Neural Networks 157 (2023) 136–146

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Classification-based prediction of network connectivity robustness

Yang Lou [a,b,*], Ruizi Wu [a], Junli Li [a], Lin Wang [c,b], Chang-Bing Tang [d], Guanrong Chen [e]

[a] College of Computer Science, Sichuan Normal University, Chengdu, 610066, China
[b] Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai, 200240, China
[c] Department of Automation, Shanghai Jiao Tong University, Shanghai, 200240, China
[d] Department of Electronics and Information Engineering, Zhejiang Normal University, Jinhua, 321004, China
[e] Department of Electrical Engineering, City University of Hong Kong, Hong Kong, China

ARTICLE INFO

Article history:
Received 26 February 2022
Received in revised form 29 August 2022
Accepted 12 October 2022
Available online 20 October 2022

Keywords:
Complex network
Connectivity
Robustness
Convolutional neural network
Prediction

ABSTRACT

Today, there is an increasing concern about malicious attacks on various networks in society and industry, against which the network robustness is critical. Network connectivity robustness, in particular, is of fundamental importance, which is generally measured by a sequence of calculated values that indicate the connectedness of the remaining network after a sequence of attacks by means of node- or edge-removal. It is computationally time-consuming, however, to measure and evaluate the network connectivity robustness using the conventional attack simulations, especially for large-scale networked systems. In the present paper, an efficient robustness predictor based on multiple convolutional neural networks (mCNN-RP) is proposed for predicting the network connectivity robustness, which is an natural extension of the single CNN-based predictor. In mCNN-RP, one CNN works as the classifier, while each of the rest CNNs works as an estimator for predicting the connectivity robustness of every classified network category. The network categories are classified according to the available prior knowledge. A data-based filter is installed for predictive data refinement. Extensive experimental studies on both synthetic and real-world networks, including directed and undirected as well as weighted and unweighted topologies, verify the effectiveness of mCNN-RP. The results demonstrate that the average prediction error is lower than the standard deviation of the tested data, which outperforms the single CNN-based framework. The runtime in assessing network connectivity robustness is significantly reduced by using the CNN-based technique. The proposed mCNN-RP not only can accurately predict the connectivity robustness of various complex networks, but also provides an excellent indicator for the connectivity robustness, better than other existing prediction measures.

1. Introduction

Many natural and engineering systems can be modeled as complex networks. The study of complex networks attracts increasing interest and attention from research communities in various scientific and engineering fields, including computer science, statistical physics, systems engineering, applied mathematics, biological sciences, and social sciences (Barabási, 2016; Chen & Lou, 2019; Chen, Wang, & Li, 2014; Newman, 2010).

Connectivity is fundamentally important for networks to function. An undirected network is connected if there is a path between every pair of nodes. While for a directed network, it is *weakly connected* if it remains to be connected after all the directions are removed, and it is *strongly connected* if there is a directed path from any node to any other node in the network.

Nowadays, malicious attacks as well as random failures literally exist in many engineering and technological applications, which degrade or even destroy certain network functions typically through destructing the network connectivity. Therefore, it has become necessary to strengthen the network connectivity against such attacks and failures (Bashan, Berezin, Buldyrev, & Havlin, 2013; Holme, Kim, Yoon, & Han, 2002; Liu, Slotine, & Barabási, 2012; Lou, Wu, Li, Wang and Chen, 2021; Schneider, Moreira, Andrade, Havlin, & Herrmann, 2011; Shargel, Sayama, Epstein, & Bar-Yam, 2003; Wang & Liu, 2019; Wang, Liu, & Jin, 2020, 2021; Xiao, Lao, Hou, & Bai, 2014). Network connectivity is fundamental and essential to support other functions, including such as controllability (Xiang, Chen, Ren, & Chen, 2019), synchronizability (Chen, 2022), and the abilities of communication and transmission. In this regard, the subject of network connectivity robustness is of fundamental and practical importance, which has been extensively investigated with applications to, for

* Corresponding author at: College of Computer Science, Sichuan Normal University, Chengdu, 610066, China.
E-mail addresses: felix.lou@ieee.org (Y. Lou), vridge@foxmail.com (R. Wu), lijunli@sicnu.edu.cn (J. Li), wanglin@sjtu.edu.cn (L. Wang), tangcb@zjnu.edu.cn (C.-B. Tang), eegchen@cityu.edu.hk (G. Chen).

https://doi.org/10.1016/j.neunet.2022.10.013
0893-6080/

example, wireless sensor networks (Qiu, Liu, Si, & Wu, 2019), transportation networks (Yang, Mao, Qian, & Wei, 2018), power grids (Chen, Wu, Xia, & Zhang, 2017), and nervous systems (Yan et al., 2017), among many others. In general, destructive attacks and random failures take place in the forms of node- and edge-removals, which may significantly change the network structure and cause severe consequences such as malfunctioning or even system crashing. In these situations, the ability of a network to maintain its connectedness against attacks or failures is usually referred to as the *connectivity robustness*, or briefly the *robustness* in this paper.

Regarding the attacks, real-world attacks and failures are modeled as *targeted* and *random* attacks in computer simulations. Targeted attacks remove some selected nodes, or edges, while random attacks do so at random. Some centrality measures such as betweenness, degree, and eigenvector-based measures are commonly used in targeted attack simulations (Iyer, Killingback, Sundaram, & Wang, 2013).

Regarding the robustness, network connectivity robustness is widely measured by recording the changes of the portion of nodes in the largest connected component (LCC) (Schneider et al., 2011) that survives from an attack. A network is deemed more robust against an attack, if it can maintain a higher value of the fraction of LCC nodes. The investigation and optimization of this measure emphasize on protecting the LCC. On the contrary, network disintegration aims at identifying and then removing a minimum set of nodes or edges, referred to as critical nodes or critical edges respectively, whose removal will lead to a maximum destruction to the network connectivity (Qi, Deng, Deng, & Wu, 2018; Zhang, Wu, Wang, Xiong, & Yang, 2016).

Given certain practical constraints, e.g., fixed numbers of nodes and edges, or fixed degrees for nodes, network robustness can be enhanced by edge rewiring (Bai, Xiao, Hou, & Lao, 2015; Chan & Akoglu, 2016; Lou, Xie, & Chen, 2020; Louzada, Daolio, Herrmann, & Tomassini, 2013; Schneider, Yazdani, Araújo, Havlin, & Herrmann, 2013; Wang & Liu, 2019; Wang et al., 2020, 2021; Wu & Holme, 2011; Zeng & Liu, 2012), which however imposes disturbances onto the network structures. After some edge rewiring operations, whether such disturbance enhances the robustness or not has to be evaluated, typically by using very time-consuming attack simulations. As a remedy, several easy-to-access indicators, e.g. assortativity (Newman, 2003) and spectral measures (Perra & Fortunato, 2008), are adopted for estimating the robustness in network applications. It is found that onion-like structured heterogeneous networks with positive assortativity coefficients are robust against attacks (Hayashi & Uchiyama, 2018; Schneider et al., 2011; Tanizawa, Havlin, & Stanley, 2012; Wu & Holme, 2011). In this research direction, there are a large number of studies on various issues regarding network robustness, for many types of networks such as a network of networks (Dong et al., 2013) and multiplex networks (Min, Do Yi, Lee, & Goh, 2014), encouraging some real-world applications in e.g. power grids (Cuadra, Salcedo-Sanz, Del Ser, Jiménez-Fernández, & Geem, 2015; Schneider et al., 2013).

Deep neural networks, on the other hand, provides a useful tool for analyzing and optimizing network-related problems that are typically NP-hard. Successful deep learning applications on complex networks include controllability robustness prediction (Dhiman, Sun, & Kooij, 2021; Lou, He, Wang, & Chen, 2022a; Lou, He, Wang, Tsang and Chen, 2022b) and critical node identification (Fan, Zeng, Sun, & Liu, 2020; Grassia, De Domenico, & Mangioni, 2021). Convolutional neural network (CNN) (Schmidhuber, 2015), as a kind of effective deep neural networks, is able to automatically and fast analyze inner features of a dataset and output desirable results about classification and regression.

Main advantages of using CNN-based algorithms include: (1) both classification and regression tasks can be managed. CNN can provide not only precise classifications at different levels, such as node classification and network classification, but more importantly also fast and good approximations of network performances, such as controllability robustness and connectivity robustness against malicious attacks (Lou et al., 2022a; Lou, Wu et al., 2021). (2) The CNN-based straightforward prediction framework provides a simple and fast method. Since there are no embedding nor aggregation operations needed, the CNN-based prediction can perform much faster than other powerful yet time-consuming algorithms, such as graph neural network (Hamilton, Ying, & Leskovec, 2017; Niepert, Ahmed, & Kutzkov, 2016). (3) The performance of CNN-based approach is reliable and stable: First, network data of different topologies, directed and undirected, as well as weighted and unweighted, *all* are acceptable as input to a CNN for processing. Second, CNN-based approach is shift-invariant in dealing with network-converted gray-scale images (Zhang, 2019), i.e., shuffling and transposing pixels of an image (while keeping the network topology unchanged) does not degrade the performance of the CNN-based predictions (Lou, He et al., 2022b; Lou, Wu et al., 2021). Third, it has been experimentally demonstrated that single CNN is tolerable if the network size is about $\pm 7.29\%$ different from the input size (Wu, Lou, Wu, Liu, & Li, 2022). To that extent, the prediction performance will not be significantly degraded.

The CNN-based robustness predictor (CNN-RP) designed in Lou, Wu et al. (2021) significantly improves the computational efficiency, compared to traditional attack simulations, but some issues may be considered for further improvement. For example, CNN-RP uses a single CNN and ignores useful prior knowledge of the network topology even when it is available. In this paper, a multi-CNN-based robustness predictor (mCNN-RP) is proposed, which takes advantage of prior knowledge of the network topology to improve the efficiency and performance of the successful CNN-based approach.

Specifically, the main contributions of this work are summarized as follows:

1. Prior knowledge of network topology is extracted and utilized for pre-processing. A multi-CNN structure is proposed, where each CNN acts as either a predictor or a classifier. The CNN classifier classifies the input network into different categories according to the available prior knowledge, and each CNN predictor works for a specific network type. Both the classifier and the predictors are insensitive to the change of network categorization.
2. A data-driven filter is trained using both training dataset and available prior knowledge about network connectivity robustness, which further refines the prediction results.
3. The effectiveness of mCNN-RP is experimentally verified for predicting network robustness in various scenarios, including synthetic and real-world networks, directed and undirected networks, as well as weighted and unweighted networks, which are under targeted and random attacks.
4. mCNN-RP also provides a better connectivity robustness estimation than the conventional spectral measures and CNN-RP.

The following text is organized as follows: Section 2 reviews the measures of network connectivity robustness against destructive node attacks. Section 3 introduces the new mCNN-RP scheme. Experimental results are presented in Section 4, with analysis and comparison. Finally, Section 5 concludes the investigation.

2. Network connectivity robustness

The *connectedness* and *weak connectedness* are employed as measures of the network connectivity for undirected and directed networks, respectively, while strong connectedness of directed networks is not discussed in this paper. Only node-removal attacks are investigated, while edge-removal attacks can be studied in a similar manner.

2.1. Robustness measures

During a node-removal attack process, the originally-connected network may be destructed to be of several components, where each component is internally connected but is disconnected from other components in the network. In the process, all the separated components are put together and considered as a whole system, among which the largest one (LCC (Schneider et al., 2011)) is regarded as the most important one. Specifically, the fraction of nodes in LCC, or in the normalized LCC (NLC), is calculated as

$$s(i) = \frac{c(i)}{N-i}, \quad i = 0, 1, \ldots, N-1, \tag{1}$$

where $c(i)$ is the number of nodes in LCC after a total number of i nodes have been removed from the network; $s(i)$ is the NLC after a total number of i nodes removed; N is the original number of nodes in the network before being attacked. Overall, the measure for the network robustness is calculated by

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

With the above-defined measure, for two networks under the same sequential attacks, the one with a larger $\bar{s}$ value has better connectivity robustness.

Different from LCC, the number of connected components (NCC) emphasizes on the number of disintegrated parts. Let $d(i)$ $(i = 0, 1, \ldots, N-1)$ denote the number of connected components after a total number of i nodes have been removed from the network. The possible range of NCC is wide, namely $1 \le d(i) \le N-i$. It is also important to determine the peak value of $d(i)$, namely $d_{\max} = \max\{d(i)\}$, which reflects how severely a network can be destructed by a certain attack strategy. Successive node-removals from a connected network will result in disintegration and consequently the NCC will increase. However, when a network has been severely decomposed into many small components, it is very likely that further node-removals will continue delete more connected components (e.g., isolated nodes); therefore, NCC will decrease accordingly. Thus, $d_{\max}$ offers a good reference to the network disintegration.

2.2. Connectivity and controllability robustness

Different from the predictors for controllability robustness (Lou et al., 2022a; Lou, He et al., 2022b), here the CNNs are used to predict the connectivity robustness against destructive attacks, which turns out to have a greater variation than the controllability robustness (Lou, Wu et al., 2021). A single node-removal will at most increase the number of needed driver nodes (a common measure of network controllability) by one. In contrast, a node-removal here may cause a significant change of the size (and the fraction) of LCC. To deal with such large variations, a multi-CNN framework is employed, where there is a specific CNN responsible for predicting the connectivity robustness of a specific network type, and moreover a filter is installed to rectify and smooth the CNN output.

Table 1
Parameter settings of the seven groups of convolutional layers.

Groups	Layer	Kernel size	Stride	Output shape	Output channel
Group 1	Conv7–64	7 × 7	1	994 × 994	64
	Max2	2 × 2	2	497 × 497	64
Group 2	Conv5–64	5 × 5	1	493 × 493	64
	Max2	2 × 2	2	246 × 246	64
Group 3	Conv3–128	3 × 3	1	244 × 244	128
	Max2	2 × 2	2	122 × 122	128
Group 4	Conv3–128	3 × 3	1	120 × 120	128
	Max2	2 × 2	2	60 × 60	128
Group 5	Conv3–256	3 × 3	1	58 × 58	256
	Max2	2 × 2	2	29 × 29	256
Group 6	Conv3–256	3 × 3	1	27 × 27	256
	Max2	2 × 2	2	13 × 13	256
Group 7	Conv3–512	3 × 3	1	11 × 11	512
	Conv3–512	3 × 3	1	9 × 9	512
	Max2	2 × 2	2	4 × 4	512

2.3. Error measures

Given three NLCs, let $\mathbf{s}_t = [s_t(0), s_t(1), \ldots, s_t(N-1)]$ denote the true curve obtained by attack simulations, and $\mathbf{s}_1 = [s_1(0), s_1(1), \ldots, s_1(N-1)]$ and $\mathbf{s}_2 = [s_2(0), s_2(1), \ldots, s_2(N-1)]$ represent two predicted curves, respectively. The prediction error is calculated by

$$\boldsymbol{\xi}_\alpha = |\mathbf{s}_t - \mathbf{s}_\alpha|, \tag{3}$$

where $\boldsymbol{\xi}_\alpha = [\xi_\alpha(0), \xi_\alpha(1), \ldots, \xi_\alpha(N-1)]$ is the sequence of errors between the two curves, where $\alpha = 1$ or 2, $\xi_\alpha(i) = |s_t(i) - s_\alpha(i)|$, $i = 0, 1, \ldots, N-1$.

The average error $\bar{\xi}_\alpha$ is then calculated by

$$\bar{\xi}_\alpha = \frac{1}{N}\sum_{i=0}^{N-1} \xi(i)_\alpha. \tag{4}$$

The vector $\boldsymbol{\xi}$ can be used to visualize the prediction errors throughout the attack process. The scalar $\bar{\xi}$ measures the overall prediction error, i.e., $\bar{\xi}_1 < \bar{\xi}_2$ means that the predicted curve $\mathbf{s}_1$ obtains lower prediction error than $\mathbf{s}_2$.

3. Predictor for network connectivity robustness

Given some prior knowledge, networks can be first classified into several categories, and then the connectivity robustness can be predicted using the corresponding predictor. Given L network types that are commonly used, the users are able to prepare a specific CNN predictor for each type, other than treating these networks in some general forms. There are $(L+2)$ CNNs in mCNN-RP, where one CNN is employed as the classifier, one CNN as a general predictor that does not have any specific knowledge about the network types, and all the other L CNNs as predictors for the L network types, respectively.

3.1. Convolutional Neural Network

CNN is employed as a classifier and also a predictor in this work. Fig. 1 shows the CNN structures, where the new design is that an extra softmax (Bishop, 2006) layer is installed in the end of the classifier, but there is no such a layer for a predictor.

Fig. 1 shows the detailed architecture of the CNN (following the VGG architecture (Simonyan & Zisserman, 2014)), with parameters summarized in Table 1. Given an input size 1000 × 1000, seven feature map (FM) groups are installed, while more FMs

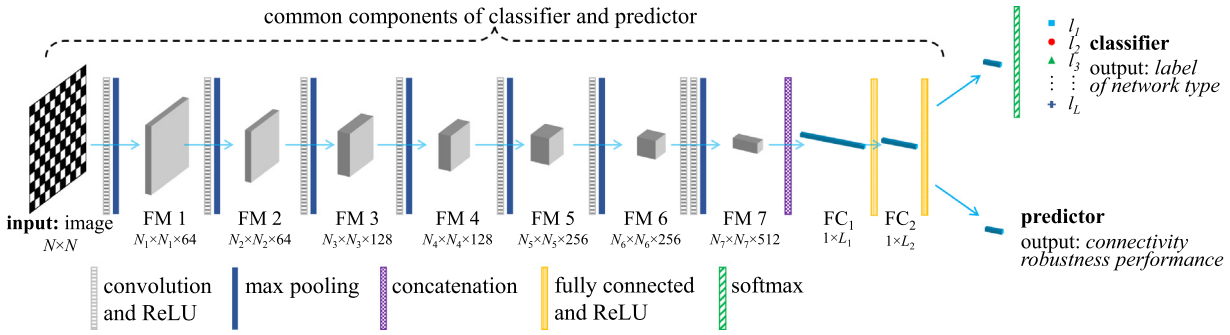

Fig. 1. [Color online] The CNN architecture for both classifier and predictor. FM stands for *feature map* and FC for *fully connected*. Data sizes are $N_i = \lceil N/2^{(i+1)} \rceil$, $i = 1, 2, \ldots, 7$. The concatenation layer rearranges the matrix into a vector, from FM 7 to FC 1, i.e., $L_1 = N_7 \times N_7 \times 512$. $L_2 \in (L_1, N-1)$ is a hyperparameter. In this paper, $L2 = 4096$. For a classifier, an additional softmax is installed following FC_2. The output of the classifier is a predicted label of the input data. For a predictor, the output is a predicted robustness curve (an N-vector), which is output from FC_2.

are needed for a greater input size. Each group of FM1–FM6 contains a convolution layer, ReLU, and a max pooling layer, while FM7 consists of two convolution layers with two ReLUs and one max pooling layer. One ReLU implements an activation function, typically $f(x) = \max(0, x)$ (Glorot, Bordes, & Bengio, 2011). The output of each hidden layer (multiplication of weights) is added together to be rectified by a ReLU and then transmitted to the next layer. To that end, the max pooling layer will reduce the dimension of the dataset as input to the next layer.

One CNN is employed as the classifier in mCNN-RP, the output of which is a vector of L real numbers, l_i $\left(\sum_{i=1}^{L} l_i = 1\right)$, representing the score of the input image belonging to network type i ($i \in \{1, 2, \ldots, L\}$). Here, a threshold $\eta = 0.8$ is set according to Lou, He et al. (2022b), i.e., for $i = 1, 2, \ldots, L$, only if $l_i \geq \eta$. Then, it will return an indication that the corresponding input image is classifiable and belongs to type i; otherwise, it will return an indication that the input is non-classifiable. If a threshold is too low, it will decrease the success rate; if the threshold is too high, it may result in many non-classifiable cases. In the simulations here, η is set to 0.8, which yields a successful rate higher than 0.90 in classification, but it is not sensitively affected when η is slightly changed (Lou, He et al., 2022b). The other $(L + 1)$ CNNs are employed as predictors; the output of each predictor is an N-vector, representing the predicted connectivity curve under certain attacks. There is no softmax layer installed into the predictors. In this design, the internal parameters of different CNNs (a classifier with $(L+1)$ predictors) will be trained differently.

The cross entropy is used as the loss function in the classifier. For given types of input predicted and true network, denoted by $\hat{l}$ and l respectively. The mean-squared error between the predicted connectivity robustness curve $\hat{s}$ and the true curve s will be used as the loss function for prediction. These two loss functions are calculated by

$$\mathcal{H} = -\sum_{i=1}^{L} l(i) \cdot \log[\hat{l}(i)], \tag{5}$$

and

$$\mathcal{L} = \frac{1}{N} \sum_{i=0}^{N-1} \|s(i) - \hat{s}(i)\|, \tag{6}$$

where $\|\cdot\|$ is the Euclidean norm. The training process for a CNN (either a classifier or a predictor) aims at adjusting the internal parameters, with the objective of minimizing either the cross entropy (Eq. (5)) or mean-squared error (Eq. (6)).

Previous investigations have shown that the CNN-based approaches not only have good tolerance if the input images have information loss up to 7.29%, but also perform consistently well regardless of the images being shuffled or not. Shuffling an network-converted image means to randomly exchange some rows and columns of pixels, such that the image will become totally different but the corresponding network will remain the same (an isomorph). Visible features of networks due to a certain generation mechanism can be filtered by shuffling (Lou, He et al., 2022b; Lou, Wu et al., 2021), but this will not degrade the CNN prediction performance.

3.2. Rectify the CNN output

For a CNN predictor, it performs regression task in a data-driven manner, which does not guarantee the logic correctness of the output. In addition, the variation of connectivity values during the attack process is large. Therefore, a filter is installed to rectify the data from two aspects: (1) the size of LCC under attacks is restricted within a certain range, and (2) the change of the size of LCC under attacks is monotonically non-increasing.

A filter $\hat{s}(i) = \max\{1, \hat{s}(i)\}$ and $\hat{s}(i) = \min\{N - i, \hat{s}(i)\}$ was designed in Lou, Wu et al. (2021) to rectify the illogical data, where $\hat{s}(i)$ represents the ith value of the predicted vector $\hat{\mathbf{s}}$. However, this prior knowledge-based filter offers a very rough boundary structure.

In the proposed mCNN-RP, a data-based filter is installed instead of the prior knowledge-based one. The upper and lower boundaries are set according to the training set, namely $ub(i) = \max\{ts^j(i)\}$ and $lb(i) = \min\{ts^j(i)\}$, where $ts^j(i)$ represents the ith value of ts^j that is the jth training sample. Thus, the output is rectified as follows:

$$\hat{s}(i) = \begin{cases} ub(i), & \text{if } \hat{s}(i) > ub(i), \\ lb(i), & \text{if } \hat{s}(i) < lb(i). \end{cases} \tag{7}$$

To ensure the monotonically non-increasing feature, interpolation is applied as in Lou, Wu et al. (2021), namely, if a data violation is detected as $\hat{s}(k) > \hat{s}(i)$ for $k > i$, then continue to search along $j = k+1, k+2, \ldots$, until $\hat{s}(j) < \hat{s}(i)$ is detected, and $\hat{s}(k)$ is rectified as follows:

$$\hat{s}(k) = \hat{s}(i) + \frac{k-i}{i-j} \cdot (\hat{s}(i) - \hat{s}(j)), \tag{8}$$

where the integers i, j, and k satisfy $i < k < j$, and $j - i \geq 2$.

Since $\hat{s}(i)$, $\hat{s}(j)$, and $\hat{s}(k)$ are all predicted values, the correctness of the prediction is not checked in this step. The filter installed in mCNN-RP consists of Eqs. (7) and (8), which rectifies the logically unreasonable data. Note that only the size of LCC is monotonically non-increasing during sequential attacks, but the fraction of nodes in LCC (namely NLC), as shown in Eq. (1), may not be so.

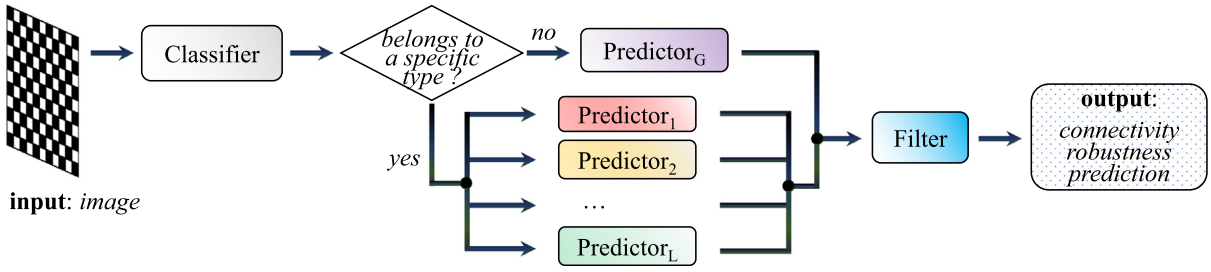

Fig. 2. Framework of mCNN-RP, where the input is a network-converted image and the output is the corresponding robustness curve under a certain type of attacks. OneCNN is employed as classifier, one CNN as predictor for general networks (denoted by CNN_G) and L CNN as predictors for L specific network types. A filter is installed to rectify the CNN output.

3.3. Framework of mCNN-RP

The framework of mCNN-RP is shown in Fig. 2, which consists of $(L+2)$ CNNs, including a classifier and $(L+1)$ predictor. Given an input image, the classifier first checks whether it is classifiable: if yes, then the input, which is classified as a type l network, will be passed to predictor l ($l = \{1, 2, \ldots, L\}$); otherwise, the input will be passed to the predictor trained using a general neural network. All predictors have the same CNN structure but are trained by different datasets; therefore, their internal parameters are usually very different.

It was reported in Lou, He et al. (2022b) that different prior knowledge will lead to different network categorizations, which will further result in different performances of classification and prediction. In this paper, information about the network topology is utilized as prior knowledge.

4. Experimental studies

Four synthetic network models are simulated, including the Erdös–Rényi (ER) random-graph (Erdös & Rényi, 1964), generic scale-free (SF) (Goh, Kahng, & Kim, 2001; Pu, Pei, & Michaelson, 2012), q-snapback (QS) (Lou, Wang and Chen, 2018), and Newman–Watts small-world (SW) (Newman & Watts, 1999) networks. Both directed and undirected, as well as weighted and unweighted synthetic networks are generated using these models. Specifically, an ER network is generated by connecting each pair of randomly-picked nodes, with a probability $p_{ER} \in (0, 1]$. An SF network is generated using predefined weights, namely the probability of connecting two nodes i and j is proportional to their weights w_i and w_j, where $w_i = (i + \theta)^{-\sigma}$ for any node i, parameters $\sigma \in [0, 1)$ and $\theta \ll N$. QS consists of a directed backbone chain and multiple snapback edges; the density of snapback edges is determined by a parameter q (Lou, Wang et al., 2018). SW consists of an N-node loop having K ($K = 2$ here) connected nearest-neighbors, with shortcuts among the nodes (Newman & Watts, 1999). All the network data are shuffled such that the intrinsic features due to the generation processes are filtered out. Detailed generation methods for these network models can be found from Lou et al. (2022a) and Lou, He et al. (2022b).

For each synthetic network topology, 1600 instances are randomly generated for training, thus there are $1600 \times 4 = 6400$ training samples in total. All networks have the same size $N = 1000$. For each network instance in either training or testing data, its average degree is randomly drawn from a uniform-random distribution $\langle k \rangle \in [4, 8]$. The averaged degree of the training network is 6.01, while that of the testing network is 5.94, with data obtained by performing posterior statistics.

mCNN-RP is compared with CNN-RP (Lou, Wu et al., 2021) in predicting the connectivity robustness for both synthetic and real-world networks under various attacks, including random attack (RA), targeted betweenness-based (TB), and targeted degree-based (TD) attacks. All experiments are performed on a PC Intel (R) Core i7-8750H CPU @ 2.20 GHz, which has memory (RAM) 16 GB with running Windows 10 Home 64-bit Operating System.

Table 2
Average prediction errors of mCNN-RP and CNN-RP, and standard deviation (*std*) of the testing data. A '(+)' denotes mCNN-RP significantly outperforms CNN-RP by obtaining lower errors, while a '(≈)' means no significant difference between the two methods.

		ER	SF	SW	QS
RA	mCNN-RP	0.0305 (+)	0.0462 (+)	0.0381 (≈)	0.0329 (+)
	CNN-RP	0.0334	0.0511	0.0398	0.0371
	std	0.0364	0.0678	0.0460	0.0405
TD	mCNN-RP	0.0517 (+)	0.0097 (+)	0.0470 (≈)	0.0463 (+)
	CNN-RP	0.0521	0.0216	0.0500	0.0497
	std	0.0613	0.0279	0.0572	0.0573
TB	mCNN-RP	0.0529 (≈)	0.0133 (+)	0.0542 (≈)	0.0569 (≈)
	CNN-RP	0.0564	0.0288	0.0545	0.0572
	std	0.0640	0.0332	0.0659	0.0703

4.1. Directed unweighted synthetic networks

Two connectivity robustness measures are predicted, namely the fraction of nodes in LCC and the number changes of the connected components during the attacking process. Fig. 3 shows the prediction results of mCNN-RP and CNN-RP, together with the attack simulation results (*tv*). The shadow in the same color represents the range of standard deviation. It is shown in Fig. 3(a), (b), (c), (e), (i), and (l) that the mCNN-RP prediction is clearly closer to the true curve than the CNN-RP prediction, while in the rest subplots, these two methods give very similar results. It is also shown in Fig. 3 that the prediction errors of mCNN-RP and CNN-RP are clearly less than the standard deviation of the testing data.

The curves shown in Fig. 3 demonstrate that mCNN-RP and CNN-RP can predict the connectivity robustness precisely; while the detailed overall prediction performance comparison is shown in Table 2, which demonstrates that the mCNN-RP and CNN-RP prediction errors are lower than the standard deviation of testing data. mCNN-RP obtains lower average prediction errors than CNN-RP in *all* the 12 sets of testing data, among which mCNN-RP obtains significantly lower prediction errors than CNN-RP in 7 sets of data (denoted by 7 (+)'s in Table 2). There is no instance that CNN-RP outperforms mCNN-RP. Each datum in Table 2 is obtained by averaging 400 testing results. Prediction errors are calculated using Eq. (4). The significance is checked using the Mann–Whitney U-test (Fay & Proschan, 2010) with a significance level of 0.05.

Other than LCC, the NCC tendency is also considered as a measure of connectivity robustness in this paper. The predicted NCC curves of the four networks under TD attacks are shown in Fig. 4, where mCNN-RP predicts better NCC curves than CNN-RP for all four networks. Detailed NCC prediction errors are shown

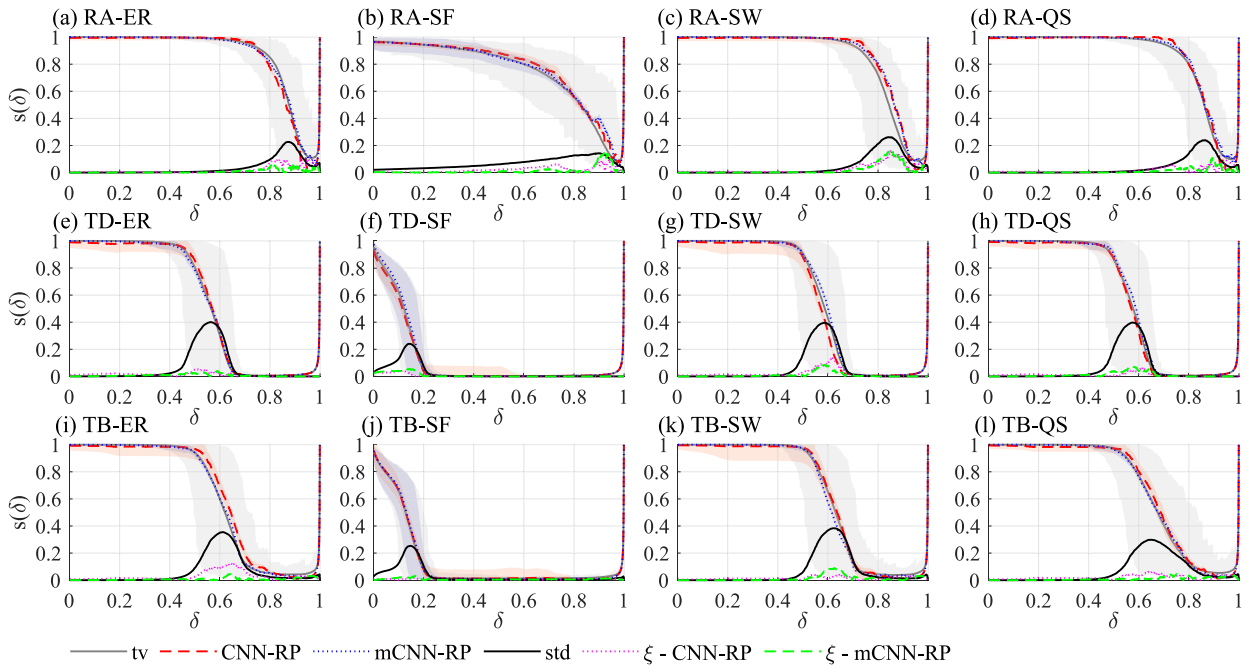

Fig. 3. Comparison of mCNN-RP and CNN-RP in predicting the network connectivity robustness. δ and $s(\delta)$ represent the proportion of removed nodes and the corresponding NLC, respectively; *tv* represents the true values obtained from attack simulations; *std* represents the standard deviation of the testing data; ξ denotes the prediction error. (a–d) ER, SF, SW, and QS networks under RA; (e–h) ER, SF, SW, and QS networks under TD; (i–l) ER, SF, SW, and QS networks under TB. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

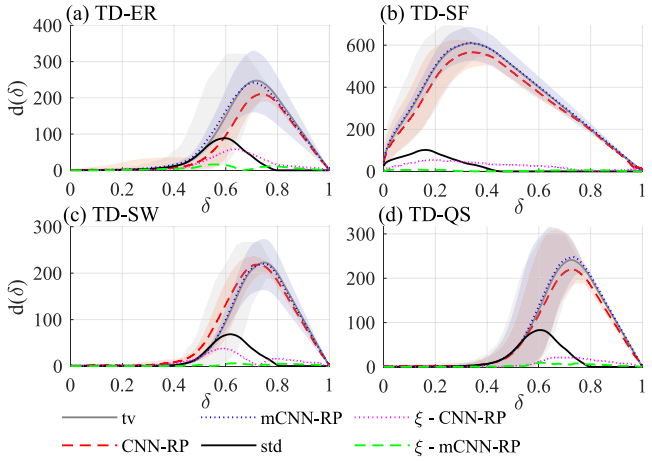

Fig. 4. [color online] Comparison mCNN-RP and CNN-RP in predicting the NCC values. δ and $d(\delta)$ represent the proportion of removed nodes and the corresponding NCC curves, respectively; *tv* denotes the true values obtained from attack simulations; *std* denotes the standard deviation of the testing data; 'ξ - CNN-RP' and 'ξ - mCNN-RP' represent the prediction errors of CNN-RP and mCNN-RP, respectively.

Table 3
The average prediction errors of mCNN-RP and CNN-RP, and the standard deviation (*std*) of the NCC prediction under TD attacks. A '(+)' denotes mCNN-RP significantly outperforms CNN-RP, while a '(≈)' means no significant difference between the two methods.

NCC	ER	SF	SW	QS
mCNN-RP	16.3558 (≈)	23.1038 (+)	11.0865 (+)	3.7083 (+)
CNN-RP	17.358	26.7253	13.3788	8.4269
std	19.243	25.6414	13.4879	17.084

in Table 3, where the U-test results show that mCNN-RP predicts significantly better than CNN-RP in 3 out 4 networks.

Table 4
Prediction errors of the peak values of NCC ($d_{\max}$) obtained by mCNN-RP and CNN-RP. A '(+)' denotes that mCNN-RP obtains significant lower prediction errors than CNN-RP, while a '(≈)' means no significant differences.

$d_{\max}$	ER	SF	SW	QS
mCNN-RP	17.0 (+)	8.4 (+)	12.5 (≈)	9.6 (+)
CNN-RP	41.7	38.8	12.3	19.6

Fig. 4 shows that the maximum value of NCC can be up to several hundreds for a 1000-node network during attacks, and Table 4 shows that the errors of the predicted maximum value of NCC by mCNN-RP can be lower than 20, which are significantly lower than the prediction errors obtained by CNN-RP in 3 out 4 networks.

The runtime of mCNN-RP is about twice of that of CNN-RP, since mCNN-RP classifies the network data before predicting, while CNN-RP predicts the results directly. Compared to the traditional attack simulations, both mCNN-RP and CNN-RP significantly reduce the runtime in collecting the connectivity robustness curves. For example, given the same computer configuration as described above, for ER networks with $N = 1000$ and $\langle k \rangle \in [4, 8]$ under random attacks, the average runtime for attack simulations is 11.65 s, while for CNN-RP and mCNN-RP it is only 0.12 and 0.25 s respectively.

4.2. Undirected and weighted synthetic networks

Although the curves in Fig. 5 show that both mCNN-RP and CNN-RP can predict the connectivity robustness curves close to the true curves, the average prediction errors in Table 5 show that mCNN-RP performs predictions significantly better than CNN-RP, for both undirected and weighted networks under random attacks (RA). The significance is checked using the Mann–Whitney U-test (Fay & Proschan, 2010) with a significance level of 0.05. Both mCNN-RP and CNN-RP predict the connectivity robustness

Table 5
Average prediction errors of mCNN-RP and CNN-RP, and standard deviation (*std*) of undirected (UD) and weighted (W) networks under RA. A '(+)' denotes mCNN-RP obtains significantly lower prediction error than CNN-RP.

Undirected		UD-ER	UD-SF	UD-SW	UD-QS
	mCNN-RP	0.0303 (+)	0.0488 (+)	0.0376 (+)	0.0326 (+)
	CNN-RP	0.0337	0.0630	0.0382	0.0373
	std	0.0377	0.0661	0.0441	0.0414
Weighted		W-ER	W-SF	W-SW	W-QS
	mCNN-RP	0.0287 (+)	0.0542 (+)	0.0381 (+)	0.0316 (+)
	CNN-RP	0.0322	0.0585	0.0500	0.0418
	std	0.0364	0.0678	0.0460	0.0405

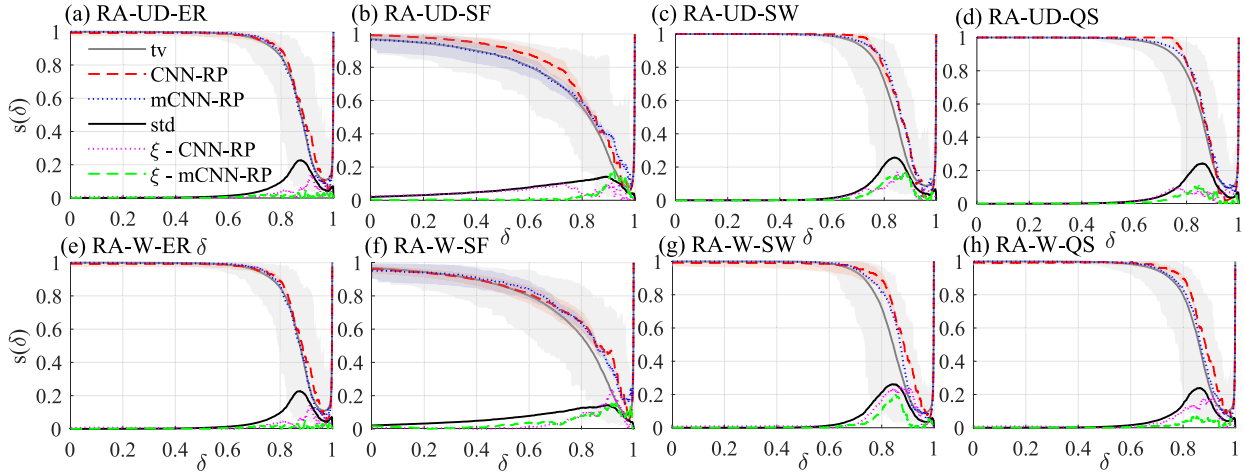

Fig. 5. [color online] Comparison of mCNN-RP and CNN-RP in predicting the robustness of undirected (UD) and weighted (W) networks under random attacks (RA). δ and $s(\delta)$ represent the proportion of removed nodes and the corresponding NLC curves, respectively; *tv* denotes the true values obtained from attack simulations; *std* denotes the standard deviation of the testing data; 'ξ - CNN-RP' and 'ξ - mCNN-RP' represent the prediction errors of CNN-RP and mCNN-RP, respectively.

with average errors lower than the standard deviation of the testing data.

As shown in Table 5, for both undirected and weighted synthetic networks, the performances of mCNN-RP and CNN-RP are consistently good, reflected by lower error values than the standard deviation. The average runtime of calculating the robustness of the undirected and weighted networks by attack simulation is 12.36 s, while for mCNN-RP and CNN-RP, the average runtime of prediction is 0.29 s and 0.16 s, respectively. In a nutshell, mCNN-RP obtains significantly lower prediction errors than CNN-RP, at the cost of the slightly increased average runtime.

4.3. Real-world networks

Two hundred real-world networks collected from Networkrepository (Rossi & Ahmed, 2016) and Graphkernels (Sugiyama, Ghisu, Llinares-López, & Borgwardt, 2017) are used for testing. The network size varies with $N \in [950, 1050]$ and an average of $\bar{N} = 997$; the average degree varies with $\langle k \rangle \in [2.1, 6.7]$, and the average $\langle \bar{k} \rangle = 2.54$.

Both mCNN-RP and CNN-RP are trained using synthetic networks as in Section 4.1, mCNN-RP classifies each real-world network first, and then performs prediction; while CNN-RP predicts directly. For the 200 real-world networks, 4 networks are classified as ER networks, 122 as SF, 47 as SW, 18 as QS, and the rest 9 are non-classifiable. The classification correctness is not checked for real-world networks, since many networks process multiple characteristics, such as random-graph and small-world characteristics. Since classification correctness for synthetic networks is higher than 90% (Lou, Wu et al., 2021), the classifier is empirically reliable. The prediction errors are shown in Fig. 6, where mCNN-RP has lower prediction errors than CNN-RP during the attack process. The prediction errors of both mCNN-RP and

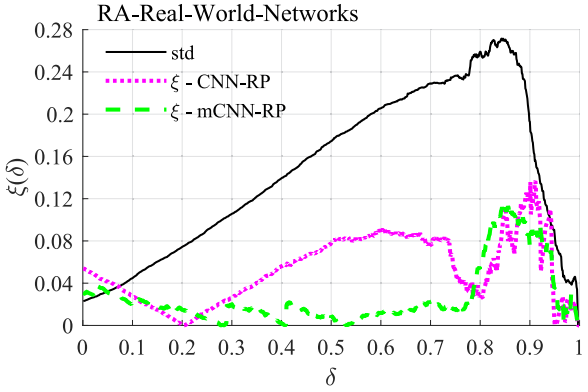

Fig. 6. [color online] Prediction errors of mCNN-RP and CNN-RP in predicting 200 real-world networks. δ and $\xi(\delta)$ represent the proportion of removed nodes and the corresponding prediction error values, respectively. 'ξ - CNN-RP' and 'ξ - mCNN-RP' represent the prediction errors of CNN-RP and mCNN-RP, respectively.

CNN-RP are clearly lower than standard deviation of the testing data. The average prediction error of mCNN-RP is 0.0614, which is significantly better than the errors of CNN-RP (0.1191) and the standard deviation of the 200 real-world network data (0.1451), using the Kruskal–Wallis H-test (Kruskal & Wallis, 1952; Lou, Yuen and Chen, 2018) with a significance level of 0.05.

Experimental results of real-world networks also verify that the CNN-based approaches are tolerable to small changes of network sizes. For mCNN-RP, unknown real-world networks are classified into different categories before predicting the robustness. The topological variance of the real-world network data can be large, but a single CNN may not be sufficient to learn the large variance. This pre-processing step reduces the learning burden

Table 6
Basic information of 6 real-world networks and the average prediction errors of mCNN-RP and CNN-RP.

		DBLP2	Movie lens-user	Grid yeast	C-elegance	PolBooks	Karate
Info	N	12 591	7602	6008	279	105	34
	$\langle k\rangle$	3.95	7.30	52.25	7.86	8.40	4.59
Prediction error	mCNN-RP	0.0632	0.1290	0.0555	0.1668	0.1262	0.0812
	CNN-RP	0.0618	0.1296	0.1077	0.2514	0.1705	0.1268

for each single CNN. Therefore, compared to CNN-RP, mCNN-RP significantly improves the prediction performance, especially for real-world networks.

To further verify the effectiveness of mCNN-RP, 6 real-world networks with either significantly greater or smaller sizes are tested, including DBLP,[1] MovieLens-User,[2] Grid Yeast,[3] C-elegance (Rossi & Ahmed, 2015), PolBooks,[4] and Karate.[5] The basic information and the prediction errors are shown in Table 6. The five networks including DBLP2, MovieLens-User, Grid Yeast, C-elegance, are PolBooks are classified as SF networks, while Karate is non-classifiable.

As shown in Table 6, mCNN-RP obtains slightly higher prediction error than CNN-RP for DBLP2, while the rest 5 real-world networks, mCNN-RP outperforms CNN-RP with consistently lower prediction errors. Note that the network size is $N = 1000$ for the training data, while the tested real-world networks possess very different network sizes. In general, networks with either significantly greater or smaller sizes are not suggested to be tested. Here, the simulation results shown in Table 6 demonstrate the performances of mCNN-RP and CNN-RP in extreme situations.

4.4. Prior knowledge and network categorization

If the full knowledge of network types are given in advance, namely the L types of networks are known, then the L corresponding CNNs can be separately trained for the L types of networks. Whereas if this information is unavailable, the network categorization can be achieved by performing pre-processing of the training data, for example K-means (MacQueen, 1967).

In the default experimental setting, the types of the synthetic networks are supposed to be known to mCNN-RP, which are ER, SF, SW, and QS. In this subsection, the scenario when the network type information is unavailable is investigated. For simplicity, only the scenario of directed networks under TD attacks is simulated and discussed. K-means clustering is used for pre-processing: For each network instance in the training data (6400 instances in total), the following five features (scalars) are calculated, including average node degree, average node betweenness heterogeneity (calculated by $h = \langle k\rangle^2/\langle k^2\rangle$, where $\langle k\rangle$ represents the average node degree and $\langle k^2\rangle$ represents the average of the squared node degree), average path-length, and average clustering coefficient. To this end, each network is represented by a 5-vector, and then K-means is performed for clustering.

Given different K values, the training network instances can be categorized accordingly. Set $L = K$, and then for each network cluster (category), a specific CNN is trained.

Table 7 shows the average prediction errors when K is set to 3, 4, 5, and 6, respectively, compared with the default categorization setting that the network type information in known. C#i ($i = 1, 2, \ldots, 6$) represents the corresponding average prediction error for each network cluster (category). Note that type C#1 networks may be very different from each other when K is set differently. As can be observed from the table, the average prediction error does not clearly change, either the network types are known in advance, or the information is unknown but a K-means clustering is performed.

This suggests that the $(L+1)$ predictor CNNs and the classifier CNN can be flexibly trained for different network types, when the number and definition of 'network type' are different.

More importantly, the prediction performance of mCNN-RP is insensitive to the parameter L, which can be flexibly set according to the prior knowledge at hand.

4.5. Compared to spectral measures

Six typical spectral measures, including algebraic connectivity (AC), effective resistance (EFR), natural connectivity (NAC), spectral gap (SG), spectral radius (SR), and spanning tree count (STC), are compared with mCNN-RP and CNN-RP in predicting the overall network robustness under TD attacks. Spectral measures are widely used to estimate the network connectivity robustness, especially for undirected networks, therefore are also adopted here. Details of spectral measures can be found in, e.g., Chan and Akoglu (2016).

The results shown in Table 8 are obtained by averaging the predicting values of the overall connectivity robustness from 1600 undirected networks (400 network instances for each synthetic network model), whose averaged degree varies, with $\langle k\rangle \in [3.63, 7.98]$ and an averaged average degree $\langle k\rangle = 5.78$.

Since different predictive measures return different values, they are unified by the predicted ranks of network robustness. Each measure returns a predicted rank-list of 1600 values. As a benchmark, the true ranks are obtained from attack simulations. The rank error σ_r is calculated by

$$\xi_r = |\hat{rl} - rl|\,, \tag{9}$$

where $\hat{rl}$ represents the predicted rank-list (by either a spectral measure, mCNN-RP or CNN-RP), and rl represents the true rank-list. For example, given $\hat{rl}_1 = [3, 5, 1, 2, 4]$, $\hat{rl}_2 = [1, 3, 5, 4, 2]$, and a true rank-list $rl = [2, 3, 1, 5, 4]$, the rank errors are obtained as $\sigma_{r1} = |\hat{rl}_1 - rl| = [1, 2, 0, 3, 0]$ and $\sigma_{r2} = |\hat{rl}_2 - rl| = [1, 0, 4, 1, 2]$, respectively. The numbers of '0' in σ_{r_1} and σ_{r_2} are counted as the correct ranks for $\hat{rl}_1$ and $\hat{rl}_2$, respectively. The average rank errors, max rank errors, and min rank errors are also calculated using σ_{r_1} and σ_{r_2}. Moreover, the number of network instances that are predicted to be within top 10% (in ordinal ranks of connectivity robustness) and also within top 10% in simulations is counted, with results included in the 'top 10%' column. Similarly, the numbers in the 'bottom 10%' column are obtained and listed.

As shown in Table 8, AC and mCNN-RP have the minimum average rank error 188, followed by CNN-RP with an average rank error 225. mCNN-RP obtains the lowest max rank error (756), followed by CNN-RP (835). AC, ERE, STC, CNN-RP and mCNN-RP have the min rank error 0; mCNN-RP predicts 5 ranks correctly, followed by CNN-RP which predicts 4 ranks correctly. Finally, STC, AC, mCNN-RP and CNN-RP predict a number of correct top 10% and bottom 10% networks, whose robustness values are truly top 10% and bottom 10% according to the simulation results.

[1] https://networkrepository.com/cit-DBLP.php.
[2] https://networkrepository.com/rec-movielens-user-movies-10~m.php.
[3] https://networkrepository.com/bio-grid-yeast.php.
[4] http://www.casos.cs.cmu.edu/computational_tools/datasets/external/polbooks/index11.php.
[5] http://konect.cc/networks/ucidata-zachary.

Table 7
Average prediction errors of mCNN-RP with different network categorizations. C#i ($i = 1, 2, \ldots, 6$) represents the corresponding average prediction error for each network cluster (category).

		C#1	C#2	C#3	C#4	C#5	C#6	Average error
K-means	$K = 3$	0.0457	0.0522	0.0497	–	–	–	0.0492
	$K = 4$	0.0457	0.0522	0.0426	0.0572	–	–	0.0494
	$K = 5$	0.0522	0.0407	0.0681	0.0497	0.0268	–	0.0475
	$K = 6$	0.0351	0.0590	0.0455	0.0426	0.0572	0.0568	0.0494
Synthetic model		ER	SF	SW	QS	–	–	–
		0.0548	0.0463	0.0491	0.0458	–	–	0.0490

Table 8
Prediction errors of the six spectral measures, mCNN-RP and CNN-RP. Bold numbers are results from the best performing prediction measures.

	Average rank error	Max rank error	Min rank error	Correct rank	Top 10%	Bottom 10%
AC	**188**	868	**0**	2	63	0
ERE	792	1563	**0**	2	0	0
NAC	599	1365	3	0	0	0
SG	599	1346	23	0	0	0
SR	599	1365	5	0	0	0
STC	255	1042	**0**	2	**104**	65
mCNN-RP	**188**	**756**	**0**	**5**	54	**135**
CNN-RP	225	835	**0**	4	51	127

Table 9
Comparison of prediction errors obtained by mCNN-RP without installing filter (NF), mCNN-RP with a filter proposed in Lou, Wu et al. (2021) (KF), and mCNN-RP with a data-based filter using Eqs. (7) and (8) (DF). Signs '+', '†', '‡' denote significant differences in performances, while a '≈' represents no significant difference.

	ER	SF	SW	QS
NF	0.0573	0.0149	0.0534	0.0538
KF	0.0546 (†)	0.0104 (†)	0.0502 (†)	0.0509 (†)
DF	0.0517 (+, ≈)	0.0097 (+, ‡)	0.0470 (+, ≈)	0.0463 (+, ‡)

The testing dataset contains 1600 networks, giving 160 networks ranked as top 10% and bottom 10%, respectively.

Results in Table 8 show that mCNN-RP returns good prediction results, better than CNN-RP and other spectral measures. More importantly, mCNN-RP and CNN-RP not only return the overall connectivity robustness, but also return the curves throughout the entire attack process. In a nutshell, spectral measures predict only an overall quantitative value of network connectivity robustness with negligible overheads, while mCNN-RP predicts a more detailed curve of connectivity robustness with a substantial amount of training data and overheads.

4.6. Utilities of the filter

The installed filter of mCNN-RP is expected to filter out some unreasonable returns by CNN and, meanwhile, reduce the prediction errors.

In this subsection, the prediction errors of mCNN-RP without installing filter (NF), mCNN-RP with a filter designed in Lou, Wu et al. (2021), namely knowledge-based filter (KF), and mCNN-RP with a data-based filter (DF), are compared.

Table 9 shows the prediction errors under TD attacks. For all network topologies, DF obtains the lowest average error among the three methods. A '†' denotes that KF significantly outperforms NF; while a '+' denotes that DF significantly outperforms NK. DF consistently obtains lower prediction errors as comparing to KF. A '‡' denotes that DF significantly outperforms KF; while a '≈' means that no significant difference between DF and KF is detected.

Although precision check or precision enhancement is not the utility of the filter, it is observed that the prediction performance is significantly improved after installing the new filter DF.

5. Conclusions

In this paper, a fast and effective multiple CNN-based approach is proposed for predicting the connectivity robustness of general complex networks against malicious node-removal attacks. Conventionally, the network robustness is evaluated by computationally time-consuming attack simulations, from which a sequence of network connectedness values are generated as record to measure the connectivity of the remaining network after a sequence of destructive attacks. Compared to CNN-RP that uses a single CNN to predict all types of network input, the proposed mCNN-RP uses prior knowledge of the network topologies. Given L known network types, mCNN-RP consists of $(L+2)$ CNNs, where one CNN is used as the classifier, one CNN as a general predictor that does not have any specific knowledge about the network types, and the other L CNNs as predictors for the L network types, respectively. The performance of mCNN-RP is insensitive to the change of network categorization and the change of the value L. The computational cost of mCNN-RP is about twice of that of CNN-RP, both are much lower than the conventional attack simulations. A data-based filter is designed for the refinement of the prediction data. Extensive numerical experiments are performed using synthetic and real-world networks, including directed and undirected, as well as weighted and unweighted ones, demonstrating the effectiveness of mCNN-RP in prediction performances: (1) mCNN-RP significantly reduces the prediction errors of connectivity robustness for real-world networks, by classifying the real-world networks into different types and predicting using a specific CNN. (2) For synthetic networks, including directed and undirected, weighted and unweighted ones, mCNN-RP outperforms CNN-RP in prediction performance. Both predictors obtain lower average errors than the standard deviation of the testing dataset. (3) mCNN-RP provides a better predictive measure than CNN-RP and several powerful spectral measures. This paper has demonstrated once again that the CNN-based prediction technique has a great potential for a wide range of applications to complex networks.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

This work was supported by the National Natural Science Foundation of China [No. 62002249, 61873167]; the Foundation of Key Laboratory of System Control and Information Processing, Ministry of Education, P. R. China [No. Scip202103]; and the Hong Kong Research Grants Council under the GRF Grant CityU11206320,

References

Bai, L., Xiao, Y.-D., Hou, L.-L., & Lao, S.-Y. (2015). Smart rewiring: Improving network robustness faster. *Chinese Physics Letters*, *32*(7), Article 078901.

Barabási, A.-L. (2016). *Network science*. Cambridge University Press.

Bashan, A., Berezin, Y., Buldyrev, S., & Havlin, S. (2013). The extreme vulnerability of interdependent spatially embedded networks. *Nature Physics*, *9*, 667–672.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Chan, H., & Akoglu, L. (2016). Optimizing network robustness by edge rewiring: A general framework. *Data Mining and Knowledge Discovery*, *30*(5), 1395–1425.

Chen, G. (2022). Searching for best network topologies with optimal synchronizability: A brief review. *IEEE/CAA Journal of Automatica Sinica*, *9*(4), 573–577.

Chen, G., & Lou, Y. (2019). *Naming game: models, simulations and analysis*. Springer.

Chen, G., Wang, X., & Li, X. (2014). *Fundamentals of complex networks: models, structures and dynamics* (2nd ed.). John Wiley & Sons.

Chen, Z., Wu, J., Xia, Y., & Zhang, X. (2017). Robustness of interdependent power grids and communication networks: A complex network perspective. *IEEE Transactions on Circuits and Systems II: Express Briefs*, *65*(1), 115–119.

Cuadra, L., Salcedo-Sanz, S., Del Ser, J., Jiménez-Fernández, S., & Geem, Z. W. (2015). A critical review of robustness in power grids using complex networks concepts. *Energies*, *8*(9), 9211–9265.

Dhiman, A., Sun, P., & Kooij, R. (2021). Using machine learning to quantify the robustness of network controllability. In *International conference on machine learning for networking* (pp. 19–39). Springer.

Dong, G., Gao, J., Du, R., Tian, L., Stanley, H. E., & Havlin, S. (2013). Robustness of network of networks under targeted attack. *Physical Review E*, *87*(5), Article 052804.

Erdös, P., & Rényi, A. (1964). On the strength of connectedness of a random graph. *Acta Mathematica Hungaric*, *12*(1–2), 261–267.

Fan, C., Zeng, L., Sun, Y., & Liu, Y.-Y. (2020). Finding key players in complex networks through deep reinforcement learning. *Nature Machine Intelligence*, *2*, 317–324.

Fay, M. P., & Proschan, M. A. (2010). Wilcoxon-mann-whitney or t-test? On assumptions for hypothesis tests and multiple interpretations of decision rules. *Statistics Surveys*, *4*, 1.

Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep sparse rectifier neural networks. In *International conference on artificial intelligence and statistics* (pp. 315–323).

Goh, K.-I., Kahng, B., & Kim, D. (2001). Universal behavior of load distribution in scale-free networks. *Physical Review Letters*, *87*(27), Article 278701.

Grassia, M., De Domenico, M., & Mangioni, G. (2021). Machine learning dismantling and early-warning signals of disintegration in complex systems. *Nature Communications*, *12*(5190).

Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Inductive representation learning on large graphs. In *International conference on neural information processing systems* (pp. 1025–1035).

Hayashi, Y., & Uchiyama, N. (2018). Onion-like networks are both robust and resilient. *Scientific Reports*, *8*.

Holme, P., Kim, B. J., Yoon, C. N., & Han, S. K. (2002). Attack vulnerability of complex networks. *Physical Review E*, *65*(5), Article 056109.

Iyer, S., Killingback, T., Sundaram, B., & Wang, Z. (2013). Attack robustness and centrality of complex networks. *PLoS One*, *8*(4), Article e59613.

Kruskal, W. H., & Wallis, W. A. (1952). Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association*, *47*(260), 583–621.

Liu, Y.-Y., Slotine, J.-J., & Barabási, A.-L. (2012). Control centrality and hierarchical structure in complex networks. *PLoS One*, *7*(9), Article e44459.

Lou, Y., He, Y., Wang, L., & Chen, G. (2022a). Predicting network controllability robustness: A convolutional neural network approach. *IEEE Transactions on Cybernetics*, *52*(5), 4052–4063.

Lou, Y., He, Y., Wang, L., Tsang, K. F., & Chen, G. (2022b). Knowledge-based prediction of network controllability robustness. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(10), 5739–5750. http://dx.doi.org/10.1109/TNNLS.2021.3071367, (online published).

Lou, Y., Wang, L., & Chen, G. (2018). Toward stronger robustness of network controllability: A snapback network model. *IEEE Transactions on Circuits and Systems I: Regular Papers*, *65*(9), 2983–2991.

Lou, Y., Wu, R., Li, J., Wang, L., & Chen, G. (2021). A convolutional neural network approach to predicting network connectedness robustness. *IEEE Transactions on Network Science and Engineering*, *8*(4), 3209–3219.

Lou, Y., Xie, S., & Chen, G. (2020). Searching better rewiring strategies and objective functions for stronger controllability robustness. *IEEE Transactions on Circuits and Systems II: Express Briefs*, *68*(6), 2112–2116.

Lou, Y., Yuen, S. Y., & Chen, G. (2018). Evolving benchmark functions using Kruskal-Wallis test. In *Genetic and evolutionary computation conference (GECCO)* (pp. 1337–1341).

Louzada, V. H., Daolio, F., Herrmann, H. J., & Tomassini, M. (2013). Smart rewiring for network robustness. *Journal of Complex Networks*, *1*(2), 150–159.

MacQueen, J. (1967). Classification and analysis of multivariate observations. In *Berkeley symposium on mathematical statistics and probability* (pp. 281–297).

Min, B., Do Yi, S., Lee, K.-M., & Goh, K.-I. (2014). Network robustness of multiplex networks with interlayer degree correlations. *Physical Review E*, *89*(4), Article 042811.

Newman, M. E. (2003). Mixing patterns in networks. *Physical Review E*, *67*(2), Article 026126.

Newman, M. E. (2010). *Networks: an introduction*. Oxford University Press.

Newman, M. E., & Watts, D. J. (1999). Renormalization group analysis of the small-world network model. *Physics Letters. A*, *263*(4–6), 341–346.

Niepert, M., Ahmed, M., & Kutzkov, K. (2016). Learning convolutional neural networks for graphs. In *International conference on machine learning (ICML)* (pp. 2014–2023).

Perra, N., & Fortunato, S. (2008). Spectral centrality measures in complex networks. *Physical Review E*, *78*(3), Article 036107.

Pu, C.-L., Pei, W.-J., & Michaelson, A. (2012). Robustness analysis of network controllability. *Physica A: Statistical Mechanics and its Applications*, *391*(18), 4420–4425.

Qi, M., Deng, Y., Deng, H., & Wu, J. (2018). Optimal disintegration strategy in multiplex networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, *28*(12), Article 121104.

Qiu, T., Liu, J., Si, W., & Wu, D. O. (2019). Robustness optimization scheme with multi-population co-evolution for scale-free wireless sensor networks. *IEEE/ACM Transactions on Networking*, *27*(3), 1028–1042.

Rossi, R., & Ahmed, N. (2015). The network data repository with interactive graph analytics and visualization. In *Twenty-ninth AAAI conference on artificial intelligence* (pp. 4292–4293). AAAI Press.

Rossi, R. A., & Ahmed, N. K. (2016). An interactive data repository with visual analytics. *SIGKDD Explorations*, *17*(2), 37–41.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, *61*, 85–117.

Schneider, C. M., Moreira, A. A., Andrade, J. S., Havlin, S., & Herrmann, H. J. (2011). Mitigation of malicious attacks on networks. *Proceedings of the National Academy of Sciences*, *108*(10), 3838–3841.

Schneider, C. M., Yazdani, N., Araújo, N. A., Havlin, S., & Herrmann, H. J. (2013). Towards designing robust coupled networks. *Scientific Reports*, *3*(1), 1–7.

Shargel, B., Sayama, H., Epstein, I. R., & Bar-Yam, Y. (2003). Optimization of robustness and connectivity in complex networks. *Physical Review Letters*, *90*(6), Article 068701.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint: 1409.1556.

Sugiyama, M., Ghisu, M. E., Llinares-López, F., & Borgwardt, K. (2017). Graphkernels: R and python packages for graph comparison. *Bioinformatics*, *34*(3), 530–532.

Tanizawa, T., Havlin, S., & Stanley, H. E. (2012). Robustness of onionlike correlated networks against targeted attacks. *Physical Review E*, *85*(4), Article 046109.

Wang, S., & Liu, J. (2019). Designing comprehensively robust networks against intentional attacks and cascading failures. *Information Sciences*, *478*, 125–140.

Wang, S., Liu, J., & Jin, Y. (2020). Surrogate-assisted robust optimization of large-scale networks based on graph embedding. *IEEE Transactions on Evolutionary Computation*, *24*(4), 735–749.

Wang, S., Liu, J., & Jin, Y. (2021). A computationally efficient evolutionary algorithm for multiobjective network robustness optimization. *IEEE Transactions on Evolutionary Computation*, *25*(3), 419–432.

Wu, Z.-X., & Holme, P. (2011). Onion structure and network robustness. *Physical Review E*, *84*(2), Article 026106.

Wu, C., Lou, Y., Wu, R., Liu, W., & Li, J. (2022). CNN-based prediction of network robustness with missing edges. In *2022 international joint conference on neural networks (IJCNN)*. IEEE.

Xiang, L., Chen, F., Ren, W., & Chen, G. (2019). Advances in network controllability. *IEEE Circuits and Systems Magazine*, *19*(2), 8–32.

Xiao, Y.-D., Lao, S.-Y., Hou, L.-L., & Bai, L. (2014). Optimization of robustness of network controllability against malicious attacks. *Chinese Physics B*, *23*(11), Article 118902.

Yan, G., Vértes, P. E., Towlson, E. K., Chew, Y. L., Walker, D. S., Schafer, W. R., et al. (2017). Network control principles predict neuron function in the caenorhabditis elegans connectome. *Nature*, *550*(7677), 519.

Yang, C., Mao, J., Qian, X., & Wei, P. (2018). Designing robust air transportation networks via minimizing total effective resistance. *IEEE Transactions on Intelligent Transportation Systems*, *20*(6), 2353–2366.
Zeng, A., & Liu, W. (2012). Enhancing network robustness against malicious attacks. *Physical Review E*, *85*(6), Article 066130.
Zhang, R. (2019). Making convolutional networks shift-invariant again. In *International conference on machine learning* (pp. 7324–7334). PMLR.
Zhang, X., Wu, J., Wang, H., Xiong, J., & Yang, K. (2016). Optimization of disintegration strategy for multi-edges complex networks. In *2016 IEEE congress on evolutionary computation (CEC)* (pp. 522–528). IEEE.