Using Machine Learning to Quantify the Robustness of Network Controllability

Ashish Dhiman, Peng Sun(✉), and Robert Kooij

Delft University of Technology, Delft, The Netherlands
ashish06.dhiman@gmail.com, P.Sun-1@tudelft.nl

Abstract. This paper presents machine learning based approximations for the minimum number of driver nodes needed for structural controllability of networks under link-based random and targeted attacks. We compare our approximations with existing analytical approximations and show that our machine learning based approximations significantly outperform the existing closed-form analytical approximations in case of both synthetic and real-world networks. Apart from targeted attacks based upon the removal of so-called critical links, we also propose analytical approximations for out-in degree-based attacks.

Keywords: Network controllability · Network robustness · Driver nodes · Machine learning

1 Introduction

In the modern world, we see networks everywhere such as the Internet, transportation networks, and communication networks [18]. It is important that these networks perform their desired functions properly. Naturally, we need to control these networks to ensure their proper functioning and maintenance. Network science offers a way to study and analyze these networks using graph theory. The entities in a network are represented by the nodes and interconnections between the nodes are represented by links. For example, in an air-transportation network, the nodes represent different airports and the links represent the flight paths that connect these airports. Network controllability is the ability to drive a system from an initial state to any other state in a finite time by application of external inputs on certain nodes [3]. For directed networks, Liu *et al.* [2] showed that the minimum number of nodes required to control a network can be identified through the maximum matching of the network. However, Cowan *et al.* [5] pointed out that the results of Liu *et al.* [2] are based on the assumption of no self-links. In other words, a state of a node can only be changed through interacting with its adjacent nodes. Recently, Sun *et al.* [1] derived closed-form analytical approximations for the minimum number of driver nodes as a function of the fraction of removed links for both random and targeted attacks. However, the approximations sometimes do not fit well with the simulations, especially

É. Renault et al. (Eds.): MLN 2020, LNCS 12629, pp. 19–39, 2021.
https://doi.org/10.1007/978-3-030-70866-5_2

when the fraction of removed links is not small. Figure 1 shows the performance of Sun's approximation as compared to simulation for a Erdős-Rényi network under targeted attack. We will discuss the analytical approximations by Sun *et al.* [1] for both random and targeted attacks in Sect. 3 of this paper.

The objective of this work is to improve the analytical approximations for both random and targeted attacks using machine learning methods. We will compare our machine learning based approximations with the existing analytical approximations and simulations. Furthermore, we will also derive an analytical approximation for out-in degree-based attacks and evaluate its performance on both synthetic and real-world networks.

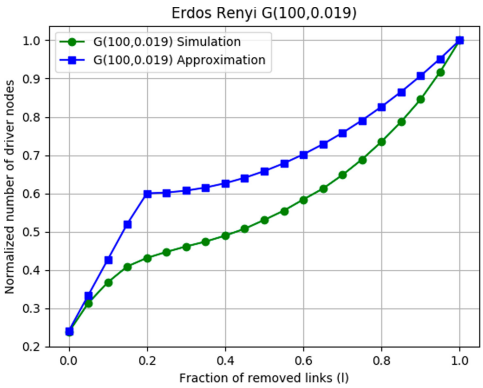

Fig. 1. Performance comparison of Sun's approximation for the normalized minimum number of driver nodes as a function of the fraction of removed links in a Erdős-Rényi network under targeted attack.

In the remainder of this paper, in Sect. 2 we describe the concept of network robustness. In Sect. 3, network controllability is discussed. In Sect. 4, we discuss the closed-form analytical approximations for the minimum number of driver nodes given by Sun *et al.* [1]. Machine learning methods and information related to training and testing data are discussed in Sect. 5. Machine learning based approximations for both random and targeted attacks are presented in Sect. 6. An analytical approximation for out-in degree-based attacks is also derived in this section. Additionally, we also analyze and compare our machine learning based approximations with Sun's approximations and simulations. Finally, in Sect. 7 we conclude this paper.

2 Network Robustness

Network robustness is the ability of a network to deal with failures and errors. In real-world networks, we encounter various failures such as power transmission line failures in an electrical network and network disruption due to natural

disasters. It is important to make networks robust to deal with such failures. A generic quantitative definition of network robustness does not exist but there are various metrics to assess network robustness depending on the type of network and its purpose. In this work, we assess network robustness in terms of controllability. Network robustness under perturbations has been studied extensively. Socievole *et al.* [6] studied network robustness in case of epidemic spreads. They investigated Susceptible-Infected-Susceptible (SIS) spreads with N-Intertwined Mean-Field Approximation (NIMFA) epidemic threshold as the robustness metric. Trajanovski *et al.* [7] considered node removals in both random and targeted attacks to study network robustness. They used two metrics to evaluate the network robustness, the size of the giant component and efficiency. Wang *et al.* [8] considered effective graph resistance as the robustness metric to investigate network robustness in case of both synthetic and real-world networks. Koç *et al.* [9] studied the robustness of networks in terms of cascading failures that lead to blackouts in electrical power grids.

Real-world networks are often challenged by perturbations in the form of random and targeted attacks [29]. In this work, we simulate these attacks by removing links. We do not consider node removals. Random attacks are the unintentional failures such as disruption of networks due to natural disasters and failures due to exhausted mechanical parts [24]. Targeted attacks are carried out by people with malicious intent to maximize the damage [25–28]. In targeted attacks, it is assumed that the attacker has the information related to network topology, functions and vulnerabilities.

3 Network Controllability

Network controllability is the ability to drive a system from an initial state to any other state in a finite time by application of external inputs on certain nodes [3]. It is classified as state controllability and structural controllability.

3.1 State Controllability

State controllability, also known as complete controllability, was introduced by Kalman in the 1960s [3]. Even though non-linear processes govern most of the real-world systems, a linearized counterpart offers a way to study the controllability of non-linear systems [2]. In this work, we consider directed networks with linear time-invariant (LTI) dynamics which are described by:

$$\frac{dx(t)}{dt} = Ax(t) + Bu(t), \tag{1}$$

where $x(t) = [x_1(t), x_2(t), ..., x_N(t)]^T$ indicates the state vector of the system at time t. $x_i(t)$ represents the state that could be the amount of traffic that passes through node i in a communication network. The $N \times N$ adjacency matrix A represents the interconnections of a network [19]. The input $N \times M$ $(M \leq N)$ matrix B represents the nodes that are directly controlled.

$u(t) = [u_1(t), u_2(t), ..., u_M(t)]^T$ is the input vector. According to Kalman's controllability condition, the system described in Eq. (1) is said to be controllable if the controllability matrix $C = (B, AB, A^2B, ..., A^{N-1}B)$ has full rank. In other words, $Rank(C) = N$. However, Kalman's rank condition for network controllability has some limitations. It is computationally expensive to check Kalman's rank condition for larger networks that consist of thousands of nodes. The rank condition also requires exact weights of the parameters of A and B but in reality, often the link weights are not known. To account for such limitations, the concept of structural controllability was introduced.

3.2 Structural Controllability

Structural controllability was introduced by Lin in 1974 [10]. The system described in Eq. (1) is said to be structurally controllable if we can fix some weights to the non-zero parameters in A and B so that the system becomes controllable in Kalman's controllability condition. To ensure full rank condition, we have to appropriately choose B which consists of a minimum number of driver nodes. A structurally controllable system is also controllable for different possible parametric realizations except for some pathological cases [2]. One of the advantages of studying structural controllability is that the controllability of a network can still be determined even if we lack information about some or all the link weights.

Liu *et al.* [2] developed the minimum input theory to achieve structural controllability of directed networks. According to Liu *et al.* [2], the minimum number of driver nodes to which external inputs needs to be applied to achieve structural controllability is determined by the maximum matching of the network. They found that the minimum number of driver nodes required to fully control a network depends on the degree distribution. Furthermore, they observed that sparse and homogeneous networks are difficult to control as compared to dense and heterogeneous networks. Liu's work is based on the assumption that there are no self-links in the networks. In this work, we also follow this assumption. Next, we discuss the concept of maximum matching to determine the minimum number of driver nodes required to fully control a network. Next, we find the matching links i.e. the links that do not have common start or end nodes. The nodes at which these links terminate are the matched nodes. The remaining nodes are the unmatched nodes or driver nodes. We will apply external input to these unmatched or driver nodes to fully control the network. In a network, a matching of maximum size is known as maximum matching. There could be multiple maximum matchings in a network but the number of driver nodes remains the same [1]. The Hopcroft-Karp algorithm [11] provides a method to find the maximum matching of a network from its bipartite equivalent.

Now we discuss the robustness of network controllability under perturbations. On removal of a critical link, the number of driver nodes increases by one [2]. In other words, we need more driver nodes to fully control the network when a critical link is removed. It means that there is a decrease in network controllability or the network becomes less robust. Nie *et al.* [12] studied the

robustness of network controllability of Erdős-Rényi and Barabási-Albert networks and observed that a Barabási-Albert network with a modest power-law exponent is more robust than an Erdős-Rényi network with a modest average degree. Pu *et al.* [13] studied network controllability and found that degree based attacks are more efficient than random attacks in affecting network controllability. Sun *et al.* [1] quantified the robustness of network controllability for two types of attacks based on the removal of links, random attacks and targeted attacks. They derived closed-form analytical approximations for the minimum number of driver nodes for both random and targeted attacks. While the results of Sun *et al.* [1] fit well with the simulations for small fractions of remove links, there is still room for improvement. In the next section, we will use machine learning to construct more accurate approximations and analyze their performance.

4 Analytical Approximations

The analytical approximations for random and targeted link removals by Sun *et al.* [1] are based on the concept of critical links. If the number of driver nodes required to control a network increases when removing a specific link, then that link is called a critical link. A link that does not belong to any maximum matching is dubbed a redundant link. A link that is neither critical nor redundant is an ordinary link. The initial number of driver nodes N_{DO} i.e. the number of driver nodes before any attack, is calculated using the Hopcroft-Karp algorithm [11]. To find the number of critical links, each link in a network is removed one by one and the Hopcroft-Karp algorithm [11] is applied simultaneously. If the current number of driver nodes N_D exceeds the initial number of driver nodes N_{DO}, then the removed link is a critical link. In a network with N nodes and L links, the Hopcroft-Karp algorithm [11] is applied L times to identify all the critical links.

4.1 Number of Driver Nodes Under Random Attacks

According to Sun *et al.* [1], for random attacks, the normalized minimum number of driver nodes is expressed as,

$$n_{D,rand} = \begin{cases} \frac{N_{DO}+lL_C}{N}, & l \leq l_C \\ al^2 + bl + c, & l \geq l_C \end{cases} \tag{2}$$

where $n_{D,rand}$ represents the normalized value of the minimum number of driver nodes required to fully control a network, L_C represents the number of critical links, l represents the fraction of removed links and $l_C = \frac{L_C}{L}$ represents the fraction of critical links. The values of a, b and c are derived from the boundary conditions described in [1] such that $a = \frac{N-N_{DO}-L_C}{N(l_C-1)^2}$, $b = \frac{L_C}{N} - 2al_C$ and $c = 1 - \frac{L_C}{N} + a(2l_C - 1)$.

4.2 Number of Driver Nodes Under Targeted Attacks

In targeted attacks, first we randomly remove all the critical links and then the remaining links. Sun *et al.* [1] derived the following analytical approximation for targeted attacks.

$$n_{D,crit} = \begin{cases} \frac{N_{DO}+lL}{N}, & l \leq l_C \\ dl^2 + el + f, & l \geq l_C \end{cases} \tag{3}$$

where d, e and f are derived from the boundary conditions described in [1] such that $d = \frac{N-N_{DO}-l_C L}{N(l_C-1)^2}$, $e = -2dl_C$ and $f = 1 + d(2l_C - 1)$.

5 Machine Learning

Machine learning is a technique to predict the outcome of a certain event by learning from data. The data could already be available from experiments, data centers or it can be generated through proper simulations. There are numerous applications of machine learning such as predicting customer's buying habits based on historical data in e-Commerce, weather forecasts and Virtual Personal Assistants such as Siri and Alexa. In broader terms, machine learning is classified as supervised learning, unsupervised learning and reinforcement learning. Furthermore, supervised machine learning is divided into classification and regression problems. In this work, we use various supervised learning methods for regression problems to predict the number of driver nodes under various attacks. Specifically, we use Linear Regression, Random Forest and Artificial Neural Networks. Recently, Lou *et al.* [30] also investigated the use of neural networks for network controllability. However, they used another type of neural networks, Convolution Neural Networks.

To develop our machine learning models, various hyper-parameters are used. Table 1 and Table 2 shows the number of hidden layers and other hyper-parameters that are used to develop our ANN models. For our linear regression model, we use the least-squares to minimize the errors. Additionally, we also use k-fold cross-validation with $k = 10$ to check for over-fitting. In our Random-Forest model, we select the number of trees as 50. Moreover, we also use feature importance scores to determine the features that contribute more to the output. A detailed explanation of the choice of hyper-parameters is presented in the master thesis report [4].

Table 1. Selection of ANN size for different networks under targeted, random and out-in degree-based attacks.

Attack	**Number of hidden layers**		
	Real-world	**Erdős-Rényi**	**Barabási-Albert**
Targeted critical link attack	512/512/512	128	512/512/512
Random attack	512/512/512	128/512/512/512	128/512/512/512
Out-in degree based attack	512/512/512	128	512/512/512

Table 2. ANN hyper-parameters selection.

Hyper-parameters	Activation function	Loss function	Dropout rate	Early stopping	Patience	Epochs	Batch size
Selection	ReLU	MSE	0.2	Yes	50	300	32

Table 3. Properties of 10 real-world networks used for testing our models.

Network	N	L	L_C	N_{DO}
Colt	153	177	38	81
Surfnet	50	68	23	15
EliBackbone	20	30	12	5
Garr200912	54	68	9	30
GtsPoland	33	37	12	14
Ibm	18	24	6	6
Arpanet19706	9	10	6	2
GtsHungary	30	31	8	18
BellCanada	48	64	17	16
Uninet	69	96	19	4

5.1 Dataset for Real-World Networks

Now we discuss the real-world dataset that we consider to construct our models. For synthetic networks, we generate data through simulations. We use the dataset available at The Internet Topology Zoo [14] for real-world networks. It is a collection of a publicly accessible dataset provided by different network operators. As the networks evolve and change, the dataset is updated and in this sense, it is not fixed. Network operators provide maps of their networks and this dataset is interpreted from those maps. However, there are various ambiguities in the dataset as the interpretations are not accurate for some networks. The dataset is available in Graph Markup Language (GML) [15] and GraphML [16] formats. In this work, we consider the dataset that is available in GraphML format as it is easy to parse using python's NetworkX library [17]. We pre-process the data to remove any disconnected networks and multigraphs. After pre-processing of the dataset, we have 232 networks out of which we use 192 networks for training and the remaining 40 networks for testing. The networks in the dataset are not directed, however, we use the information available in two attributes of the GraphML format, edge source and target, to make these networks directed.

The networks in the dataset have small average degrees. The smallest network is the Arpanet196912 network with 4 nodes and 4 links. Cogentco network is the largest network with 197 nodes and 243 links. Additionally, there are some networks that have zero critical links. We conclude that the networks in this dataset vary a lot and machine learning models might have difficulties in learning from such a varying dataset. Table 3 lists the properties of some of the real-world networks we use for testing.

5.2 Datasets for Synthetic Networks

We generate data for synthetic networks using simulations. We consider two types of synthetic networks, Erdős-Rényi and Barabási-Albert networks. These networks come under the class of random graphs [20]. In Erdős-Rényi (ER) random graphs $G(N, p)$ [21], N denotes the number of nodes and p denotes the probability of an outbound link from a node to another node. For Erdős-Rényi networks, we generate networks with different values of N and p. For each such network, we generate 100 corresponding networks and determine the average values of network characteristics such as the average degree, the average number of links, the number of critical links and graph metrics such as diameter and clustering coefficient.

In the Barabási-Albert (BA) scale-free model $G(N, M)$ [22,23], N indicates the number of nodes and M indicates the number of links of a new node that attaches itself to the original network. To generate a BA network, we assume a complete digraph of M_O nodes where M_O equals M. Then we add new nodes one by one with a probability proportional to the number of links of the existing nodes. We generate BA networks with different values of N and M using simulations. For each BA network, we also generate 100 corresponding networks to get the average values of the network characteristics such as the average degree, the average number of links, the average number of critical links and graph metrics such as diameter and clustering coefficient. Moreover, it is to be noted that in a targeted critical link attack, first, the critical links are removed randomly and then the remaining links. For such random removal of links, we use 10,000 simulations. Furthermore, in random attacks, all the links are removed uniformly at random and we also use 10,000 simulations to get the average values of the minimum number of driver nodes.

6 Measuring the Robustness of Network Controllability Using Machine Learning

6.1 Targeted Critical Link Attack

To develop a machine learning based approximation for targeted critical link attack, we predict the difference in the normalized minimum number of driver nodes between the simulation value and the analytical approximation Eq. (3) at $l = l_C$. We use various input features such as the number of nodes N, number of links L, number of critical links L_C, clustering coefficient, average degree and diameter. We choose to estimate the difference at l_C as the original approximation fits well with the simulation for $l \ll l_C$ [1], while the difference can be significant at $l = l_C$, see also Fig. 1 , where $l_c = 0.2$. We subtract this predicted difference to get a new value n_{DX} that is closer to the simulation. We assume a linear relationship similar to the analytical approximation Eq. (3) for $l \leq l_C$. The value of the normalized minimum number of driver nodes at $l = 0$ is n_{DO} where, $n_{DO} = \frac{N_{DO}}{N}$ and at $l = l_C$, the value is assumed to be n_{DX}. From these two conditions we get,

$$n_{D,crit,ML} = n_{DO} + \frac{n_{DX} - n_{DO}}{l_C} l, \tag{4}$$

where $n_{D,crit,ML}$ gives us the new machine learning based normalized minimum number of driver nodes for $l \leq l_C$. When the fraction of removed links l is greater than or equal to the fraction of critical links l_C i.e. for $l \geq l_C$, we estimate the normalized minimum number of driver nodes using a parabolic approximation of the form,

$$n_{D,crit,ML} = d_{ML} l^2 + e_{ML} l + f_{ML}, \tag{5}$$

where d_{ML}, e_{ML} and f_{ML} are derived from the boundary conditions. For the first boundary condition, $n_{D,crit,ML}$ equals n_{DX} at $l = l_C$. When all the links are removed, we need to control all the nodes. Hence, at $l = 1$, $n_{D,crit,ML}$ equals one. Finally, for the third boundary condition, we assume the derivative of the parabola is zero at $l = l_C$. Using these boundary conditions, we get $d_{ML} = \frac{1-n_{DX}}{l_C^2 - 2l_C + 1}$, $e_{ML} = -2 d_{ML} l_C$ and $f_{ML} = 1 + d_{ML}(2l_C - 1)$. Finally, the machine learning based approximation for targeted attacks can be expressed as,

$$n_{D,crit\_ML} = \begin{cases} n_{DO} + \frac{n_{DX} - n_{DO}}{l_C} l, & l \leq l_C \\ d_{ML} l^2 + e_{ML} l + f_{ML}, & l \geq l_C \end{cases} \tag{6}$$

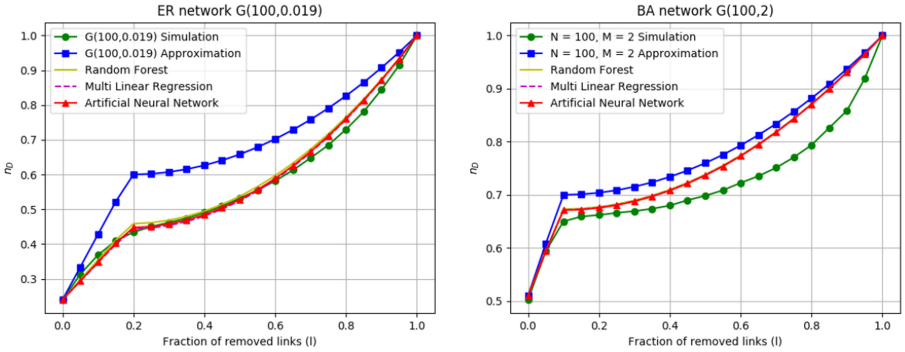

Fig. 2. Comparison of different methods to get the normalized values of minimum number of driver nodes n_D needed to control the network as a function of the fraction of removed links in synthetic networks under targeted attacks. Simulations are based on 10,000 realizations of attacks.

In Fig. 2, we compare the performance of linear regression, random forest and artificial neural network models with simulation and analytical approximation Eq. (3) for synthetic networks under targeted attacks. We notice that the machine learning based approximation fits better with the simulations than the analytical approximation Eq. (3). To further quantify the performance, we use

mean absolute errors and mean relative errors to compare the performance of different approximations. Table 4 compares the performance of ANN with the analytical approximation Eq. (3) for a few synthetic networks. We observe that the mean relative error decreases from 19.07% to 2.13% using the ANN-based approximation for ER network with $N = 100$ and $p = 0.019$. For BA network with $N = 100$ and $M = 2$, we see an improvement from 7.04% to 4.67%. Furthermore, the mean relative errors are larger for Barabási-Albert networks as compared to Erdős-Rényi networks. This is because, in BA networks, there are a few nodes with high degrees, so even after removal of some links, the minimum number of driver nodes does not change significantly and hence, the curve is less steep in BA networks as compared to ER networks as also evident from Fig. 2.

Table 4. Performance indicators for synthetic networks under targeted attacks.

Network	Mean absolute error		Mean relative error	
	Approximation	ANN	Approximation	ANN
ER (100, 0.019)	0.1000	0.0124	0.1907	0.0213
ER (200, 0.0063)	0.0663	0.0115	0.1008	0.0175
ER (400, 0.0026)	0.0472	0.0046	0.0659	0.0071
BA (50, 2)	0.0590	0.426	0.0821	0.0582
BA (100, 2)	0.051	0.0351	0.0704	0.0467

Next, we evaluate the performance of machine learning based approximation for real-world networks under targeted attacks. The model is trained on 192 real-world networks and tested on 40 networks. Figure 3 shows that machine learning based curves fit better with the simulations than the analytical approximation Eq. (3) for Colt and Surfnet network. We also compare the performance of different machine learning models based on the root mean squared errors (RMSE). The RMSE values are found to be 0.0723, 0.0550 and 0.0430 for linear regression, random forest and artificial neural network model respectively. We observe that the ANN model performs slightly better than the random forest model. The linear regression model performs the least amongst the three machine learning models. This can be explained based on the non-linear relationship between the input features and the difference that we predict.

In Table 5, we compare the performance of the ANN-based approximation and the analytical approximation Eq. (3) for 10 real-world networks. We notice that machine learning based approximation performs the best in the case of the Colt network with a mean relative error of 1.46% and the worse in Ibm network with a mean relative error of 8.3%. Furthermore, we observe that 9 out of 10 networks have mean relative errors of less than 5%. Among the 40 test networks, the machine learning based approximation performs better than the analytical approximation Eq. (3) in 30 networks. For the remaining 10 networks, the analytical approximation performs only slightly better with a difference of

less than 2%. The results of the remaining test networks are available in the master thesis report [4].

6.2 Random Attack

In this section, we develop a machine learning based approximation for the normalized minimum number of driver nodes as a function of the fraction of removed links for random attacks. Furthermore, we compare our approximation with the analytical approximation Eq. (2) and simulations. We also evaluate the performance of different machine learning algorithms. For real-world networks, the RMSE comes out to be 0.0165 for the ANN model and 0.0192 for the random forest model. Again, the ANN model performs slightly better in terms of RMSE.

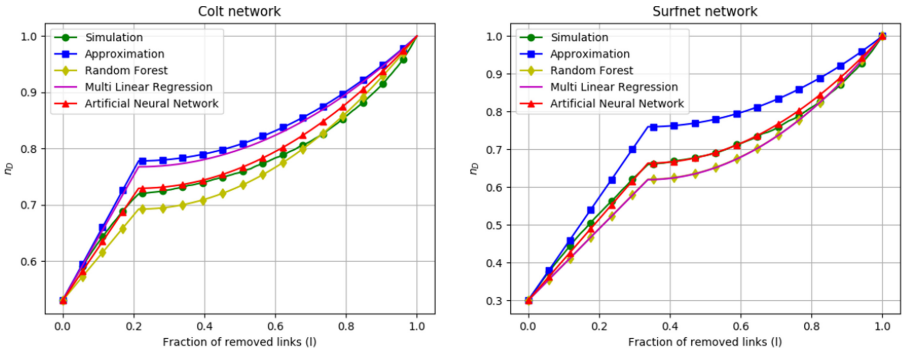

Fig. 3. Comparison of different methods to get the normalized values of minimum number of driver nodes n_D needed to control the network as a function of the fraction of removed links in real-world under targeted attacks. Simulations are based on 10,000 realizations of attacks.

Table 5. Performance indicators for real-world networks under targeted attacks.

Network	Mean absolute error		Mean relative error	
	Approximation	ANN	Approximation	ANN
Colt	0.0393	0.0116	0.0512	0.0146
Surfnet	0.0597	0.0095	0.0866	0.0151
EliBackbone	0.1468	0.0201	0.2471	0.0376
Garr200912	0.0223	0.0202	0.0277	0.0251
GtsPoland	0.0266	0.0171	0.0335	0.0235
Ibm	0.0595	0.0519	0.0956	0.0832
Arpanet19706	0.0440	0.0255	0.0588	0.0434
GtsHungary	0.0269	0.0321	0.0311	0.0373
BellCanada	0.0502	0.0135	0.0757	0.0230
Uninet	0.1195	0.0309	0.184	0.0485

Table 6. Performance indicators for synthetic networks under random attacks.

Network	Mean absolute error		Mean relative error	
	Approximation	ANN	Approximation	ANN
ER (50, 0.082)	0.0712	0.0105	0.3080	0.0675
ER (100, 0.016)	0.0085	0.0024	0.0137	0.0044
BA (50, 2)	0.035	0.0032	0.0517	0.0051
BA (100, 2)	0.032	0.0030	0.0455	0.0049

In the remainder of this section, we will only consider ANN. For random attacks, we predict the normalized minimum number of driver nodes for different values of the fraction of removed links starting with $l = 0$ to $l = 1$ in steps of 0.05. In other words, for each value of N and p in ER networks, 21 data points are generated for training. The same approach is followed for BA networks for each N and M value. The reason for such an approach is that at l_C, the difference between the approximation value and the simulation value is not significant as the approximation fits well for $l \leq l_C$ [1].

Next, we compare our machine learning based approximation for random attacks with the analytical approximation Eq. (2) and simulation. Figure 4 shows that the ANN curves fit better with the simulations for both Erdős-Rényi and Barabási-Albert networks. To quantify this improvement, Table 6 compares the performance of ANN and analytical approximation Eq. (2) based on the mean absolute errors and mean relative errors. We notice a significant improvement in mean relative error from 30.80% to 6.75% for ER network with $N = 50$ and $p = 0.082$ using ANN. Similarly, we see an improvement from 13.70% to 0.44% in the mean relative error in ER network with $N = 100$ and $p = 0.016$. Furthermore, for BA network with $N = 100$ and $M = 2$, the mean relative error improves from 4.55% to 0.49%.

Specifically for ER networks under random attacks, Liu *et al.* [2] also derived an approximation based on generating functions. According to Liu *et al.* [2], the normalized minimum number of driver nodes is given by,

$$n_D = w_1 - w_2 + k(1-l)w_1(1-w2), \tag{7}$$

Table 7. Performance indicators for all three approximations for ER networks under random attacks.

Network	Mean relative error		
	Approximation by Sun *et al.* Eq. (2)	ANN	Approximation by Liu *et al.* Eq. (7)
ER (100, 0.015)	0.0162	0.0084	0.0045
ER (100, 0.017)	0.0156	0.0097	0.0020
ER (200, 0.006)	0.0117	0.0059	0.0018

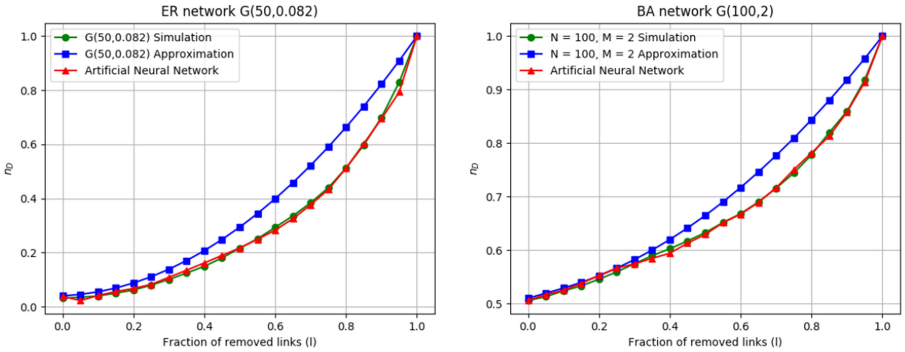

Fig. 4. Comparison of different methods to get the normalized values of minimum number of driver nodes n_D needed to control the network as a function of the fraction of removed links in synthetic networks under random attacks. Simulations are based on 10,000 realizations of attacks.

where k is the average out-degree of an ER network expressed as $k = p(N-1)$. The solution of the implicit equation $w_1 = e^{-k(1-l)e^{-k(1-l)w_1}}$ gives us the value of w_1 and w_2 is given by, $w_2 = 1 - e^{-k(1-l)w_1}$.

Now we will compare our ANN-based approximation with Sun's approximation Eq. (2), Liu's approximation Eq. (7) and simulations. From Table 7, it is evident that Liu's approximation Eq. (7) outperforms both ANN based approximation and Sun's approximation Eq. (2). In $ER\,(100, 0.015)$ network, the mean relative error using Sun's approximation Eq. (2) comes out to be 1.62%. Our ANN based approximation and Liu's approximation Eq. (7) both performs better than Sun's approximation Eq. (2) with mean relative errors of 0.84% and 0.45% respectively.

We note that Liu's approximation is based upon the use of generating functions for the degree and excess degree distribution, whose expressions are not known for targeted link removals.

For real-world networks under random attacks, we follow a different approach. Here we do not predict the normalized minimum number of driver nodes for the entire range of the fraction of removed links. This is because of the availability of a limited dataset for training and hence, the model always performs worse than the analytical approximation. Moreover, difference estimation at l_C is also not a suitable choice as the original analytical approximation is already good for $l \leq l_C$ [1]. For larger values of the fraction of removed links, the difference in n_D values between the approximation and simulation is significant. So, we choose a point $l = 0.4$ to predict the difference and subtract it from the approximation value to get a new value n_{DX}. Let the value at $l = 0.4$ be l_X. At $l = 0$, the normalized minimum number of driver nodes equals n_{DO} and at $l = 0.4$, n_D equals n_{DX}. From these two points we get,

$$n_{D,rand,ML} = n_{DO} + \frac{n_{DX} - n_{DO}}{l_X} l, \tag{8}$$

where, $n_{D,rand,ML}$ gives the normalized minimum number of driver nodes as a function of the fraction of removed links for $l \leq l_X$. For l values greater than or equal to l_X, we calculate the normalized minimum number of driver nodes using a parabolic approximation,

$$n_{D,rand,ML} = a_{ML}l^2 + b_{ML}l + c_{ML}, \tag{9}$$

where we derive the values of a_{ML}, b_{ML} and c_{ML} from the boundary conditions. At $l = l_X$, the value and derivative of Eq. (9) equals that of Eq. (8). Hence, we get $a_{ML}l_X^2 + b_{ML}l_X + c_{ML} = n_{DX}$ and $2a_{ML}l_X + b_{ML} = \frac{n_{DX}-n_{DO}}{l_X}$. At $l = 1$ i.e. when all the links are removed, we need to control all the nodes. Hence, n_D equals one and we get, $a_{ML} + b_{ML} + c_{ML} = 1$. Using these boundary conditions we get, $a_{ML} = \frac{n_{DO}-1+\frac{n_{DX}-n_{DO}}{l_X}}{-l_X^2+2l_X-1}$, $b_{ML} = \frac{n_{DX}-n_{DO}}{l_X} - 2a_{ML}l_X$ and $c_{ML} = 1 + a_{ML}(2l_X - 1) - \frac{n_{DX}-n_{DO}}{l_X}$. Finally, we express machine learning based normalized minimum number of driver nodes for real-world networks under random attacks as,

$$n_{D,rand,ML} = \begin{cases} n_{DO} + \frac{n_{DX}-n_{DO}}{l_X}l, & l \leq l_X \\ a_{ML}l^2 + b_{ML}l + c_{ML}, & l \geq l_X \end{cases} \tag{10}$$

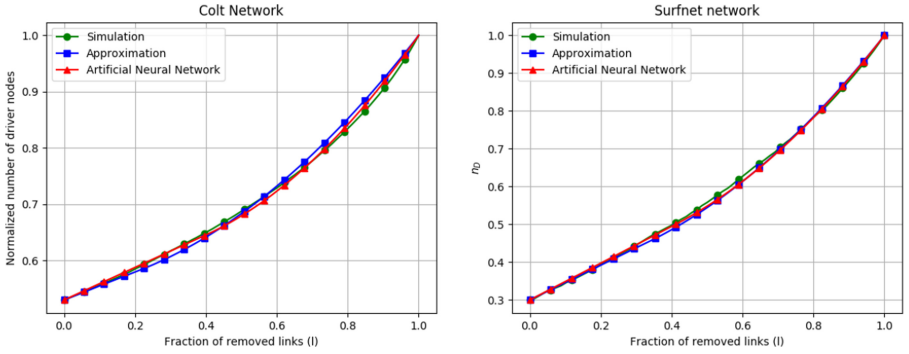

Fig. 5. Comparison of different methods to get the normalized values of minimum number of driver nodes n_D needed to control the network as a function of the fraction of removed links in real-world networks under random attacks. Simulations are based on 10,000 realizations of attacks.

Figure 5 compares our ANN-based approximation and Sun's approximation Eq. (2) with simulations for two real-world networks. We observe that ANN-based approximation fits better with the simulations. To analyze this comparison, Table 8 quantifies the performance using mean absolute and mean relative errors for 10 considered real-world networks. It can be noticed that our ANN-based approximation performs the best in the Colt network with a mean relative

Table 8. Performance indicators for real-world networks under random attacks.

Network	Mean absolute error		Mean relative error	
	Approximation	ANN	Approximation	ANN
Colt	0.0079	0.0043	0.0106	0.0058
Surfnet	0.0072	0.0052	0.0128	0.0090
EliBackbone	0.0256	0.0160	0.0454	0.0274
Garr200912	0.0121	0.0094	0.0156	0.0130
GtsPoland	0.0081	0.0046	0.0127	0.0068
Ibm	0.0072	0.0086	0.012	0.015
Arpanet19706	0.0046	0.0062	0.0073	0.0123
GtsHungary	0.0082	0.0072	0.0098	0.0088
BellCanada	0.0105	0.0071	0.0197	0.0122
Uninet	0.0207	0.0166	0.0338	0.0275

error of 0.58% and the least in the Uninet network with a mean relative error of 2.75%. Moreover, the ANN-based model does not always perform better than the analytical approximation. For example, in Ibm and Arpanet19706, the mean relative errors using ANN-based model are larger than the analytical approximation based mean relative errors. This can be explained based on the availability of a limited amount of training dataset for real-world networks. Among the 40 test real-world networks, the machine learning based approximation performs better than the analytical approximation in 28 networks. The results of the remaining networks are presented in the master thesis report [4].

6.3 Out-In Degree-Based Attack

In this section, we will derive an analytical approximation for the normalized minimum number of driver nodes n_D as a function of the fraction of removed links l for out-in degree-based attacks. Out-in degree of a link is defined as the sum of the out-degree of a source node and the in-degree of a target node. First, we compare different out-in based-attack strategies to select the most efficient one. In the first strategy, we remove links based on the increasing order of out-in degrees, second, if the out-in degrees are the same then links are removed based on the increasing order of out-degrees and finally, in the third strategy, we remove the links based on the decreasing order of out-in degrees. Based on simulations, we found that the first two strategies overlap and are the most efficient ones. So, for the remainder of this section, we will use the first strategy in which we remove links based on the increasing order of out-in degrees. It is to be noted that after removing a link, we re-calculate the out-in degrees in order to determine the next link to be removed.

Case 1: $l \leq l_C$ Similar to [1], when the fraction of removed links is less than or equal to the fraction of critical links, we assume a linear relationship between the

minimum number of driver nodes and the fraction of removed links such that,

$$n_{D,out\_in} = \frac{N_{DO} + lL}{N}. \tag{11}$$

Case 2: $l \geq l_C$ When the fraction of removed links is greater than or equal to the fraction of critical links, we approximate the minimum number of driver nodes using a quadratic equation,

$$f(l) = n_D = gl^2 + hl + i, \tag{12}$$

where g, h and i can be derived from the boundary conditions. For the first boundary condition we assume , at $l = l_C$, n_D equals $\frac{N_{DO}+l_C L}{N}$. Second, at $l = 1$, n_D equals one. Third, we assume that the derivative equals zero at $l = 1$. Using these boundary conditions we get, $g = \frac{x-1}{l_C^2 - 2l_C + 1}$, $h = -2g$ and $i = 1 - g - h$ where $x = \frac{N_{DO}+l_C L}{N}$. Finally, for out-in degree-based attacks we can write,

$$n_{D,out\_in} = \begin{cases} \frac{N_{DO}+lL}{N}, & l \leq l_C \\ gl^2 + hl + i, & l \geq l_C \end{cases} \tag{13}$$

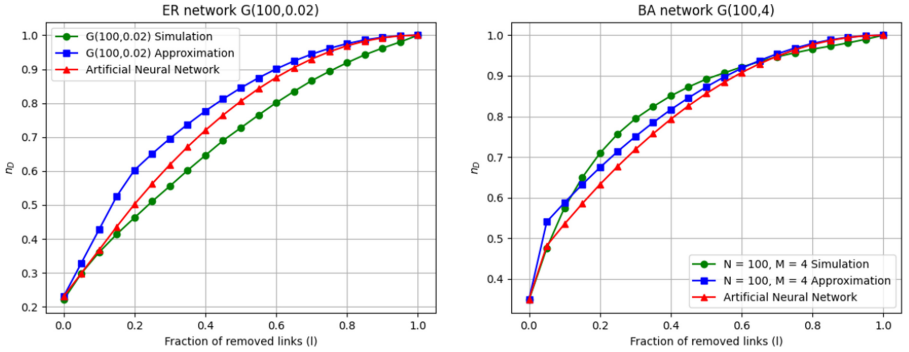

Fig. 6. Performance comparison of the machine learning based approximation Eq. (16) with the analytical approximation Eq. (13) to get the normalized values of minimum number of driver nodes n_D needed to control the networks as a function of the fraction of removed links in synthetic networks under out-in degree-based attacks.

Figure 6 shows the performance of our analytical approximation Eq. (13) for Erdős-Rényi and Barabási-Albert networks. We notice that the analytical approximation fits better with the simulations for Barabási-Albert networks. The same is also evident from Table 9 in which we show the performance of some synthetic networks. We notice that the mean relative errors are less than 3% for BA networks and greater than 10% for ER networks. We also analyze the performance of our approximation in real-world networks. Figure 7 shows

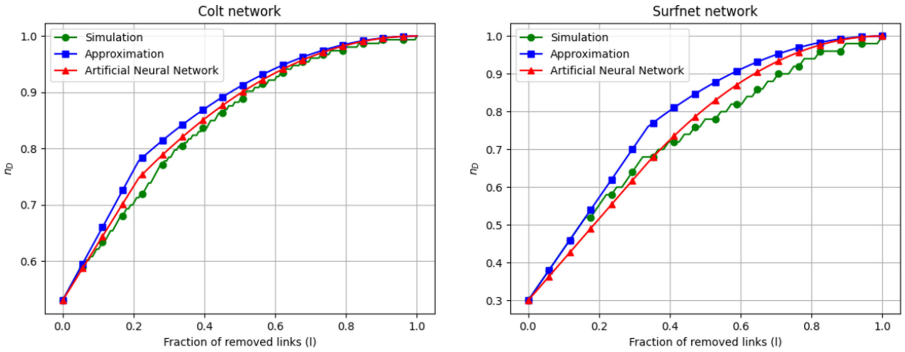

Fig. 7. Performance comparison of the machine learning based approximation Eq. (16) with the analytical approximation Eq. (13) to get the normalized values of minimum number of driver nodes n_D needed to control the networks as a function of the fraction of removed links in real-world networks under out-in degree-based attacks.

the performance of our approximation for the Colt and Surfnet networks. It can be observed that the approximation fits fairly well with the simulations. Furthermore, we analyze the performance of 10 considered real-world networks in Table 10. We notice that the mean relative errors are less than 10% in 8 out of 10 real-world networks. Moreover, the approximation performs the best in the GtsHungary network and the least in the Uninet network with mean relative errors of 1.53% and 13.61% respectively.

Next, we use ANN to further improve the performance of the analytical approximation Eq. (13). We will use ANN to predict the difference in the values of the normalized minimum number of driver nodes between the approximation value and the simulation value at l_C. We will then subtract this difference from the approximation value to get a new value n_{DX} that is closer to the simulation. At $l = 0$, the minimum number of driver nodes can be found from Eq. (13) and at $l = l_C$, the value is n_{DX}. From these two points, we get,

$$n_{D,out\_in,ML} = n_{DO} + \frac{n_{DX} - n_{DO}}{l_C} l, \tag{14}$$

where $n_{D,out\_in,ML}$ gives us the machine learning based normalized minimum number of driver nodes for $l \leq l_C$. For $l \geq l_C$, we assume a quadratic relationship for the normalized minimum number of driver nodes such that,

$$f_{ML}(l) = n_{D,out\_in,ML} = g_{ML} l^2 + h_{ML} l + i_{ML}, \tag{15}$$

To get the values of g_{ML}, h_{ML} and i_{ML}, we again use three boundary conditions. n_D equals n_{DX} at $l = l_C$. At $l = 1$, n_D equals one. The derivative $f'_{ML}(1)$ is assumed to be equal to zero at $l = 1$. Using these boundary conditions we get, $g_{ML} = \frac{n_{DX}-1}{l_C^2 - 2l_C + 1}$, $h_{ML} = -2g_{ML}$ and $i_{ML} = 1 - g_{ML} - h_{ML}$. Hence, the machine learning based approximation for the minimum number of driver nodes can be expressed as,

Table 9. Performance indicators for synthetic networks under out-in degree-based attacks.

Network	Mean absolute error		Mean relative error	
	Approximation	ANN	Approximation	ANN
ER (50, 0.048)	0.0959	0.0568	0.1786	0.0924
ER (100, 0.02)	0.0828	0.0463	0.1380	0.0680
BA (50, 4)	0.0193	0.0189	0.0278	0.0266
BA (100, 4)	0.0201	0.0308	0.0276	0.0400

$$n_{D,out\_in,ML} = \begin{cases} n_{DO} + \frac{n_{DX} - n_{DO}}{l_C} l, & l \leq l_C \\ g_{ML} l^2 + h_{ML} l + i_{ML}, & l \geq l_C \end{cases} \tag{16}$$

In Fig. 6, we compare the performance of ANN-based approximation Eq. (16) with the analytical approximation Eq. (13) and simulations in case of synthetic networks. While we notice that the ANN-based approximation improves the performance in case of Erdős-Rényi networks, it does not always improve the performance of Barabási-Albert networks as the original analytical approximation Eq. (13) already fits well. In terms of mean absolute errors and mean relative errors, Table 9 compares the performance of both approximations. We observe that for $ER\,(100, 0.02)$ network, the mean relative error decreases from 13.80% to 6.80% with ANN-based approximation. We notice similar improvements for other ER networks as shown in Table 9. For BA networks, we do not always see an improvement which is also evident in $BA(100, 4)$ network in which the mean relative error increase from 2.76% to 4.0% as the original approximation already fits well with the simulations.

Table 10. Performance indicators for real-world networks under out-in degree-based attacks.

Network	Mean absolute error		Mean relative error	
	Approximation	ANN	Approximation	ANN
Colt	0.0210	0.0102	0.0267	0.0129
Surfnet	0.0469	0.0280	0.0609	0.0395
EliBackbone	0.0846	0.0373	0.1188	0.0539
Garr200912	0.0229	0.0213	0.0262	0.0242
GtsPoland	0.0256	0.0357	0.0309	0.0447
Ibm	0.0665	0.0682	0.0922	0.0951
Arpanet19706	0.0416	0.0340	0.0522	0.0519
GtsHungary	0.0140	0.0135	0.0153	0.0148
BellCanada	0.0546	0.0657	0.0742	0.0917
Uninet	0.0956	0.0586	0.1361	0.0829

Figure 7 compares the performance of ANN based approximation Eq. (16) with the analytical approximation Eq. (13) and simulations for real-world networks. The performance of all the considered 10 real-world networks is shown in Table 10. We notice that the ANN-based approximation Eq. (16) performs better than the analytical approximation Eq. (13) in 7 out of 10 considered real-world networks.

All the simulations are performed on a PC with the following specifications - 8 GB RAM and Intel Core i5 processor with 2 cores. With these specifications, for a dataset consisting of 232 networks, it costs less than 0.6 s to train the linear regression and random forest models whereas, it costs approx. 2–3 seconds to train the artificial neural network model. Once the models have been trained, after getting the average values of 10,000 simulations as inputs to the models, it costs less than 0.5 s to get the predictions.

7 Conclusion

In this work, we used various machine learning methods to quantify the minimum number of driver nodes N_D as a function of the fraction of removed links l. We studied the robustness of network controllability using machine learning based approximations on both synthetic and real-world networks under random and targeted attacks. We also derived an analytical approximation for out-in degree-based attacks. In case of targeted critical link attack, we first compared the performance of ANN, RF and LR models and conclude that the LR model performs the least due to the nonlinear relationship between the input features and the output difference. ANN model performed slightly better than the RF model. Our machine learning based approximation outperformed the analytical approximation in both synthetic and real-world networks. However, for real-world networks, our approximation performed better than the original analytical approximation in 75% of the networks. For random, attacks our approximation performed better than the analytical approximation in 70% of the real-world networks. We also compared our machine learning based approximation with Liu's approximation and Sun's approximation for ER networks under random attacks. Liu's approximation performed better than both machine learning based approximation and Sun's approximation. We also derived analytical approximation for out-in degree-based attacks. For synthetic networks, the approximation performed better in case of BA networks than ER networks. Furthermore, in 8 out of 10 considered real-world networks, the mean relative errors are less than 10%. We further improved our analytical approximation for out-in degree-based attacks using ANN and the mean relative errors reduced to less than 6% in 7 out of 10 real-world networks.

References

1. Sun, P., Kooij, R. E., He, Z., Van Mieghem, P.: Quantifying the robustness of network controllability. In 2019 4th International Conference on System Reliability and Safety (ICSRS), pp. 66–76. IEEE, November 2019

2. Liu, Y.Y., Slotine, J.J., Barabási, A.L.: Controllability of complex networks. Nature **473**(7346), 167–173 (2011)
3. Kalman, R.E.: Mathematical description of linear dynamical systems. J. SIAM Ser. A Control **1**(2), 152–192 (1963)
4. Dhiman, A.K.: Measuring the robustness of network controllability. M.Sc. Thesis, Delft University of Technology (2020)
5. Cowan, N.J., Chastain, E.J., Vilhena, D.A., Freudenberg, J.S., Bergstrom, C.T.: Nodal dynamics, not degree distributions, determine the structural controllability of complex networks. PloS ONE **7**(6), e38398 (2012)
6. Socievole, A., De Rango, F., Scoglio, C., Van Mieghem, P.: Assessing network robustness under SIS epidemics: the relationship between epidemic threshold and viral conductance. Comput. Netw. **103**, 196–206 (2016)
7. Trajanovski, S., Martín-Hernández, J., Winterbach, W., Van Mieghem, P.: Robustness envelopes of networks. J. Complex Netw. **1**(1), 44–62 (2013)
8. Wang, X., Pournaras, E., Kooij, R.E., Van Mieghem, P.: Improving robustness of complex networks via the effective graph resistance. Eur. Phys. J. B **87**(9), 221 (2014). https://doi.org/10.1140/epjb/e2014-50276-0
9. Koç, Y., Warnier, M., Van Mieghem, P., Kooij, R.E., Brazier, F.M.: The impact of the topology on cascading failures in a power grid model. Phys. A Stat. Mech. Appl. **402**, 169–179 (2014)
10. Lin, C.T.: Structural controllability. IEEE Trans. Autom. Control **19**(3), 201–208 (1974)
11. Hopcroft, J.E., Karp, R.M.: An n5/2 algorithm for maximum matchings in bipartite graphs. SIAM J. Comput. **2**(4), 225–231 (1973)
12. Nie, S., Wang, X., Zhang, H., Li, Q., Wang, B.: Robustness of controllability for networks based on edge-attack. PloS One **9**(2), e89066 (2014)
13. Pu, C.L., Pei, W.J., Michaelson, A.: Robustness analysis of network controllability. Physica A Stat. Mech. Appl. **391**(18), 4420–4425 (2012)
14. Knight, S., Nguyen, H.X., Falkner, N., Bowden, R., Roughan, M.: The internet topology zoo. IEEE J. Sel. Areas Commun. **29**(9), 1765–1775 (2011)
15. Himsolt, M.: GML: a portable graph file format, p. 35. Technical report 94030, Universitat Passau (1997)
16. Brandes, U., Eiglsperger, M., Herman, I., Himsolt, M., Marshall, M.S.: GraphML progress report structural layer proposal. In: Mutzel, P., Jünger, M., Leipert, S. (eds.) GD 2001. LNCS, vol. 2265, pp. 501–512. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45848-4_59
17. NetworkX. Network analysis in python. https://networkx.github.io/
18. Tirpak, T.M.: Telecommunication network resource management based on social network characteristics. U.S. Patent Application No. 12/463,445 (2010)
19. Harary, F.: The determinant of the adjacency matrix of a graph. SIAM Rev. **4**(3), 202–210 (1962)
20. van der Hofstad, R.: Random graphs models for complex networks, and the brain. Complex. Sci. **1**, 199–246 (2019)
21. Erdős, P., Rényi, A.: On the evolution of random graphs. Publ. Math. Inst. Hung. Acad. Sci **5**(1), 17–60 (1960)
22. Albert, R., Barabási, A.L.: Statistical mechanics of complex networks. Rev. Mod. Phys. **74**(1), 47 (2002)
23. Barabási, A.L., Ravasz, E., Vicsek, T.: Deterministic scale-free networks. Phys. A Stat. Mech. Appl. **299**(3–4), 559–564 (2001)
24. Cohen, R., Erez, K., Ben-Avraham, D., Havlin, S.: Resilience of the internet to random breakdowns. Phys. Rev. Lett. **85**(21), 4626 (2000)

25. Cetinay, H., Devriendt, K., Van Mieghem, P.: Nodal vulnerability to targeted attacks in power grids. Appl. Netw. Sci. **3**(1), 34 (2018)
26. Holme, P., Kim, B.J., Yoon, C.N., Han, S.K.: Attack vulnerability of complex networks. Phys. Rev. E **65**(5), 056109 (2002)
27. Huang, X., Gao, J., Buldyrev, S.V., Havlin, S., Stanley, H.E.: Robustness of interdependent networks under targeted attack. Phys. Rev. E **83**(6), 065101 (2011)
28. Mengiste, S.A., Aertsen, A., Kumar, A.: Effect of edge pruning on structural controllability and observability of complex networks. Sci. Rep. **5**(1), 1–14 (2015)
29. Van Mieghem, P., et al.: A framework for computing topological network robustness. Delft University of Technology, Report 20101218 (2010)
30. Lou, Y., He, Y., Wang, L., Chen, G.: Predicting network controllability robustness: a convolutional neural network approach. IEEE Trans. Cybern. **2**, 1–12 (2020)