A systematic review. Expert Systems with Applications, 240, 122156.
[13] Cheng, D., Zou, Y., Xiang, S., & Jiang, C. (2025). Graph neural networks for financial fraud detection: a review. Frontiers of Computer Science, 19(9), 1-15.
[14] Yoo, Y., Shin, J., & Kyeong, S. (2023). Medicare fraud detection using graph analysis: a comparative study of machine learning and graph neural networks. IEEE Access.
[15] Li, R., Liu, Z., Ma, Y., Yang, D., & Sun, S. (2022). Internet financial fraud detection based on graph learning. IEEE Transactions on Computational Social Systems, 10(3), 1394-1401.
[16] Xu, B., Shen, H., Sun, B., An, R., Cao, Q., & Cheng, X. (2021, May). Towards consumer loan fraud detection: Graph neural networks with role-constrained conditional random field. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 35, No. 5, pp. 4537-4545).
[17] Wang, S., & Yu, P. S. (2022). Graph neural networks in anomaly detection. Graph Neural Networks: Foundations, Frontiers, and Applications, 557-578.
[18] Wu, B., Chao, K. M., & Li, Y. (2024). Heterogeneous graph neural networks for fraud detection and explanation in supply chain finance. Information Systems, 121, 102335.
[19] Ahmed, M., Ansar, K., Muckley, C. B., Khan, A., Anjum, A., & Talha, M. (2021). A semantic rule based digital fraud detection. PeerJ Computer Science, 7, e649.
[20] del Mar Roldán-García, M., García-Nieto, J., & Aldana-Montes, J. F. (2017). Enhancing semantic consistency in anti-fraud rule-based expert systems. Expert Systems with Applications, 90, 332-343.
[21] Hajek, P. (2019, May). Interpretable fuzzy rule-based systems for detecting financial statement fraud. In IFIP international conference on artificial intelligence applications and innovations (pp. 425-436). Cham: Springer International Publishing.
[22] Yang, F., Hu, G., & Zhu, H. (2025). A Novel Ensemble Belief Rule-Based Model for Online Payment Fraud Detection. Applied Sciences, 15(3), 1555.
[23] Ashtiani, M. N., & Raahemi, B. (2021). Intelligent fraud detection in financial statements using machine learning and data mining: a systematic literature review. Ieee Access, 10, 72504-72525.

[24] Alghofaili, Y., Albattah, A., & Rassam, M. A. (2020). A financial fraud detection model based on LSTM deep learning technique. Journal of Applied Security Research, 15(4), 498-516.
[25] Huang, Z., Zheng, H., Li, C., & Che, C. (2024). Application of machine learning-based k-means clustering for financial fraud detection. Academic Journal of Science and Technology, 10(1), 33-39.
[26] Bin Sulaiman, R., Schetinin, V., & Sant, P. (2022). Review of machine learning approach on credit card fraud detection. Human-Centric Intelligent Systems, 2(1), 55-68.
[27] Chen, Y., & Wu, Z. (2022). Financial fraud detection of listed companies in china: A machine learning approach. Sustainability, 15(1), 105.
[28] Innan, N., Sawaika, A., Dhor, A., Dutta, S., Thota, S., Gokal, H., ... & Bennai, M. (2024). Financial fraud detection using quantum graph neural networks. Quantum Machine Intelligence, 6(1), 7.
[29] Cheng, D., Wang, X., Zhang, Y., & Zhang, L. (2020). Graph neural network for fraud detection via spatial-temporal attention. IEEE Transactions on Knowledge and Data Engineering, 34(8), 3800-3813.
[30] Tong, G., & Shen, J. (2023). Financial transaction fraud detector based on imbalance learning and graph neural network. Applied Soft Computing, 149, 110984.
[31] Zhang, G., Li, Z., Huang, J., Wu, J., Zhou, C., Yang, J., & Gao, J. (2022). efraudcom: An e-commerce fraud detection system via competitive graph neural networks. ACM Transactions on Information Systems (TOIS), 40(3), 1-29.
[32] Li, E., Ouyang, J., Xiang, S., Qin, L., & Chen, L. (2024, August). Relation-aware heterogeneous graph neural network for fraud detection. In Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data (pp. 240-255). Singapore: Springer Nature Singapore.
[33] Jiang, N., Duan, F., Chen, H., Huang, W., & Liu, X. (2021). MAFI: GNN-based multiple aggregators and feature interactions network for fraud detection over heterogeneous graph. IEEE Transactions on Big Data, 8(4), 905-919.
[34] Hengshan Zhang,Adong He,Jiaze Sun & Yanping Chen. (2025). A large scale group decision making with expert guidance via discrete conditional variational autoencoder. Applied Intelligence(6),437-437.
[35] Xishi Liu,Haolin Wang & Dan Li. (2025). Overseas short video recommendations: A multimodal graph convolutional network approach incorporating cultural preferences. Egyptian Informatics Journal100616-100616.
[36] Tian Li,Shuqi Liu,Guoqing Fan,Hanlin Zhao,Mengmeng Zhang,Jieyu Fan & Changxing Li. (2025). Spatial heterogeneity effect of built environment on traffic safety using geographically weighted atrous convolutions neural network. Accident Analysis and Prevention107934-107934.
[37] Burak Gülmez. (2025). GA-Attention-Fuzzy-Stock-Net: An optimized neuro-fuzzy system for stock market price prediction with genetic algorithm and attention mechanism. Heliyon(3),e42393-e42393.