A Multi-Convolutional Neural Network Framework For Network Connectivity Robustness Prediction

Ruizi Wu, Junli Li, Zhuoran Yu and Sheng Li

2021 China Automation Congress (CAC) | 978-1-6654-2647-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/CAC53003.2021.9727711

Abstract—**Connectivity robustness reflects the ability of a complex network maintaining its basic connected structures and functions under various external attacks. Conventionally, connectivity robustness is evaluated by simulating attacks, which may take expensive computational resources, especially for large-scale networked systems. Past studies have shown that convolution neural networks (CNN) have excellent performance in both regression and classification. Complex networks are transformed into grey images and processed by CNNs. In this paper, a multi-CNN framework is come out to predict the connectivity robustness, where each CNN performs either network classification or robustness prediction. Experimental studies are carried on four network topologies under random and targeted attacks. The simulations results shows that the proposed multi-CNN approach can precisely predict the connectivity robustness of complex networks. Further experiments also shows that multi-CNN performs better than single CNN method.**

Key words——Complex Network, Convolution Neural Network, Connectivity Robustness

I. Introduction

In real world, most complex systems can be described by various types of networks. In addition to widely used in mathematics, engineering, economics and other disciplines, complex networks are more closely related to our daily life, such as in the dissemination of information, the evolution of language, the spread and blocking of epidemics, etc., complex networks provide a very valuable reference model. Since 1999, complex network science has developed rapidly. It involves the knowledge and basic theory of many subjects because of its intersecting and complexity. It has attracted wide attention in statistical physics, computer technology, systems control engineering, economics, applied mathematics, biology and sociology, etc.[1][2][3]

The complex networks are made of many nodes and the links between nodes. The nodes represent the instances in the system, and the links between nodes represent the relationships between node entities. The complexity of nodes and connections in a network is essential to the study of the basic functions of the network (controllability and synchronization). However, in the real world, complex networks are faced with various malicious attacks from the outside of the system, which has a bad influence on the connectivity of complex networks and causes the loss of structure and function of the network. At present, how to improve the network connectivity, and then improve the network resistance to such attacks and catastrophic failures has become the focus of current research[4][4][5][6][7][8]. Removing nodes or links in the network is the main way of malicious attacks and failures, which will cause badly negative effects on the network, or even a complete collapse to the network. The capability of a network to keep its connectivity in the face of malicious attacks and failures is called "connectivity robustness".

Attacks to a network can be divided into random attacks and centrality-targeted attacks. Random attack follows the principle of “randomness”, which randomly chooses a target with a certain probability, and then attacks the network. Targeted attack aims at finding nodes or edges that have the highest centrality. The definition of centrality includes degree, intermediate, adjacency matrix eigenvalue, and so on. In addition, common methods include neighborhood similarity[10], structural hole method[11], etc. In the attack of network, the size of giant connected component is often used to evaluate the attack effect. Many studies which uses this variable to calculate connectivity has found that heterogeneous networks with onion topology have the best robustness in experimental networks [6][12][13][14].

In recent years, the development of deep neural networks has provided a new solution to the problem of unstructured data regression and classification. Compared with traditional neural network, which requires artificial definition of features, deep neural network can completely ignore these requirements and complete data feature extraction driven by data, eliminating the risk of errors caused by human bias. Moreover, deep neural network can complete the regression and classification of complex input data in a very short time. Convolutional Neural Network (CNN), as a classical deep neural network[15], performances a great effect that is difficult to be achieved by traditional machine learning in the direction of image processing through three strategies of local receptive field, weight sharing and down-sampling. Convolutional neural network has shown excellent performance in network controllability robustness prediction [16][17][18] and key node identification [19].

It is vital to evaluate the connectivity robustness of the network for optimizing the robustness and preventing the fault defects of the complex system in reality. However, the calculation of the connectivity robustness needs to be obtained in the process of simulation attack. The simulation attack process includes the iterative processes of centrality calculation, attacking nodes determination and nodes (or edges) removal, which requires a huge amount of computation in a large-scale network system, so that it becomes infeasible in a large number of real networks.

Network has the same matrix storage mode as grayscale image being processed by computer, as well as the lack of manual intervention and completely data driven of deep

Ruizi Wu, Junli Li, Zhuoran Yu and Sheng Li are with the College of Computer Science, Sichuan Normal University, Chengdu, China.
(e-mail: vridge@foxmail.com;li.junli@vip.163.com).

This research was supported in part by the National Natural Science Foundation of China (No. 62002249) and in part by the Open Project Program of the State Key Lab of CAD&CG (A2112), Zhejiang University.

(Corresponding author: Junli Li)

learning make it possible of using CNN to predict connectivity robustness. In the experiment, it is found that the CNN predictor trained by the synthetic graph has a good effect on predicting network's connectivity, of which has the same type as the training network, but it lacks generalization on different types of network. To solve this problem, a combined multi-CNN is proposed in this paper, which makes full use of the sensitivity of CNN in classification, and makes up for the generalization ability of different types of CNN through multiple predictors. The proposed multi-CNN consists of a classifier and several predictors. The classifier is essentially a CNN module that achieves the network classification task, and the predictor part is a combination module of multiple CNNs that achieves the regression task. In addition, the predictor also contains a filter, which uses the prior knowledge of the network attacking process, such as the curve trend, upper and lower bounds, etc., to limit the output of the predictor. The proposed multi-CNN classifier and the CNN structure of the predictor are same as those used in the controllability robustness predictor [16][17][18] in this paper, but they have different goals and functions. Experimental results show that the multi-CNN can predict the connectivity robustness quite accurately, and has nice generalization ability for deleting sequential nodes in both undirected or directed graphs and weighted or unweighted graphs.

The following parts of this paper are as follows: the second part introduces the definition and physical meaning of network connectivity robustness; the third part introduces the structure of the combined multi-CNN predictor. The fourth part includes the setting of the experiments and the specific results and analysis; The fifth part contains the introduction and description of the conclusion.

II. Robustness of network

The connectivity of a complex network is often defined by the size of the network's LCC(largest connected component), while the connectivity robustness refers to the network's ability to maintain its connectivity during an attack. In the experiment, the connectivity robustness is defined as, given an attack sequence, the size of network giant connected component changes during the attack process. That is, connectivity robustness is represented as a vector of the same length as the number of attack times.

In this paper, the connectivity of complex networks is defined by the size of LCC, of which the standardized representation is as follows:

$$s(i) = \frac{N_{LCC}(i)}{N-i},\ \ i = 0,1,\dots,N-1， \tag{1}$$

Where, i represents the number of attacks on a network, N represents the initial number of vertexes in the network, and $N_{LCC}(i)$ represents the size of LCC in the network after undertaking i times attacking to the network. $s(i)$ represents the connectivity robustness of the network after removing i nodes. For an network with N nodes, if the attack process lasts $N-1$ times, the calculated connectivity robustness result is a vector with the length of $N-1$.

In order to facilitate the comparison of the connectivity robustness of different networks, it is necessary to scalarize the connectivity robustness, which is commonly used by summing or averaging. In this paper, the definition of connectivity robustness adopts the standard connectivity robustness calculation method presented by Christian M in his research [6]:

$$\bar{s} = \frac{1}{N}\sum_{i=0}^{N-1} s(i). \tag{2}$$

Given two vectors of connectivity robustness,s_1 and s_2, the error of them can be calculated as follows:

$$\sigma = |s_1 - s_2| \tag{3}$$

As for the average error, that is to scalarize the error by Equ.4:

$$\bar{\sigma} = \frac{1}{N}\sum_{i=0}^{N-1} \sigma(i) \tag{4}$$

The vector σ could easily evaluate the destructive ability of different methods to networks. The scalar $\bar{\sigma}$ can easily compare the performance of multiple networks under the same attack mode.

III. CNN STRUCTURE

Some research proves that deep learning has excellent performance in predicting controllability robustness[16][18]. In this paper, we combined many predictors to predict the connectivity robustness. Compared with the controllability robustness, range of values is wider and the degree of variation is larger. So we add a filter based on the prior knowledge after predictor to eliminate the illegal trend of the output.

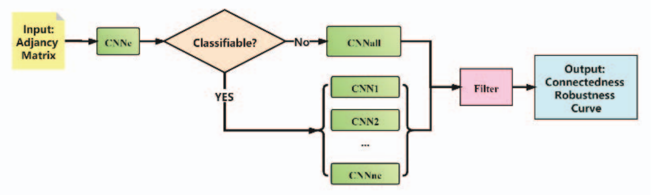


Fig.1: The framework of Multi-CNN Predictor, which includes serval CNN. CNN_c is used for network classification and CNN_i , $i = 1,\dots,nc$ is used for predicting where nc is number of networks classes. The CNN_c classify networks into classes, and later specific CNN_i will be applied to predict the connectivity robustness.

The basic structure of the combined multi-CNN proposed in this paper is shown in Fig.1. The input data is the adjacency matrix of the network. If the network can be classified by CNN_c, the network will be predicted using the predictor of the corresponding category. Otherwise, use CNN_{all} for predicting. The CNN structure contains multiple convolutional layers, which are processed by linear rectification unit Relu. After each convolutional layer is a maximum pooling layer ($f(x) = \max\{0, x\}$). The convolution part is followed by two fully connected layers, which are used to process the features and reshape the output into $N-1$.

TABLE.I Parameter details of CNN

Group	Layer	Kernel size	Output Channel
Group 1	Conv2D-64	7	64
	MaxPooling	2	64
Group 2	Conv2D-64	5	64
	MaxPooling	2	64
Group 3	Conv2D-128	3	128
	MaxPooling	2	128
Group 4	Conv2D-128	3	128
	MaxPooling	2	128

Group 5	Conv2D-256	3	256
	MaxPooling	2	256
Group 6	Conv2D-256	3	256
	MaxPooling	2	256
Group 7	Conv2D-512	3	512
	MaxPooling	2	512

The CNN structure of the combined multi-CNN is shown in Fig.2, and the specific parameter settings are shown in Table 1. The structure of CNN in this paper is borrowed from the design of VGG[20], and its structure and scale are simplified. Seven convolution blocks are retained, each convolution block is extracted by a convolution layer, and is treated by Relu function as the excitation function, and then is processed by maximum pooling layer to reduce the matrix size.

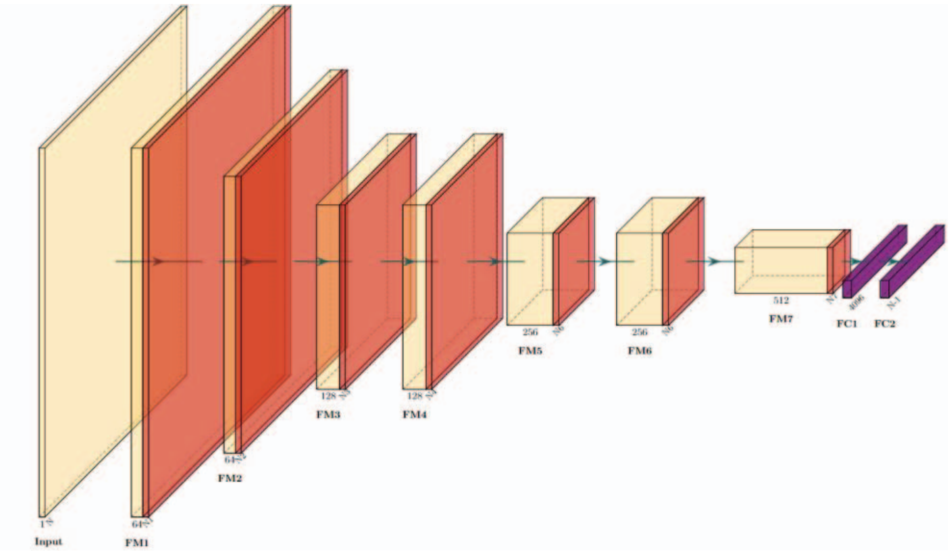


Fig.2 The architecture of the CNN. FM is the feature map outputted by convolutional layers. FC represents fully connected layers. Every FM includes a convolution layer and a pooling layer. The data size $N_i = \lceil N/2^{(i+1)} \rceil, i = 1,2,\ldots,7$. The concatenation layer flattens the matrix to a vector, from FM 7 to FC 1, i.e., $N_{FC1} = N_7 \times N_7 \times 512$. N_{FC2} is a hyperparameter and $N_{FC2} \in (N_{FC1}, N-1)$. Always set the size of $N_{FC1} = 4096$. Set the networks sizes $N = 1000$.

In the regression problem, the square error function is the commonly used loss function. In this paper, the connectivity robustness curve of the network under simulation attack is adopted as the true label, and the loss function is defined as the square error between the connectivity robustness under simulation and the predicted value:

$$Loss = \frac{1}{N}\sum_{i=0}^{N-1} ||\hat{s}(i) - s(i)|| \tag{5}$$

$\hat{s}(i)$ represents the i-th value of the predicted connectivity robustness curve, and $s(i)$ is the connectivity robustness value after the i-th attack in the simulation attack. $||\cdot||$ operator on behalf of the calculation of Euclidean distance. In the experimental part, the training process parameters of CNN are updated by Equ.5.

In the process of network attacking, the size of the giant connected component is constantly decreasing. Then the unstandardized predicted result of the connectivity robustness must be a non-increasing curve. However, in the experiment, the curve predicted by CNN will show a local increase. And the connectivity robustness curve must have upper bound and lower bound. After each attack, the upper bound of the current connectivity robustness is the number of nodes in the current network (representing the connected state of the network), and the lower bound is 1(indicating that all nodes in the network have no connections and are all isolated nodes). However, due to the data-driven feature of CNN, its output may obviously contrary to logic. Therefore, a filter based on prior knowledge is added in this paper. The filter mainly performs two functions: (1) limit the upper and lower bounds of the curve value; (2) correct the local increasing trends.

The filter uses prior knowledge to limit the upper and lower bounds of the curve in the first part, just as follows:

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

In the second part, the filter processes these increasing trend with linear interpolation. If there is a local increasing segment in the predicted result, the two points on both sides of the increasing segment are used as reference points to interpolate the increasing segment between the two reference points. Assume that $N_{LCC}(k) > N_{LCC}(i), (k \geq i+1)$ in the predicted result, that is, the result begins to increase after the i-th attack. The coordinate of $(i, N_{LCC}(i))$ is taken as the first reference point, and then traverse the curve, that is, $j = k+1, k+2, \ldots,$ until $N_{LCC}(i) \geq N_{LCC}(j)$. Here, $(j, N_{LCC}(j))$ is taken as the second reference point. The filter interpolates the curve segment $(N_{LCC}(i), N_{LCC}(i+1), \ldots, N_{LCC}(j-1))$ as follows:

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

The filter corrects the predicting results by adjusting the upper and lower bounds of illegal data and interpolating the increasing data segments. In additional, the two steps have a strictly logical sequence. To ensure that the data are distributed within the scope of prior knowledge, the upper and lower bounds of illegal data must be adjusted first, and then restrict the increasing segment.

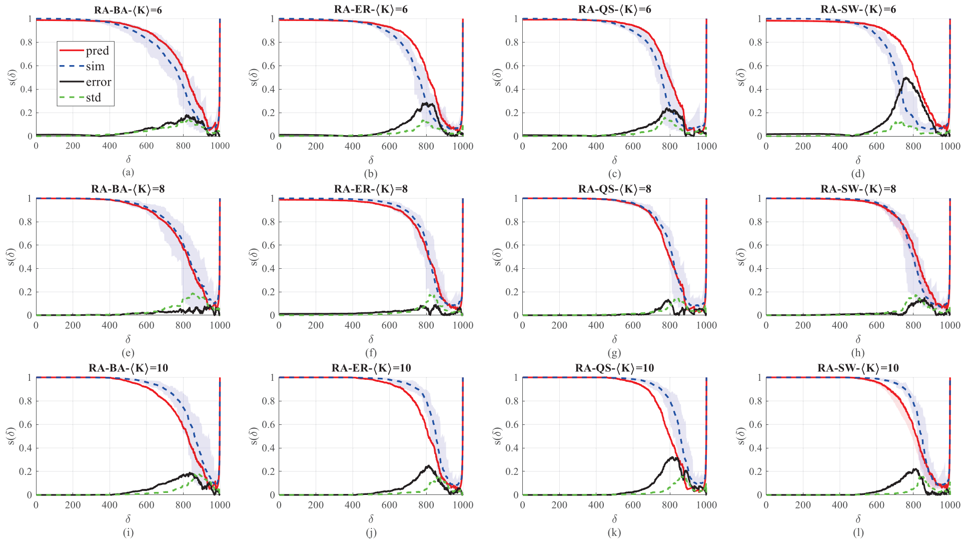


Fig. 3: Results of multi-CNN for testing networks under RA. δ represents the times of removed nodes; s(δ) represents the fraction of LCC versus the current number of network nodes, as shown in Eq. (1). The twelve subgraphs respectively represent the predicting results of the four types of networks (BA, ER, QS, SW) with degree distributions of 6, 8, and 10.

IV. Experimental studies

In order to verify the performance of the combined multi-CNN, we designed the following experiments.

In the experiment of this paper, directed unweighted synthetic graphs are used: Erdös-Rényi(ER)[21], scale-free(SF),q-snapback(QS) and Newman-Watts(SW).

In the experiment, our combined multi-CNN adopts the above four network types with 1000 nodes, and conducts training under three average degree distribution of 6, 8, and 10 respectively. CNN_C classifier and $CNN_i, i = 1,2\ldots nc$ predictor is trained separately. The input data of classifier CNN_C is the synthetic graph's adjacency matrix, labeled as the type of the network. The sample input data of the predictor CNN_i is the adjacency matrix of synthetic graph, and the label is the real value of the connectivity robustness calculated by simulation attack on the network. In the process of classifier training and testing, the total number of networks in dataset is 4*3*450=5400(4 topologies and 3 degree distributions), of which 1000 for testing and 4400 for training. As for predictors, there are also 5400 samples in the predictor training process. A total of 4x3x20=240 synthetic networks with the same degree distribution and topologies as the training set were selected for the test data.

In the classification process, SoftMax function is used to process and classify the network features. At this time, the threshold value is set at 0.8. If the classification accuracy probability is greater than or equal to 0.8, the network will be classified into the corresponding category and the corresponding CNN predictor will be used for prediction. Otherwise, the network is classified into categories that cannot be clearly classified, and CNN_{all} is used for prediction.

In this paper, we have completed two parts of experiments, which respectively predict RA(Random Attack) and TD(Targeted Attack) on the directed unweighted networks by combined multi-CNN.

A. Network classification

The multi-CNN in this paper includes two parts: network classification and prediction. The classifier mainly classifies the input network to determine the subsequent prediction using the predictor of the corresponding type. In the experiment of RA and TD, the same CNN model is used in the classification part. The confusion matrix of network classification results is shown in Table. II. In the experiment where the classification threshold is set to 0.8, the classification effect performs well, and the topology of the network can be effectively classified.

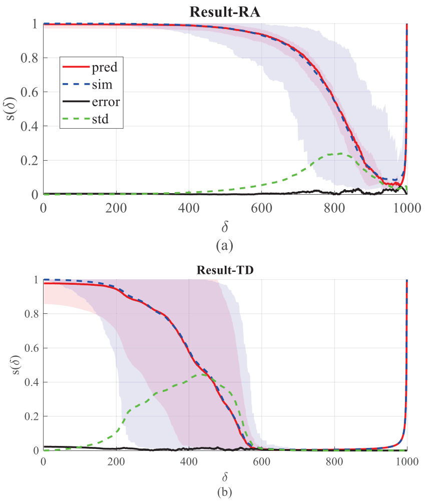
TABLE.II Confusion matrix of the classifier for classifying directed unweighted networks. Other means the input is non-classifiable; (pred) represents the predicted type and (real) represents the actual type of the network.

	BA(pred)	ER(pred)	QS(pred)	SW(pred)	Other
BA(real)	**60**	0	0	0	0
ER(real)	0	**60**	0	0	0
QS(real)	0	0	**60**	0	0
SW(real)	0	0	0	**60**	0
Other	0	0	0	0	0

B. Attack Prediction

RA means random attacks. This method removes nodes in the network which are randomly selected with a certain probability. It is often used to simulate the random failure in the real system. TD attack, which define the node centrality as node degree and then remove those nodes which has the best centrality. The attack method calculates the degree of all nodes before each attack, and selects the node with the largest degree to attack. The prediction results of the two attack methods mentioned above are shown in Fig.3.

The data in Fig.3 and Fig.4 is the average result of all the samples. The red curve represents the prediction results of the predictor for all the sample networks, the blue dot line represents the results of simulation attacks on the network, the black curve represents the absolute error between the predicted value and the real value of the simulation attack, and the green dot line represents the standard deviation of the simulation attack results.



(a)

(b)

Fig.4 Results of multi-CNN for testing data under RA and TD respectively. The meaning of δ and s(δ) are just the same as Fig.3.

The prediction results of different types of networks with different degree distributions are shown in Fig.4 below.

In Fig.3 and Fig.4, the predicted value can fit the trend and details of the real value very well. In terms of the results of all samples, the error between the prediction results and the simulation results can be stably less than the standard deviation of the true value, indicating that the prediction effect has a very excellent accuracy. As is shown in Fig.4, the predicted results of all groups well regressed the trend of real values.

C. Comparison

The experiments above have revealed the accuracy of the proposed method. Our recent work has proved that single CNN performs well predicting connectivity robustness.[22]

In this experiment, we trained a single CNN with the same training set as the previous experiment. And then we apply the two approaches on the same test data which has 240 synthetic networks.

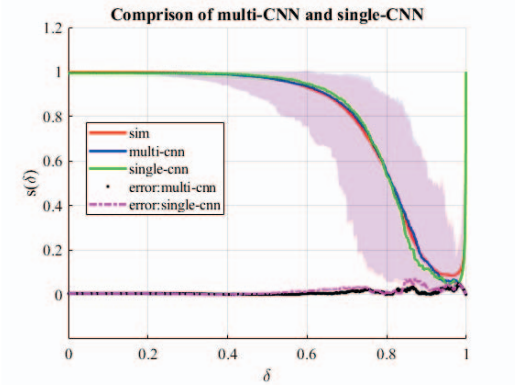


Fig.5 Results of comparison of multi-CNN and single-CNN. The meaning of δ and s(δ) are just the same as Fig.3.

Fig.5 showed the comparison of multi-CNN and single-CNN under the random attack scenario. The red, green and blue curves represent the true value (obtain by attacking simulation), results of single-CNN and results of multi-CNN, respectively. The black and pink curves reveal the error between the two methods and the true value.

Generally speaking, the two methods can fit the trend well. In the beginning, the two curves are within the same margin of error. But in the middle and later part of the curves, multi-CNN obviously has lower error than single CNN does. The results indicates that multi-CNN could deal with topology features better, which has lower error level then the existing method.

Conclusion

In this paper, we propose a multi-CNN framework to predict the connectivity robustness of complex networks under destructive attacks.

Compared with traditional simulation method, the proposed method performs faster and more effectively. The experimental results demonstrate that our multi-CNN based prediction can accurately regress the trends of the connectivity robustness of the network under random attack and degree attack scenarios.

References

[1].Barabási A.L., Network Science. Cambridge University Press, 2016.

[2].Newman M. E., Networks: An Introduction. Oxford University Press,2010.

[3].G. Chen and Y. Lou, Naming Game: Models, Simulations and Analysis. Springer, 2019.

[4].P. Holme, B. J. Kim, C. N. Yoon, and S. K. Han, "Attack vulnerability of complex networks" *Physical Review E*, vol. 65, no.5, p. 056109,2002.

[5].B. Shargel, H. Sayama, I. R. Epstein, and Y. Bar-Yam, "Optimization of robustness and connectivity in complex networks" *Physical Review Letters*, vol. 90, no. 6, p. 068701, 2003.

[6].C. M. Schneider, A. A. Moreira, J. S. Andrade, S. Havlin, and H. J. Herrmann, "Mitigation of malicious attacks on networks," *Proceedings of the National Academy of Sciences*, vol. 108, no. 10, pp. 3838–3841,2011.

[7].Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási. "Control centrality and hierarchical structure in complex networks," *PLOS ONE*, vol. 7, no. 9, p. e44459, 2012.

[8].Bashan, Y. Berezin, S. Buldyrev, and S. Havlin, "The extreme vulnerability of interdependent spatially embedded networks," *Nature Physics*, vol. 9, pp. 667–672, 2013.

[9].Y.-D. Xiao, S.-Y. Lao, L.-L. Hou, and L. Bai, "Optimization of robustness of network controllability against malicious attacks," *Chinese Physics* B, vol. 23, no. 11, p. 118902, 2014.

[10].Y.-R. Ruan, S.-Y. Lao, J.-D. Wang, L. Bai, and L.-D. Chen, "Node importance measurement based on neighborhood similarity in complex network," *Acta Physica Sinica*, vol. 66, no. 3, p. 038902, 2017.

[11].H. Yang and S. An, "Critical nodes identification in complex networks," *Symmetry*, vol. 12, no. 1, p. 123, 2020.

[12].Z.-X. Wu and P. Holme, "Onion structure and network robustness," *Physical Review E*, vol. 84, no. 2, p. 026106, 2011.

[13].T. Tanizawa, S. Havlin, and H. E. Stanley, "Robustness of onion like correlated networks against targeted attacks," *Physical Review E*, vol. 85,no. 4, p. 046109, 2012.

[14].Y. Hayashi and N. Uchiyama, "Onion-like networks are both robust and resilient," *Scientific Reports*, vol. 8, 2018.

[15].J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[16].Y. Lou, Y. He, L. Wang, and G. Chen, "Predicting network controllability robustness: A convolutional neural network approach,"

IEEE Transactions on Cybernetics, 2020, doi:10.1109/TCYB.2020.3013251.

[17].A. Dhiman, P. Sun, and R. Kooij, “Using machine learning to quantify the robustness of network controllability,” *International Conference on Machine Learning for Networking (MLN2020)*. Springer International Publishing, 2021, pp. 19–39.

[18].Y. Lou, Y. He, L. Wang, K. F. Tsang, and G. Chen, “Knowledge-based prediction of network controllability robustness” *IEEE Transactions on Neural Networks and Learning Systems*, 2021, doi:10.1109/TNNLS.2021.3071367.

[19].C Fan, Zeng L, Sun Y, et al. Finding key players in complex networks through deep reinforcement learning[J]. Nature machine intelligence, 2020, 2(6): 317-324.

[20].K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv Preprint: 1409.1556, 2014.

[21].P. ErdÖs and A. Renyi, “On the strength of connectivity of a random graph,” Acta Mathematica Hungarica, vol. 12, no. 1–2, pp. 261–267,1964.

[22].Y. Lou, R. Wu, J. Li, L. Wang and G. Chen, "A Convolutional Neural Network Approach to Predicting Network Connectivity Robustness," in IEEE Transactions on Network Science and Engineering, doi: 10.1109/TNSE.2021.3107186.