Robustness Prediction of Complex Networks Based on CNN Improved by Graph Representation Learning Operators

Junhua Li[1], Changxin Xi[2*], Yun Ke[3]
[1]School of Mathematics and Computer Science, Hanjiang Normal University, Shiyan, 442000, China
[2]School of Mathematics and Physics, Jingchu University of Technology, Jingmen, 448000, China
[3]Department of Network and New Media, Wuhan College, Wuhan, 430212, China
E-mail: rachel20240816@163.com, changxinxi@163.com, keyun9405@163.com
*Corresponding author

Keywords: deep learning, graph representation learning, CNN, complex network, robustness prediction

Received: October 12, 2024

Network robustness prediction evaluates the stability and reliability of network systems. A complex network robustness prediction method based on graph representation learning and improved convolutional neural network is proposed to address the low computational efficiency and insufficient prediction accuracy in network robustness. This method introduces prior knowledge of network topology and uses adjacency matrices to extract features of complex networks to improve the efficiency and accuracy of robust prediction. Aiming at the low prediction accuracy, a low frame rate convolutional neural storage area network algorithm based on icon learning is proposed to improve the prediction accuracy and generalization of complex network robustness. The results showed that the proposed algorithm reduced the prediction errors of undirected network robustness by 10.94%, 23.41%, and 13.86% under random attacks, and reduced the prediction errors of weighted network robustness by 0.0041, 0.0043, and 0.0105, respectively. In the robustness prediction of scale-free network and Q-recovery network, the prediction errors of the proposed algorithm were 0.1843 and 0.0278, respectively, reducing by 47.76% and 22.90%, respectively. In the four real networks, including Movie Lens-user, Grid Yeast, C-Elegance, and Polbooks, the connectivity robustness prediction error values of the storage area low frame rate-convolutional neural network algorithm were 0.0906, 0.1106, 0.0715, and 0.1052, respectively, and the controllability robustness prediction error values were 0.5155, 0.1882, 0.0458, and 0.1456, respectively, all of which were superior to existing methods. The proposed algorithm has certain practical application value in the fields of network system design and optimization.

Povzetek: Predlagana metoda za napovedovanje robustnosti kompleksnih omrežij združuje globoko učenje in reprezentacijsko učenje grafov, izboljšano z algoritmom za konvolucijske nevronske mreže, kar povečuje kvaliteto in učinkovitost napovedi.

1 Introduction

With the progress of network systems, predicting the robustness of complex networks has been widely applied in network design and optimization [1]. In the fields of the Internet, neural networks, etc., when the network fails, the robustness of complex network systems can ensure that the network structure does not change [2]. Efficient and accurate robustness prediction of complex networks can maintain the performance of network systems, operating safely and stably in complex environments [3]. However, in large-scale real-world networks, the network has complex nodes, resulting in low operational efficiency and reduced prediction accuracy of network robustness [4]. Therefore, exploring high-precision prediction algorithms and technologies in large-scale network systems has become a difficult research topic [5]. Graph Neural Network (GNN) can optimize networks that learn complex graph relationships. Scholars have conducted extensive research on this topic. Ji Z et al. proposed a representation learning method for molecular graphs, which utilized self-supervised pre-training of GNNs to address the scarcity limitation of task specific labels. The average area under the characteristic curve was 2.6% higher than the baseline model, and the F1 score was improved by 7-18% for drug prediction [6]. Yang X et al. proposed a simple and efficient heterogeneous graph neural network. This method reduced complexity by eliminating excessive neighbor attention and avoiding repeated neighbor clustering during each training period. The results show that this method had a simple network structure, high prediction accuracy, and fast training speed [7]. Heiter E et al. evaluated graph representation learning and graph layout methods in visualization to improve the predictive performance of machine learning models. By comparing graph representation learning and graph layout methods based on distance metrics, the results showed that graph representation learning methods could provide better quality graph layouts [8]. Peng L et al. proposed an unsupervised graph representation learning method to address the limited generalization ability in representation learning. This method maximized

semantic and structural information to design constrained graph representation learning methods. The multi-extraction method was superior to the current more

Table 1: Comparison of methods, dataset, and results of related works.

Method	Dataset	Results	Reference
Self-supervised pre-training method based on graph neural network	MUTAG dataset, BACE dataset	Drug prediction increases F1 value by 7-18%	Ji Z et al. [6]
Simple and efficient heterogeneous graph neural network	Ogbn-mag databases	The proposed method has a simple network structure, high prediction accuracy, and fast training speed	Yang X et al. [7]
Application of evaluation graph representation learning and graphic layout methods in visualization	Molecular graph dataset	Graph representation learning methods provide high-quality graph layouts	Heiter E et al. [8]
Unsupervised graph representation learning method	OGB dataset	Superior to current state-of-the-art methods in different downstream tasks	Peng L et al. [9]
Unsupervised graph representation learning method	PRO TENS dataset	Model performance outperforms popular methods	Mo Y et al. [10]
A prior knowledge routing transfer reinforcement learning algorithm based on generative adversarial networks	Network Topology Dataset	Higher training efficiency across different topologies and network state distributions	Dong T et al. [12]
Inductive reasoning method for network topology	FB15k-237 dataset, WN18RR dataset	Effective prediction of semantic correlation between relationships	Chen J et al. [13]
Learning classification method based on machine learning combined with network prior knowledge	A large number of literature datasets	Evaluating and classifying a large number of literature has demonstrated the reliability of the method	Von RL et al. [14]
Deep neural network model based on prior knowledge of network topology	ImagNete dataset, CIFAR-10 dataset	Improved the robustness of the model	Li X et al. [15]

advanced methods in different downstream tasks [9]. Mo Y et al. proposed an unsupervised graph representation learning method aimed at achieving efficient and effective contrastive learning. This method reduced intra class variation by increasing the upper bound loss to achieve finite distance between positive embedding and anchor embedding. Compared with currently popular methods, this method exhibited better model performance [10].

The prior knowledge of network topology based on deep learning has also been widely applied to evaluate network model performance [11]. Dong T et al. designed a prior knowledge routing transfer reinforcement learning algorithm for generative adversarial networks, aimed at improving training efficiency. This algorithm utilized the routing domain invariant features of deep reinforcement learning. The proposed algorithm had higher training efficiency on different topologies and network state distributions [12]. Chen J et al. proposed a network topology inductive reasoning method to address the semantic relevance issue of existing inductive reasoning models. Network topology patterns were classified and learned to predict inductive links based on different related network patterns. The experiment showed that this method could effectively predict the semantic correlation between relationships [13]. Von R L et al. built a learning classification method based on machine learning combined with network prior knowledge. This method represented different prior knowledge in the learning system, including logical rules and simulation results. The method could evaluate and classify much literature, proving the reliability [14]. Li X et al. designed a deep neural network model on the basis of prior knowledge of network topology, aiming to accurately recognize objects. This model segmented objects from images, evaluated the segmentation results using predefined prior knowledge, and outputted the final prediction results. The method could improve the model robustness [15].

Overall, GNN and network topology prior knowledge on the basis of deep learning have achieved good research results in various fields. However, traditional GNN algorithms have low prediction efficiency and poor prediction performance for network robustness. This study proposes a Multiple Convolutional Neural Network-Region Proposal (MCNN-RP) algorithm based on topological priors to improve the prediction efficiency of network robustness. Aiming at the low prediction accuracy, a Storage Area Low Frame Rate-CNN (SA-LFR-CNN) algorithm is proposed to optimize the prediction accuracy of network robustness. The research innovation lies in the improvement of the traditional CNN-Region Proposal (CNN-RP) algorithm, which utilizes the superior feature learning capability to effectively predict the robustness of complex networks. Then, by combining the Storage Area Network (SAN) operator and MCNN-RP algorithm, a high-precision network robustness predictor is constructed to improve the model generalization and achieve efficient and accurate robustness prediction of complex networks.

The methods, datasets, and results comparison of the relevant research in the literature review are shown in Table 1.

2 Methods and materials

This chapter first constructs an MCNN-RP model to predict the robustness of complex networks by incorporating prior knowledge of network topology. Aiming at the poor universality of robustness predictors, an improved SA-LFR-CNN with SAN operator is constructed to enhance the predictive ability of the predictor.

2.1 Construction of a robustness prediction model for complex networks based on CNN-RP network topology prior knowledge

With the continuous advancement of deep learning, CNN has been extensively applied in network robustness prediction. This method can accurately classify target images and has excellent data processing capabilities [16]. In the processing and analysis of complex networks, the simulation efficiency of robustness is low, and the learning ability of CNN on discrete degree distribution network samples is limited, resulting in lower prediction accuracy of complex network robustness predictors [17]. Therefore, the study adds prior knowledge of multiple network topologies to the predictor. An MCNN-RP algorithm is proposed, which uses adjacency matrix to extract features of complex networks and predicts network robustness through the predictor MLP.

In the research of complex networks, robustness is usually evaluated through two measures: controllability robustness and connectivity robustness. Controllability robustness refers to the degree to which a network's controllability decreases when it is attacked or malfunctioning. Controllability refers to the ability to effectively control a network through control nodes, usually quantified by the density of control nodes. It is used to evaluate the stability of the network in control node failure. Its expression is shown in equation (1) [18-19].

$$n_D = \frac{N_D}{N} \tag{1}$$

In equation (1), n_D signifies the density of control nodes. N_D signifies the total sum of control nodes. N signifies the total nodes. This equation is used to calculate the density of control nodes in a network, reflecting the ability of the network to maintain a controllable state when under attack. The higher the density of control nodes, the stronger the controllability and robustness of the network. The calculation method for controlling the number of nodes includes structural and precise controllability calculations, as shown in equation (2).

$$\begin{cases} N_D = \max\left\{1, N - \left|E^*\right|\right\} \\ N_D = \max\left\{1, N - rank(A)\right\} \end{cases} \tag{2}$$

In equation (2), $\left|E^*\right|$ signifies the maximum number of edges that can be matched in the network. When the network is not completely matched, $N_D = N - \left|E^*\right|$ control nodes are required to complete effective control of the system. When matrix A is full rank, $N_D = 1$ control nodes are required. Otherwise, $N_D = rank(A)$ control nodes are required. The robustness of controllable sequences and edge attacks are defined as equation (3).

$$\begin{cases} n_D^N(i) = \dfrac{N_D(i)}{N-i}, i = 0,1,\ldots,N-1 \\ n_D^E(i) = \dfrac{N_D(i)}{N}, i = 0,1,\ldots,M \end{cases} \tag{3}$$

In equation (3), $n_D^N(i)$ and $n_D^E(i)$ respectively signify the robustness of controllable sequences and the robustness of edge attacks. $N_D(i)$ signifies the number of nodes in the i-th control network after the network is attacked. $N-i$ represents the number of nodes in the i-th remaining network after the network is attacked. M represents the number of times the tolerance for edge attacks reaches network edge connection. Connectivity robustness refers to the change in the size of the maximum connected branch of a network when it is under attack or failure. It is used to evaluate the network's ability to maintain connectivity in node or edge failures. This method is shown in equation (4).

$$S_D = \frac{N_{LCC}}{N} \tag{4}$$

In equation (4), S_D represents the connectivity robustness of the network. N_{LCC} represents the maximum connected branch size of the network. The calculation process for the connectivity robustness of the attack process sequence is shown in equation (5).

$$\begin{cases} S_D^N(i) = \dfrac{N_{LCC}(i)}{N-i}, i = 0,1,\ldots,N-1 \\ S_D^E(i) = \dfrac{N_{LCC}(i)}{N}, i = 0,1,\ldots,M \end{cases} \tag{5}$$

In equation (5), $S_D^N(i)$ represents the network robustness based on edge attacks. $S_D^E(i)$ represents the network robustness against attacks on nodes. $N_{LCC}(i)$ represents the size of the remaining connected branches after attacking the network. This equation quantifies the network's ability to maintain connectivity under continuous attacks. A high value indicates that the network's connections are more robust, indicating that it can maintain significant connections even after being attacked. The CNN-RP in the network robustness predictor is shown in Figure 1. In this network architecture, there are convolutional fast and regressor blocks. The target features are nonlinearly adjusted

through the convolutional layer [20]. In Figure 1, the adjacency matrix is taken as the input part, and the final robustness prediction is taken as the output part. The final connection layer in this structure is used as a predictor to regress the robustness prediction structure.

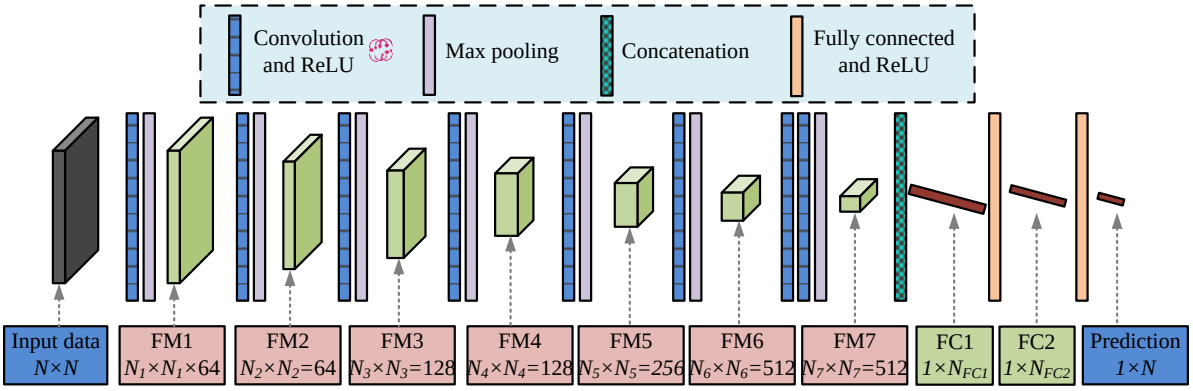


Figure 1: Structure of CNN-RP: Convolutional neural network-region proposal for network robustness prediction.

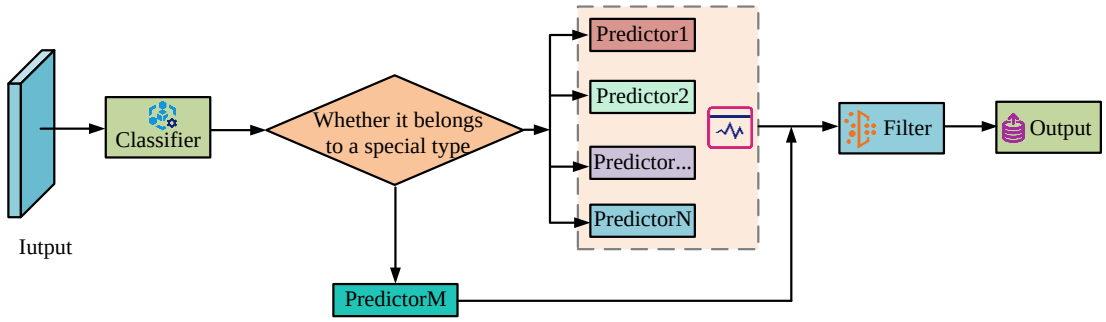


Figure 2: Workflow of MCNN-RP: Multiple convolutional neural network-region proposal for enhanced network robustness prediction.

bounds and local increasing intervals of the network output [21]. The working principle of the filter is shown in equation (6).

$$N_{LCC}(i)=\begin{cases} N-i, if N_{LCC}(i)>N-i \\ 1, if N_{LCC}(i)<1 \end{cases} \quad (6)$$

In equation (6), $N_{LCC}(i)$ represents the existence of an upper bound $N-i$. After i attacks on the network, the maximum value of its connected branches is $N-i$, and the lower bound of the network is 1. This equation is used to limit the upper limit of network robustness prediction results, ensuring that the prediction results conform to the actual characteristics of the network topology. The correction of local increment in network results is shown in equation (7).

$$N_{LCC}(k)=N_{LCC}(i)+\frac{k-i}{i-j}(N_{LCC}(i)-N_{LCC}(j)) \quad (7)$$

In equation (7), $N_{LCC}(k)$ represents a local increase in the robustness curve of the connected network. When $N_{LCC}(k)>N_{LCC}(i),(k\geq i+1)$ occurs, the network output results show a local increase, starting with element k to search for the $j(j=k+1,k+2,\ldots,N)$ -th element, making it satisfy $N_{LCC}(j)<N_{LCC}(i)$, and recording the index j. The value in section $[k+1,k+2,\ldots,j-1,j]$ is corrected. The result of the final filter correction is used as the final output result. The study uses supervised learning to train, validate, and test the CNN-RP model, with its loss function using mean square error, as shown in equation (8).

$$Loss=\frac{1}{N}\sum_{i=0}^{N-1}\left\|S_D^N(i)-\hat{S}_D^N(i)\right\| \quad (8)$$

In equation (8), $Loss$ represents the loss function of the CNN-RP. N represents the scale of the network. $S_D^N(i)$ and $\hat{S}_D^N(i)$ represent the true value and predicted value of the i-th sample, respectively. The MCNN-RP algorithm consists of multiple CNN network structures. The prior knowledge of network topology can preprocess CNNs. After training the model, the network classifier is used to classify each CNN and finally predict the network robustness [22]. The MCNN-RP first initializes multiple CNN models, each with different configurations to adapt to different network topology types. Then, the adjacency matrix is preprocessed using prior knowledge of network topology. The network features are extracted through multiple CNN models. Afterwards, a classifier is used to classify the network topology types and select the most

The output of CNNs violates the prior knowledge of network topology. Therefore, a filter based on the prior knowledge is designed to modify the upper and lower

suitable CNN model. The network robustness is predicted using the selected CNN model and MLP. Finally, the prediction results are corrected based on prior knowledge of network topology. The main workflow is shown in Figure 2. Firstly, the network topology prior input to the classifier is classified, and appropriate CNNs are selected for different types of network topology for prediction.

The classification and predictor in the MCNN-RP algorithm have similar CNN structures, and their feature extraction is also the same. The last layer of the classifier is often used as the $Soft\max(f(Xi)) = e^{Xi} / \sum_j e^{Xi}$ -layer

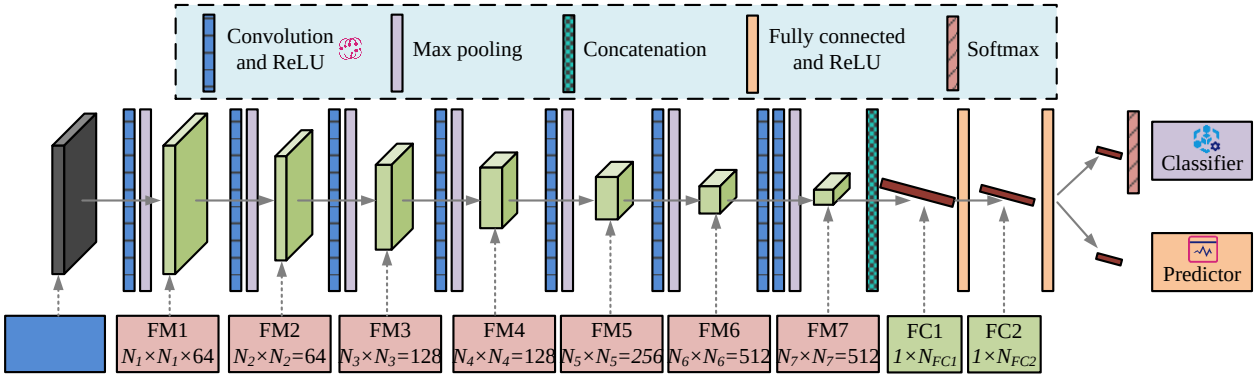


Figure 3: CNN architecture within MCNN-RP: Convolutional neural network structure for robustness prediction.

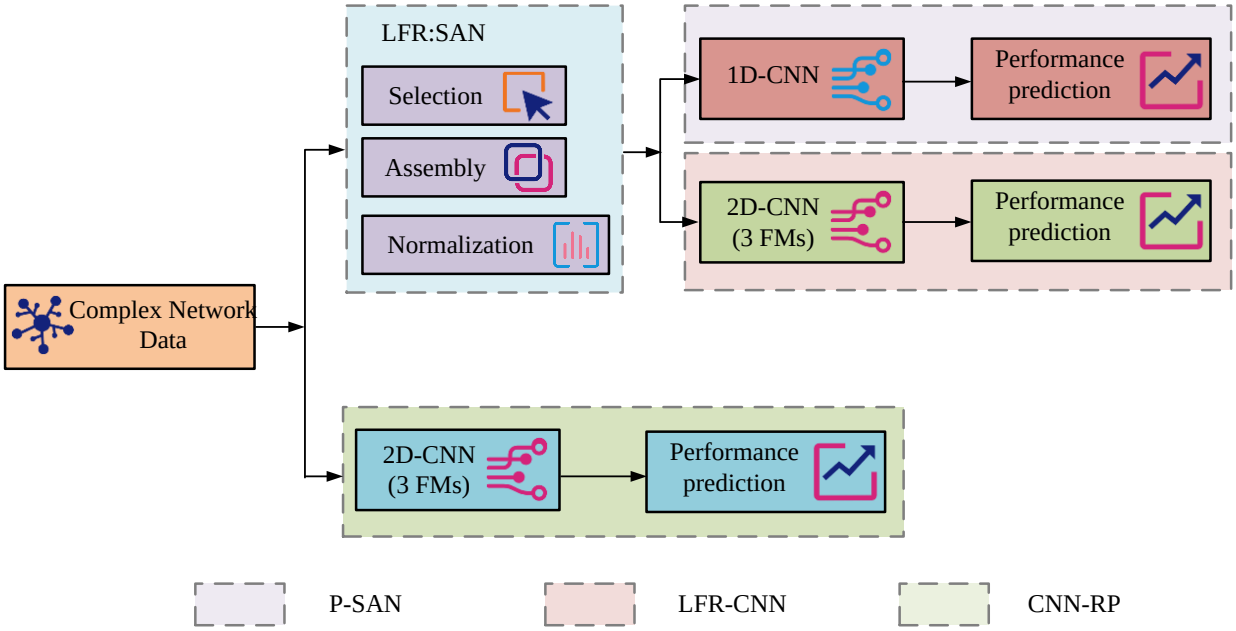


Figure 4: Structure of SA-LFR-CNN: Storage area low frame rate convolutional neural network for robustness prediction.

for classification work. The CNN structure in this algorithm is shown in Figure 3.

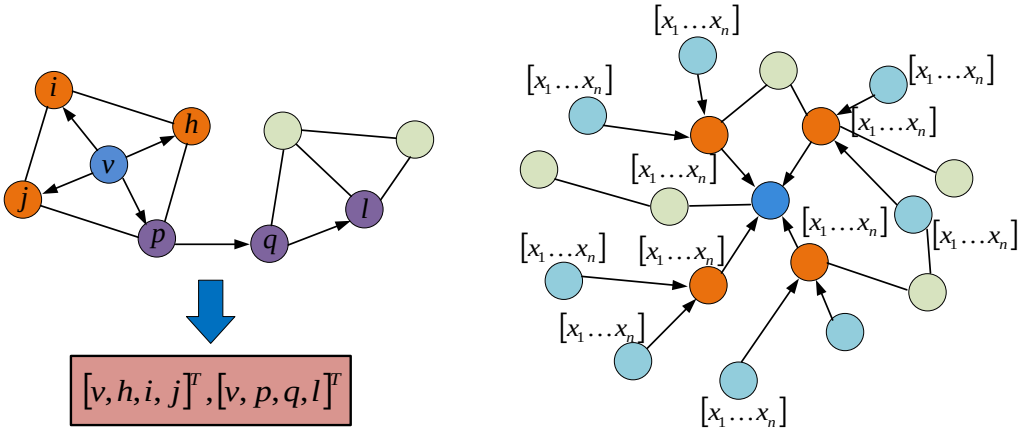
2.2 Robustness prediction of complex networks using graph representation learning algorithm based on improved SAN operator

GNN can effectively process data using graph data feature extraction operators to process graph structures [23]. CNN processing tensors limits the fixed network size, and the generalization ability of robust predictors in complex networks is poor, resulting in lower prediction accuracy [24]. Controllability robustness and connectivity robustness, as important characteristics of complex networks, are closely related to graph learning methods. Based on graph learning methods, the topological features of the network can be extracted more effectively, thereby improving the robust prediction accuracy. In addition, graph learning methods also have important application value in network optimization and design. By predicting the robustness of the network, guidance can be provided for network optimization, helping to design more stable and reliable network structures. A SA-LFR-CNN algorithm is proposed, which combines MCNN-RP and Patchy Storage Area Network (P-SAN) algorithm to improve the operators in GNN algorithm. It adds operators based on graph

representation learning to extract target graph features and then predicts the network robustness.

The SA-LFR-CNN first initializes the improved CNN model, combined with the SAN operator and graph representation learning module. The prior knowledge of network topology is used to preprocess the adjacency matrix. Then, the graph representation learning module learns the graph representation of the network, samples nodes from the graph representation, and constructs subgraphs for normalization. Subsequently, the CNN model is used to extract features from the normalized subgraph, and the extracted features are used to predict the network robustness. Finally, the prediction results are adjusted based on prior knowledge of network topology. The structure is displayed in Figure 4.

The SA-LFR-CNN algorithm represents the target features, normalizes the sampled nodes and constructed subgraphs, and then splits the network. The split graph is used as a feature node input to the algorithm, and the target features are extracted through a CNN. Finally, the network robustness is subjected to regression prediction [25]. The loss function of this algorithm first aggregates the structural information, and trains the designed loss function on the structural information. The basis of each node aggregation is related to the location of the layers. The central node trains the nodes through two aggregations to obtain feature data. The specific process of obtaining node feature information is shown in Figure 5.



(a) Diagram structure information sample diagram

(b) Aggregation graph structure and node feature information

Figure 5: Graph structure and node feature information: Schematic of graph representation learning for robustness prediction.

In the supervised learning, the algorithm uses a designed loss function to capture the first-order and second-order distance information of the graph. The first-order distance joint probability distribution is shown in equation (9).

$$p_1(v_i, v_j) = \frac{1}{1+\exp(-\vec{u}_i^T \cdot \vec{u}_j)} \tag{9}$$

In equation (9), $p_1(v_i, v_j)$ represents the joint probability of node v_i and node v_j. $\vec{u}_i \in R^d$ represents the representation vector of the node in low latitude space. This equation is used to calculate the similarity between nodes, reflecting their proximity in low dimensional space. In this way, first-order distance information between nodes can be captured, thus better understanding the topology of the network. In the training process, the confidence in the first-order distance structure is captured directly by minimizing the objective function, as shown in equation (10).

$$O_1 = d(\hat{p}_1(\cdot,\cdot), p_1(\cdot,\cdot)) \tag{10}$$

In equation (10), O_1 represents the objective function of first-order distance. $d(\cdot,\cdot)$ represents the distance between probability distributions $\hat{p}_1$ and p_1. By calculating the square difference between the true joint probability and the predicted joint probability, and adding up all nodes and their adjacent nodes, the prediction error of the model on the first-order distance is quantified. The obtained objective function is shown in equation (11).

$$O_1 = \sum_{(i,j)\in E} \omega_{ij} \log p_1(v_i, v_j) \tag{11}$$

In equation (11), $\omega_{ij} \log p_1(v_i, v_j)$ represents the distance between the relative entropy probability distributions. Nodes play different roles in second-order distance, and the second-order distance needs to be calculated using different representation vectors. For each edge $(i,j) \in E$ in the graph, the probability of the context node is shown in equation (12).

$$p2(v_j \mid v_i) = \frac{\exp(\vec{u}_j'^T \cdot \vec{u}_i)}{\sum_{k=1}^{|V|} \exp(\vec{u}_j'^T \cdot \vec{u}_i)} \tag{12}$$

In equation (12), v_i represents the source node. v_j represents the context node. $\vec{u}_j'^T$ and $\vec{u}_i$ represent the representation vectors of the context node and the source node, respectively. $|V|$ signifies the number of all nodes. $p_2(v_i, v_j)$ represents the probability of obtaining the context node v_j. Based on above methods, the second-order distance information between nodes can be

captured, thus better understanding the global topology of the network. The objective function for capturing second-order distance structure information is displayed in equation (13).

$$O_2 = \sum_{i \in V} \lambda_i d(\hat{p}_2(\cdot|v_i), p_2(\cdot|v_i)) \tag{13}$$

In equation (13), O_2 represents the objective function of second-order distance. $d(\hat{p}_2(\cdot|v_i), p_2(\cdot|v_i))$ represents the distance between probability distributions $\hat{p}_2$ and p_2. λ_i represents the weight of node p_2. By minimizing this objective function, this equation can optimize the low dimensional representation of nodes. Therefore, the model can better capture the indirect connection relationships between nodes in the network, thereby

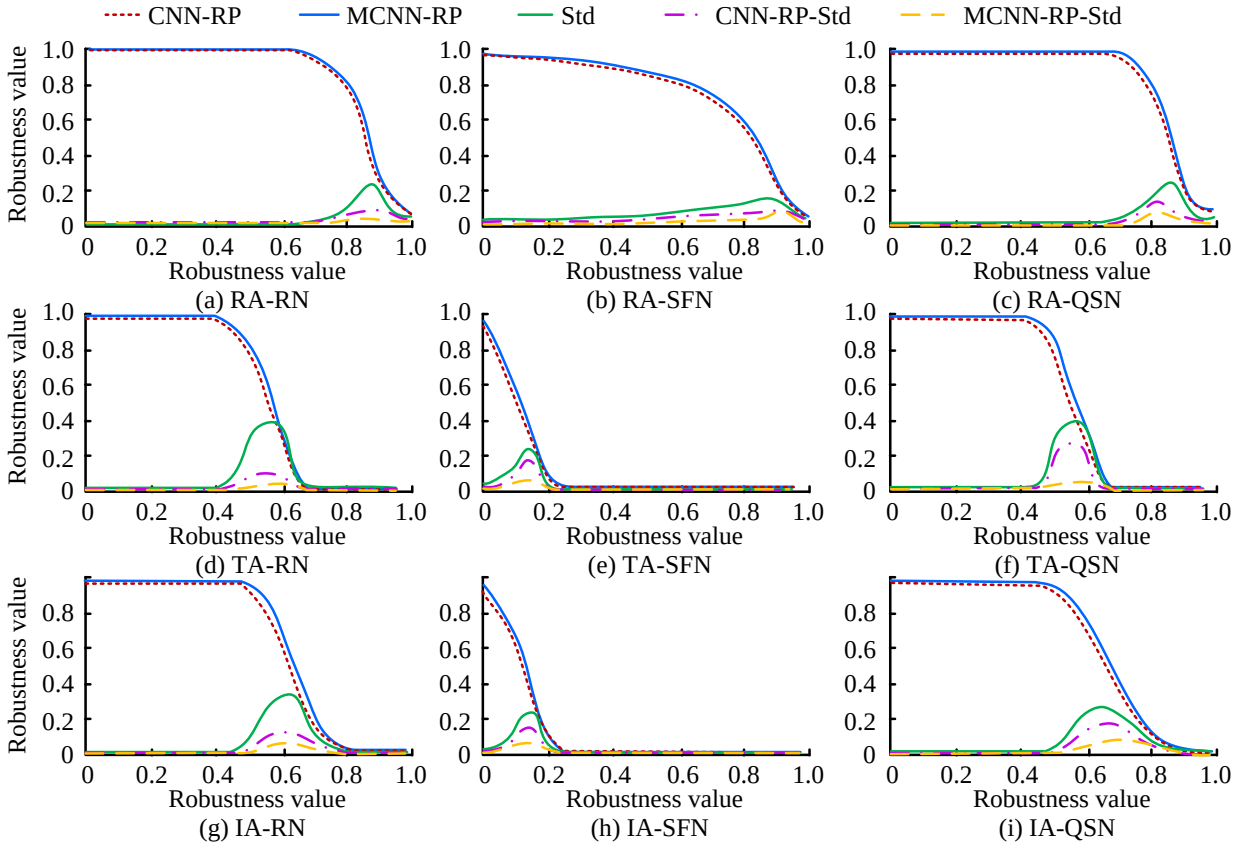


Figure 6: Prediction results of CNN-RP and MCNN-RP algorithms for network connectivity robustness.

improving the accuracy of robust prediction. The empirical distribution $\hat{p}_2(\cdot|v_i)$ is displayed equation (14).

$$\hat{p}_2(v_j|v_i) = \frac{\omega_{ij}}{d_i} \tag{14}$$

In equation (14), ω_{ij} represents the weight of edge $(i.j)$ in the graph. v_i represents the degree of the node. By introducing empirical distributions, the connection relationships between nodes can be more accurately captured, thereby improving the model's understanding of network topology. To better capture second-order distance information, the study uses relative entropy to represent the distance of node probability distribution, as shown in equation (15).

$$O_2 = \sum_{(i,j) \in E} \omega_{ij} \log p_2(v_j|v_i) \tag{15}$$

In equation (15), $\omega_{ij} \log p_2(v_i|v_j)$ represents the second-order distance between the relative entropy probability distributions. Relative entropy not only considers the absolute differences in probability distribution, but also considers the relative differences in probability distribution. Therefore, it can more accurately reflect the prediction error of the model. After obtaining information on node features, the robustness of complex networks is predicted.

3 Results

This chapter analyzes the network robustness prediction performance of the improved topology prior MCNN-RP algorithm through three attack methods, verifying the effectiveness. Aiming at the robustness and generalization of complex networks, a graph representation learning algorithm based on SA-LFR-CNN is analyzed to verify the robustness prediction performance of complex networks.

3.1 Analysis of robustness prediction performance of complex networks

based on improved CNN-RP network topology prior knowledge

The experiment uses three network models, Random Network (RN), Scale Free Network (SFN), and Q-Snapback Network (QSN), to generate network samples. Supervised learning is used to train the MCNN-RP network topology model. Three attack methods are set, including Tree Attack (TA), Intermediate Attack (IA), and Random Attack (RA). In the RN, SFN, and QSN datasets, their sizes all contain 1,000 nodes. For RN, the connections between nodes are randomly generated and do not have a specific topology structure. SFN follows a power-law distribution, with a few definitions having a large number of connections, and a large number of nodes having only a few connections. It has a clear

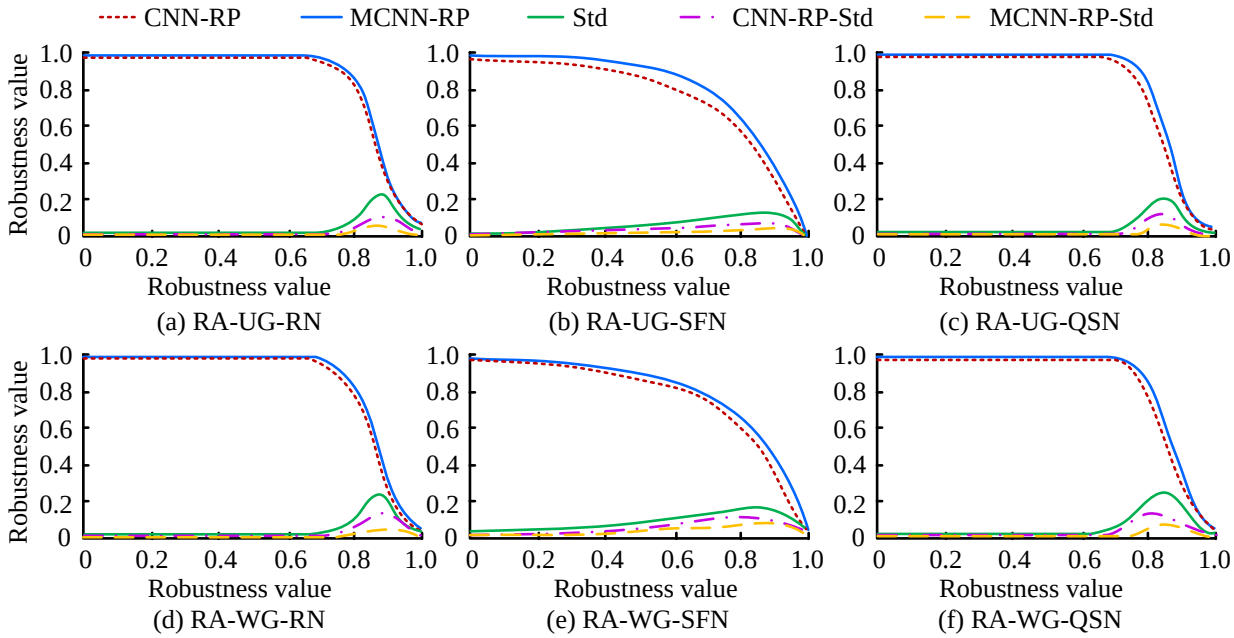


Figure 7: Network robustness prediction results of undirected graphs and weighted graphs under random attacks.

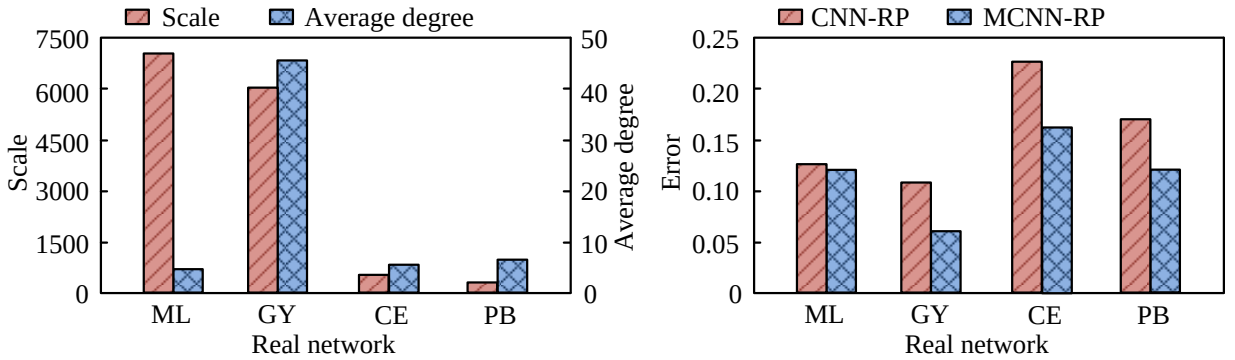
central node, and the network has a small world characteristic with shorter paths. For QSN, it has Guzan's dynamic characteristics, where the connection relationships of nodes change over time. Due to the dynamic changes in connection relationships, the robustness of the network is highly uncertain. During the training process, the average degrees of the training and testing samples are 6.01 and 5.49, respectively. The training set is 6,400, and both the validation and test sets are 1,600. The operating system for this experiment is Ubuntu 20.04 LTS, the deep learning framework is PyTorch 1.8, and the programming language is Python 3.8. The model learning rate is set to 0.001, the batch size is 32, and the number of training rounds is 100.

The prediction error and robustness prediction accuracy are selected to measure the predictive performance of the network. The prediction error directly reflects the accuracy of the model prediction. The lower prediction error indicates that the model can more accurately predict the robustness of the network after being attacked. In practical applications, accurate robustness prediction can help network optimizers better understand the vulnerability of the network and take effective measures to enhance its stability. The accuracy of robustness prediction reflects the reliability of the model in predicting network robustness. The higher prediction accuracy indicates that the model can more accurately capture the topology and dynamic characteristics of the network, providing more valuable references for network optimization.

The experiment conducts three types of attacks on three networks and compares the connectivity robustness prediction results of CNN-RP and MCNN-RP, as shown in Figure 6. Under random attacks, the CNN-RP algorithm for generating network samples from RN, SFN, and QSN had errors of 0.0332, 0.0501, and 0.0384 in predicting the robustness of complex networks, respectively. The MCNN-RP algorithm had prediction errors of 0.0312, 0.0454, and 0.0335 for network robustness, respectively. Under degree attack, the prediction errors of the CNN-RP for the robustness of the three network samples were 0.0532, 0.0224, and 0.0503, respectively. Compared with the CNN-RP, the prediction errors of the MCNN-RP were reduced by 0.0015, 0.0125, and 0.0040, respectively. Under the betweenness centrality attack, the CNN-RP algorithm for RN, SFN, and QSN network samples had prediction errors of 0.0566, 0.0287, and 0.0574 for the robustness of complex networks, respectively. Compared with CNN-RP, the prediction errors of the MCNN-RP algorithm were reduced by 6.53%, 53.65%, and 1.05%. The prediction error of MCNN-RP algorithm is significantly lower than that of CNN-RP, which proves that the algorithm has

good predictive performance for the robustness of complex networks.

The experiment predicts the network robustness through undirected and weighted graphs to verify the effectiveness of the MCNN-RP, as shown in Figure 7. The prediction results in Figure 7 showed that under random attacks, the CNN-RP algorithm based on RN, SFN, and QSN had prediction errors of 0.0338, 0.0632, and 0.0375 for the robustness of undirected complex networks, respectively. The prediction errors for the robustness of weighted complex networks were 0.0326, 0.0584, and 0.0419, respectively. The MCNN-RP for three types of network samples had prediction errors of 0.0301, 0.0484, and 0.0323 for the undirected complex network, and 0.0285, 0.0541, and 0.0314 for the weighted complex network, respectively. Compared with the CNN-RP algorithm, the MCNN-RP algorithm reduced the robustness prediction of undirected networks by 10.94%, 23.41%, and 13.86%, respectively, and reduced the robustness prediction of weighted networks by 0.0041, 0.0043, and 0.0105, respectively. The results



(a) The scale and average of the real network model (b) Error of CNN-RP and MCNN-RP algorithms

Figure 8: Robustness prediction errors in real networks (ML, GY, CE, and PB).

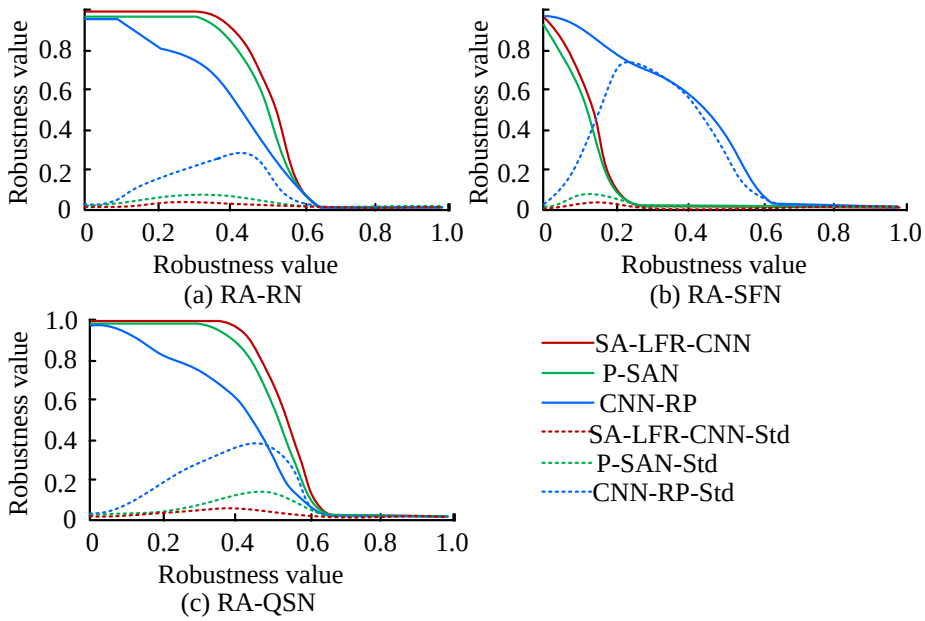


Figure 9: Connectivity robustness prediction in undirected graph networks: Comparison of CNN-RP, P-SAN, and SA-LFR-CNN.

have verified the effectiveness and feasibility of the MCNN-RP algorithm.

To verify the superiority of the MCNN-RP, it is compared with the CNN-RP algorithm in a network resource library. Real networks include Movie Lens-user (ML), Grid Yeast (GY), C-Elegance (CE), and Polbooks (PB). The results are displayed in Figure 8. In Figure 8 (a), the scales of ML, GY, CE, and PB real networks were 7041, 6002, 277, and 110, respectively, with average degrees of 7.20, 47.56, 7.88, and 8.45, respectively. In Figure 8 (b), the CNN-RP and MCNN-RP had prediction robustness errors of 0.1295 and 0.1284 for ML network, 0.1079 and 0.0556 for GY network, 0.2518 and 0.1667 for CE network, and 0.1707 and 0.1266 for PB network, respectively. Compared with the CNN-RP, the MCNN-RP reduced the robustness prediction error by 0.85%, 48.47%, 33.79%, and 25.83% in four real networks, respectively. The MCNN-RP has better robustness prediction performance for real networks.

3.2 Verification of network robustness prediction performance based on SA-LFR-CNN graph representation learning algorithm

The experiment compares the undirected graph connectivity robustness prediction results of CNN-RP, P-SAN, and SA-LFR-CNN algorithms, as shown in Figure 9. In Figure 9 (a), under random attacks, the prediction accuracy of CNN-RP, P-SAN, and SA-LFR-CNN algorithms for the robustness of random networks were 0.965, 0.974, and 0.988, respectively, with prediction error values of 0.1225, 0.0432, and 0.0320. The SA-LFR-CNN reduced prediction errors by 0.0793 and 0.0112 compared with the CNN-RP and P-SAN algorithms, respectively. In Figure 9 (b), the prediction accuracy of these three algorithms for the scale-free network was 0.984, 0.971, and 0.981, respectively, with prediction error values of 0.2566, 0.0723, and 0.0445. Compared with the other two algorithms, the SA-LFR-CNN algorithm reduced prediction errors by 0.1843 and 0.0278. In Figure 9 (c), the prediction accuracy of the three algorithms for the robustness of the Q-recovery network was 0.980, 0.984, and 0.992, respectively, with prediction error values of 0.1095, 0.0572, and 0.0441. Compared with the other two algorithms, the SA-LFR-CNN algorithm reduced prediction errors by 47.76% and

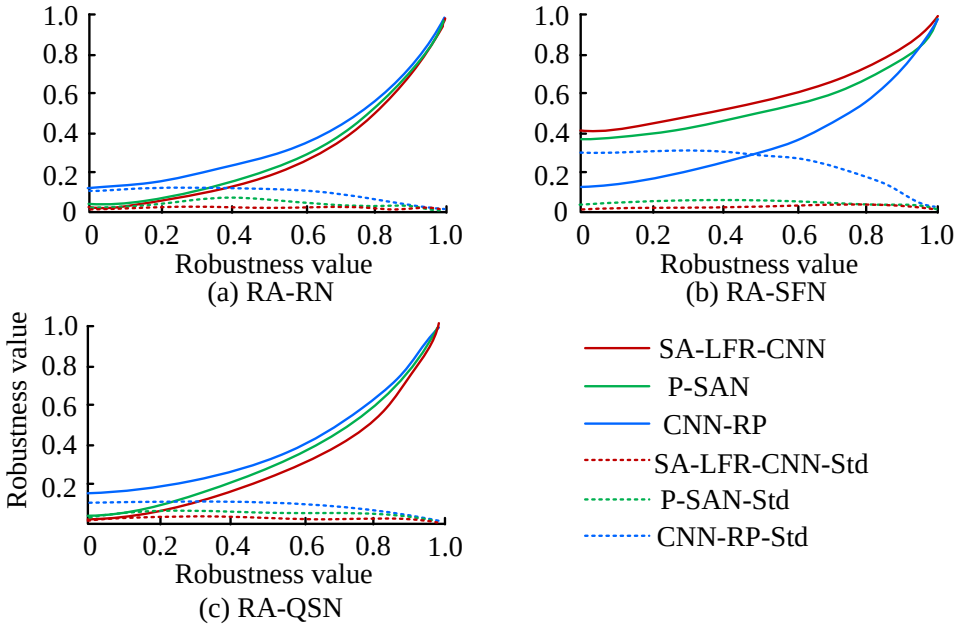


Figure 10: Controllability robustness prediction in directed graph networks: Comparison of CNN-RP, P-SAN, and SA-LFR-CNN.

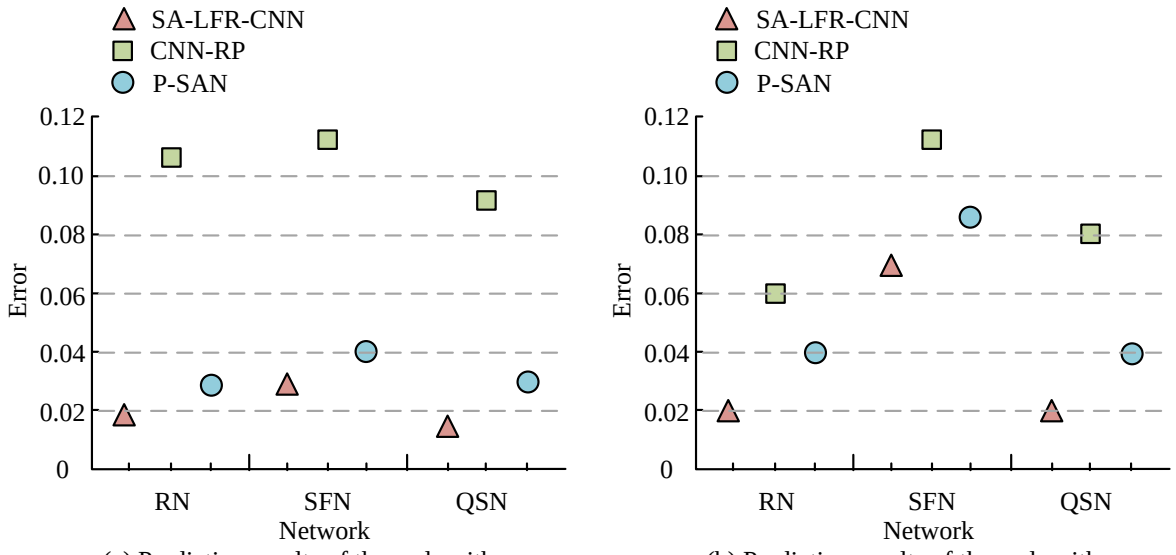
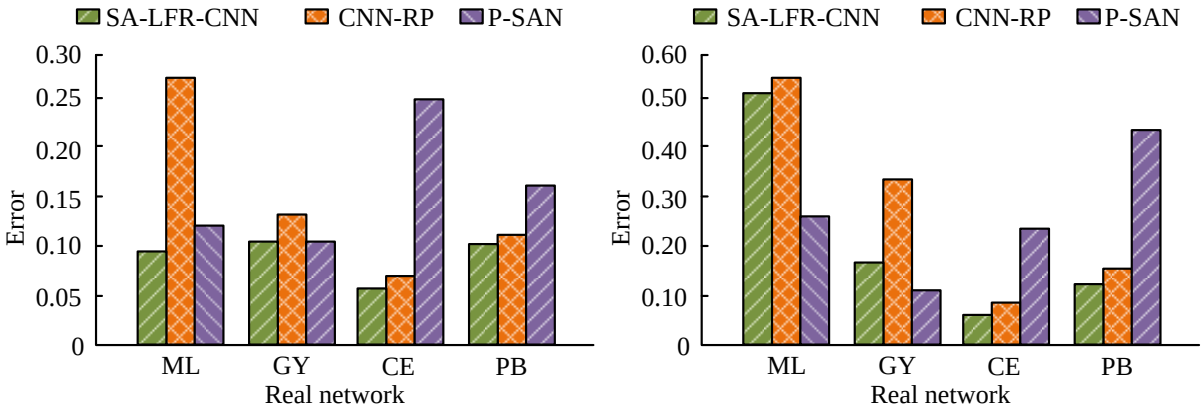


Figure 11: Robustness prediction under different network sizes: CNN-RP, P-SAN, and SA-LFR-CNN.

22.90%, respectively. The SA-LFR-CNN has high prediction accuracy, which is superior to other algorithms.

To further verify the predictive performance of the SA-LFR-CNN, the directed graph controllability robustness prediction results of CNN-RP, P-SAN, and SA-LFR-CNN algorithms are shown in Figure 10. In Figure 10 (a), under random attacks, the prediction error values of CNN-RP, P-SAN, and SA-LFR-CNN for the robustness of the random network were 0.0678, 0.0639, and 0.0300, respectively. The SA-LFR-CNN algorithm reduced prediction errors by 0.0378 and 0.0339, respectively, compared with the CNN-RP and P-SAN. Figure 10 (b) displayed that the prediction error values of CNN-RP, P-SAN, and SA-LFR-CNN algorithms for the robustness of scale-free networks were 0.1152, 0.0832, and 0.0825. The SA-LFR-CNN algorithm reduced prediction errors by 0.0327 and 0.0007. Figure 10 (c) displayed that the prediction error values of CNN-RP, P-SAN, and SA-LFR-CNN algorithms for the robustness of Q-recovery networks were 0.0703, 0.0643, and 0.0302. The SA-LFR-CNN reduced prediction errors by 57.04% and



(a) The connectivity robust prediction results of three algorithms

(b) The controllability robust prediction results of three algorithms

Figure 12: Robustness prediction errors in real networks (ML, GY, CE, and PB): CNN-RP, P-SAN, and SA-LFR-CNN.

Table 2: MAE, RMSE and confidence interval results of different algorithms.

Model	Dataset	MAE	RMSE	Confidence interval	Statistical significance
CNN	RN	0.1225	0.1532	[0.1150, 0.1300]	P<0.05
	SFN	0.2566	0.3054	[0.2450, 0.2682]	P <0.05
	QSN	0.1095	0.1321	[0.1000, 0.1190]	P <0.05
MCNN-RP	RN	0.0312	0.0391	[0.0285, 0.0339]	P <0.05
	SFN	0.0454	0.0567	[0.0412, 0.0496]	P <0.05
	QSN	0.0335	0.0421	[0.0301, 0.0369]	P <0.05
SA-LFR-CNN	RN	0.0320	0.0385	[0.0290, 0.0350]	P <0.05
	SFN	0.0445	0.0543	[0.0408, 0.0482]	P <0.05
	QSN	0.0441	0.0512	[0.0405, 0.0477]	P <0.05

Note: P<0.05 indicates that the statistical test results are significant.

53.03%, respectively. The SA-LFR-CNN has good robustness prediction performance, verifying the effectiveness.

Different network sizes can affect the robustness prediction results of the network. Therefore, the robustness prediction algorithms under different network scales are compared in the experiment, as displayed in Figure 11. In Figure 11 (a), when the network size was 800, the robustness prediction errors of CNN-RP, P-SAN, and SA-LFR-CNN algorithms under RN samples were 0.1124, 0.0320, and 0.0185, respectively. The prediction errors of the three algorithms under SFN network samples were 0.1145, 0.0402, and 0.0326, respectively. The prediction errors under QSN network samples were 0.0921, 0.0336, and 0.0128. Figure 11 (b) indicated that when the network size was 1,000, the CNN-RP algorithm predicted network robustness with error values of 0.0603, 0.1148, and 0.0813 for RN, SFN, and QSN network samples, respectively. The P-SAN algorithm predicted robustness with error of 0.0403, 0.0827, and 0.0396. The SA-LFR-CNN predicted network robustness with error of 0.0204, 0.0722, and 0.0195. The experimental results show that under different network sizes, the SA-LFR-CNN has high prediction accuracy.

To further verify the robustness and generalization ability of the SA-LFR-CNN algorithm, the robustness prediction results of four real networks are compared, ML, GY, CE, and PB, as shown in Figure 12. In Figure 12 (a), the CNN-RP algorithm had connectivity robustness prediction error values of 0.2745, 0.1334, 0.0715, and 0.1076 for ML, GY, CE, and PB networks, respectively. The P-SAN algorithm had prediction error values of 0.1296, 0.1108, 0.2504, and 0.1706 for the four real

networks, and the SA-LFR-CNN algorithm had prediction error values of 0.0906, 0.1106, 0.0715, and 0.1052 for the four networks, respectively. In Figure 12 (b), the CNN-RP algorithm had prediction errors of 0.5744, 0.3746, 0.0672, and 0.1895 for the controllability robustness of ML, GY, CE, and PB networks, respectively. The P-SAN had prediction errors of 0.2663, 0.1120, 0.2656, and 0.4328 for the four real networks, respectively. The SA-LFR-CNN algorithm had prediction errors of 0.5155, 0.1882, 0.0458, and 0.1456 for the four networks, respectively. The SA-LFR-CNN has a small prediction error value for network robustness and high prediction accuracy.

To further evaluate and compare the performance of different algorithms in predicting network robustness. The study compares the Mean Average Error (MAE), Root Mean Square Error (RMSE), and confidence intervals of different algorithms using the ANOVA method. The statistical significance of performance differences is verified. The results are shown in Table 2. The results showed that the MAE and RMSE of the proposed model were lower than those of the CNN model, indicating that the algorithm had better performance in predicting network robustness.

Table 3: Analysis of ablation experiment results.

Model	Dataset	Prediction error of connectivity robustness	Prediction error of controllability robustness
Baseline model	RN	0.1225	0.0678
	SFN	0.2566	0.1152
	QSN	0.1095	0.0703
Remove SAN operator	RN	0.0432	0.0378
	SFN	0.0723	0.0832
	QSN	0.0572	0.0643
Remove topological prior knowledge	RN	0.0332	0.0525
	SFN	0.0501	0.0963
	QSN	0.0387	0.0589
Complete model	RN	0.0320	0.0300
	SFN	0.0445	0.0825
	QSN	0.0352	0.0302

Further ablation experiments are conducted to better explain the performance of the model. The results are shown in Table 3. When applying SAN operator and topology prior knowledge, the robustness error of the model significantly increased.

4 Discussion and conclusion

This study proposed a robust prediction method for complex networks based on graph representation learning and improved CNN. To verify the effectiveness and superiority of the proposed method, the proposed method was comprehensively compared with existing CNN-RP and P-SAN baseline methods. Under random attacks, MCNN-RP reduced the prediction errors of undirected networks by 10.94%, 23.41%, and 13.86%, respectively, and reduced the prediction errors of weighted networks by 0.0041, 0.0043, and 0.0105, respectively. In addition, the SA-LFR-CNN reduced prediction errors by 0.1843 and 0.0278, as well as 47.76% and 22.90%, respectively, in the robustness prediction of scale-free networks and Q-recovery networks. In real network testing, the connectivity robustness prediction error values of SA-LFR-CNN algorithm were 0.0906, 0.1106, 0.0715, and 0.1052, respectively, and the controllability robustness prediction error values were 0.5155, 0.1882, 0.0458, and 0.1456, respectively. The results show that under random attacks, degree attacks, and betweenness attacks, the MCNN-RP algorithm has significantly lower robustness prediction errors for undirected and weighted networks than the CNN-RP algorithm. Compared with CNN-RP, MCNN-RP significantly improves the training and prediction speed of the model by introducing multiple convolutional neural network structures and prior knowledge of network topology. The SA-LFR-CNN algorithm further optimizes the computation process by improving the SAN operator and graph representation learning, making it more scalable when dealing with large-scale complex networks. The graph representation learning method can extract deep structural features of networks, thereby improving the ability to represent complex networks. Through graph representation learning, the model can better capture the topology and node relationships of the network, thereby improving prediction accuracy. In addition, the improved SAN operator can further optimize the computation process, making it more efficient and accurate in handling large-scale networks.

In summary, the MCNN-RP and SA-LFR-CNN algorithms proposed in this study demonstrate significant superiority in predicting the robustness of complex networks. By introducing graph representation learning and prior knowledge of network topology, these two algorithms outperform existing baseline methods on prediction accuracy, computational efficiency, and practical applications.

The limitations of this study are as follows. (1) Synthetic networks are created through specific generative models, which may not fully reflect the complexity and diversity of real networks. The generation model of this network is usually based on specific assumptions and parameter settings, which may result in the generated network deviating from the real network in certain characteristics. (2) The size and complexity of real network datasets may have an impact on model training and testing. The noise in the dataset may mask the true topology and dynamic

characteristics of the network, thereby affecting the generalization ability of the model. Therefore, future research can consider developing synthetic network generation models that are closer to the characteristics of real networks, which can better simulate the dynamic changes and local properties of real networks.

Funding

The research is supported by: Hubei Province Universities Outstanding Young and Middle-aged Scientific and Technological Innovation Team Project in 2022 (No. T2022035); The Natural Science Foundation of Hubei Province of China in 2022 (No. 2022CFB928); School Level Scientific Research Innovation Team of Wuhan College in 2020 (No. kyt202001); Hubei Yidan University Education Development Foundation, Research on Human-Machine Collaborative Governance Model for Short Video Content Ecosystem in the Digital Intelligence Era (No. JJA202510).

References

[1] Xin Zheng, and Xiaodong Zhang. Robustness of cloud manufacturing system based on complex network and multi-agent simulation. Entropy (basel), 25(1):45-62, 2022. https://doi.org/10.3390/e25010045

[2] Yang Lou, Yaodong He, Lin Wang, and Guanrong Chen. Predicting network controllability robustness: A convolutional neural network approach. IEEE Transactions on Cybernetics, 52(5):4052-4063, 2022. https://doi.org/10.1109/TCYB.2020.3013251

[3] Josef Baumgartner, Alexandra Schneider, Ulugbek Zhenis, Franz Jager, and Josef Winkler. Mastering neural network prediction for enhanced system reliability. Fusion of multidisciplinary research, 3(1):261-274, 2022.

[4] Yuxiao Yang, Shaoyu Qiao, Omid G. Sani, J. Isaac Sedillo, Breonna Ferrentino, Bijan Pesaran, and Maryam M. Shanechi. Modelling and prediction of the dynamic responses of large-scale brain networks during direct electrical stimulation. Nature Biomedical Engineering, 5(4):324-345, 2021. https://doi.org/10.1038/s41551-020-00666-w

[5] Anastasios Valkanis, Georgios Papadimitriou, Georgia Beletsioti, Emmanouel Varvarigos, and Petros Nicopolitidis. Efficiency and fairness improvement for elastic optical networks using reinforcement learning-based traffic prediction. Journal of Optical Communications and Networking, 14(3):25-42, 2021. https://doi.org/10.1364/JOCN.440590

[6] Zewei Ji, Runhan Shi, Jiarui Lu, Fang Li, and Yang Yang. ReLMole: Molecular representation learning based on two-level graph similarities. Journal of Chemical Information and Modeling, 62(22):5361-5372, 2022. https://doi.org/10.1021/acs.jcim.2c00798

[7] Xiaocheng Yang, Mingyu Yan, Shirui Pan, Xiaochun Ye, and Dongrui Fan. Simple and efficient heterogeneous graph neural network. In Proceedings of the AAAI Conference on Artificial Intelligence. 37(9):10816-10824, 2023. https://doi.org/10.1609/aaai.v37i9.26283

[8] Edith Heiter, Bo Kang, Tijl De Bie, and Jefrey Lijffijt. Evaluating representation learning and graph layout methods for visualization. IEEE Computer Graphics and Applications, 42(3):19-28, 2022. https://doi.org/10.1109/MCG.2022.3160104

[9] Liang Peng, Yujie Mo, Jie Xu, Jialie Shen, Xiaoshuang Shi, and Xiaoxiao Li. GRLC: Graph representation learning with constraints. IEEE Transactions on Neural Networks and Learning Systems, 35(6):8609-8622, 2024. https://doi.org/10.1109/TNNLS.2022.3230979

[10] Yujie Mo, Liang Peng, Jie Xu, Xiaoshuang Shi, and Xiaofeng Zhu. Simple unsupervised graph representation learning. Proceedings of the AAAI Conference on Artificial Intelligence, 36(7):7797-7805, 2022. https://doi.org/10.1609/aaai.v36i7.20748

[11] Kavita Bhosle, and Vijaya Musande. Evaluation of deep learning CNN model for recognition of devanagari digit. Artificial Intelligence and Applications, 1(2):114-118, 2023. https://doi.org/10.47852/bonviewAIA3202441

[12] Tianjian Dong, Qi Qi, Jingyu Wang, Alex X. Liu, Haifeng Sun, and Zirui Zhuang. Generative adversarial network-based transfer reinforcement learning for routing with prior knowledge. IEEE Transactions on Network and Service Management, 18(2):1673-1689, 2021. https://doi.org/10.1109/TNSM.2021.3077249

[13] Jiajun Chen, Huarui He, Feng Wu, and Jie Wang. Topology-aware correlations between relations for inductive link prediction in knowledge graphs. Proceedings of the AAAI Conference on Artificial Intelligence, 35(7):6271-6278, 2021. https://doi.org/10.1609/aaai.v35i7.16779

[14] Laura von Rueden, Sebastian Mayer, Katharina Beckh, Bogdan Georgiev, Sven Giesselbach, Raoul Heese, Birgit Kirsch, Julius Pfrommer, Annika Pick, Rajkumar Ramamurthy, Michal Walczak, Jochen Garcke, Christian Bauckhage, and Jannis Schuecker. Informed machine learning–a taxonomy and survey of integrating prior knowledge into learning systems. IEEE Transactions on Knowledge and Data Engineering, 35(1):614-633, 2021. https://doi.org/10.1109/TKDE.2021.3079836

[15] Xiao Li, Ziqi Wang, Bo Zhang, Fuchun Sun, and Xiaolin Hu. Recognizing object by components with human prior knowledge enhances adversarial robustness of deep neural networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(7):8861-8873, 2023. https://doi.org/10.1109/TPAMI.2023.3237935

[16] Entesar Abdulsaed, Maytham Alabbas, and Raidah Khudeyer. Hyperparameter optimization for convolutional neural networks using the salp swarm algorithm. Informatica, 47(9):133-144, 2023. https://doi.org/10.31449/inf.v47i9.5148

[17] Sebastian Ehlert, Marcel Stahn, Sebastian Spicher, and Stefan Grimme. Robust and efficient implicit solvation model for fast semiempirical methods. Journal of Chemical Theory and Computation, 17(7):4250-4261, 2021. https://pubs.acs.org/doi/10.1021/acs.jctc.1c00471

[18] Yang Lou, Yaodong He, Lin Wang, Kim Fung Tsang, and Guanrong Chen. Knowledge-based prediction of network controllability robustness. IEEE Transactions on Neural Networks and Learning Systems, 33(10):5739-5750, 2021. https://doi.org/10.1109/TNNLS.2021.3071367

[19] Yang Lou, Ruizi Wu, Junli Li, Lin Wang, Xiang Li, and Guanrong Chen. A learning convolutional neural network approach for network robustness prediction. IEEE Transactions on Cybernetics, 53(7):4531-4544, 2022. https://doi.org/10.1109/TCYB.2022.3207878

[20] Yangyan Liu, and Bolin Pan. Profit estimation model and financial risk prediction combining multi-scale convolutional feature extractor and BGRU model. Informatica, 48(11):15-32, 2024. https://doi.org/10.31449/inf.v48i11.5941

[21] Zhiwen Chen, Jiamin Xu, Tao Peng, and Chunhua Yang. Graph convolutional network-based method for fault diagnosis using a hybrid of measurement and prior knowledge. IEEE Transactions on Cybernetics, 52(9):9157-9169, 2021. https://doi.org/10.1109/TCYB.2021.3059002

[22] Emrah Irmak. Multi-classification of brain tumor MRI images using deep convolutional neural network with fully optimized framework. Iranian Journal of Science and Technology, Transactions of Electrical Engineering, 45(3):1015-1036, 2021. https://doi.org/10.1007/s40998-021-00426-9

[23] Xiaocheng Yang, Mingyu Yan, Shirui Pan, Xiaochun Ye, and Dongrui Fan. Simple and efficient heterogeneous graph neural network. Proceedings of the AAAI Conference on Artificial Intelligence, 37(9):10816-10824, 2023. https://doi.org/10.1609/aaai.v37i9.26283

[24] Martin Genzel, Jan Macdonald, and Maximilian März. Solving inverse problems with deep neural networks–robustness included? IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(1):1119-1134, 2022. https://doi.org/10.1109/TPAMI.2022.3148324

[25] Harry Rogers, Beatriz De La Iglesia, Tahmina Zebin, Grzegorz Cielniak, and Ben Magri. Advancing precision agriculture: Domain-specific augmentations and robustness testing for convolutional neural networks in precision spraying evaluation. Neural Computing and Applications, 36(32), 20211-20229, 2024. https://doi.org/10.1007/s00521-024-10142-0