

Research Article

A Graph Convolution Neural Network-Based Framework for Communication Network K-Terminal Reliability Estimation

Zi-xuan Zhang ^{1,2}, You-wei Xu ¹, Yi-ning Cao ², and Lin Yang ²

¹College of Command & Control Engineering, Army Engineering University of PLA, Nanjing, 210007, China

²Systems Engineering Institute, Academy of Military Science PLA China, Beijing, 100071, China

Correspondence should be addressed to Zi-xuan Zhang; ariadneeaa@163.com and Yi-ning Cao; 13910856393@163.com

Received 28 February 2022; Accepted 30 September 2022; Published 26 October 2022

Academic Editor: Shahram Babaie

Copyright © 2022 Zi-xuan Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The exact computation of network k-terminal reliability is an NP-hard problem, and many approximation methods have been proposed as alternatives, among which the neural network-based approaches are believed to be the most effective and promising. However, the existing neural network-based methods either ignore the local structures in the network topology or process the local structures as Euclidean data, while the network topology represented by the graph is in fact non-Euclidean. Seeing that the Graph Convolution Neural network (GCN) is a generalization of convolution operators onto non-Euclidean data structure, in an effort to fill in the gap, this paper proposes a GCN-based framework for the estimation of communication network reliability. First, a dataset with sufficient sample size is constructed, by calculating the k-terminal reliability via the exact contraction-deletion method for the generated network samples. Then, an estimation model based on GCN is built, where several graph convolution layers process input information and extract node-level structural features from the network topology, a concatenation layer fuses the structural features into a graph-level representation feature, and a multi-layer perceptron computes the k-terminal reliability as output. To demonstrate the practicality and rationality of our proposed model, comparative experiments are carried out on 12 datasets, the results of which show that our proposed GCN model has an average of 59.60% and 57.52% improvement over existing methods on homogeneous datasets and heterogeneous datasets, respectively.

1. Introduction

Communication networks, as a fundamental infrastructure to many critical systems such as power transmission system [1–3], emergency response system [4, 5], and aircraft navigation system [6], has become an indispensable part of our day-to-day life [7]. Disruptions in communication systems can severely undermine the performances of these critical systems, or sometimes even lead to destructive or catastrophic events. Therefore, the analysis of communication network reliability is a practical issue of great importance.

For a given system, its reliability can be loosely defined as the probability that said the system can remain operational after some of its components fail [8–10]. For network system, the notion of reliability is mainly associated with probabilistic connectedness. For an arbitrary network $G = (V, L)$ where nodes are perfectly reliable and links fail

independently, based on the number of specified nodes, there are 3 popular network reliability measures [11–13]. The first is 2-terminal reliability, it is defined as the probability that a communication path exists for a given pair of nodes, i.e., source node s and target node t . The second is all-terminal reliability, defined as the probability that all nodes in V are able to communicate with each other via a working path. The last is k-terminal reliability, defined as the probability that for any given pair of nodes in, a series of consecutive links can connect them to one another. It is easy to see that the k-terminal reliability is reduced to 2-terminal reliability when $|K| = 2$, and all-terminal reliability when $K = V$. Thus, in this paper we consider the k-terminal reliability measure seeing that it is the most general form out of three.

Many methods have been proposed to compute the exact k-terminal network reliability, including state enumeration

[14–16], minimal cut method [17, 18], sum-of-disjoint products method [19–21], and factoring theorem [22–25]. Unfortunately, the computation complexity for all these methods remains a problem. The analytic methods for calculating k -terminal network reliability are NP-hard [26–30], and the computation time grows exponentially with the increase of node or link numbers. To overcome this problem, several approximation methods have been put forward, and so far, the neural network-based approach is believed to be the most efficient and promising among them [31, 32].

In [33], He F. and Qi H. proposed to utilize Artificial Neural Network (ANN) for all-terminal network reliability estimation, where the network topology and link reliability are taken as input and the all-terminal reliability value obtained via Monte Carlo simulation is assigned as a target. Furthermore, Srivaree-Ratana C., Konak A. and Smith A. E. [34] added the reliability upper-bounds as input to the ANN model and integrated the idea of ensemble learning by training a specialized ANN for network samples with reliability values higher than 0.9. In [35], the authors categorized the network samples as homogeneous and heterogeneous depending on whether the link reliability values are identical or not, and applied the ANN model for both types of networks. To shorten the length of the input feature and reduce the parameters of the ANN model, in [36] Altiparmak F. et al. put forward a general encoding scheme based on the degree centrality of nodes for the homogeneous networks.

The basic idea of a neural network-based estimation model is that there lies a highly complex, nonlinear relationship between the network's topology and its reliability, which can be learned via the neural network. However, the network reliability is not determined by one link or one node alone, rather it is determined by the local structures formed by several nodes and links in conjunction with each other. Although in the ANN models, the adjacency matrix flattened into a long vector can solely determine the network topology, this type of node-level feature is inadequate for network-level representations because it overlooks the potential correlation between nodes and links that are close to one another. Stemming from this idea, in [37] Alex Davila-Frias and Om Prakash Yadav proposed to make use of the Convolution Neural Network (CNN) for the all-terminal reliability estimation for homogeneous networks. The network topology is processed as a $n \times n$ pixel image determined by its adjacency matrix, and several convolution layers are stacked to extract the high-level structure feature of the network.

Unfortunately, one problem still remains with the CNN-based estimation model. The network topology represented by the graph is, in essence, non-Euclidean data, whereas the CNN model is best known for its dominating performances on Euclidean data structures like text [38, 39], image [40–42], and videos [43–44]. In [37], because the network topology is encoded as the pixel image derived from the

adjacency matrix, the “local” structures are learned for nodes and links with consecutive indices, rather for nodes and links that are actually near each other in the topology. If the nodes are reindexed, the input features can change dramatically. Without an appropriate node indexing strategy in place, the high-level feature extracted by the convolution filters lacks rationality. Besides, the CNN model in [37] assumes homogeneity for the network link reliability, which is an unrealistic simplification of many real-world systems and makes it unsuitable for certain applications.

In light of this, a general framework based on Graph Convolution Network (GCN) [45–49] for the k -terminal reliability estimation of both homogeneous and heterogeneous communication networks, is proposed in this paper. In this framework, the dataset is constructed by generating network topology and link reliability vector, then computing their corresponding k -terminal reliability. Here, the exact contraction-deletion method is chosen, as opposed to the Monte Carlo simulation approach utilized in most of the previous research, to improve the precision and practicality of the estimation model. Next, a GCN-based estimation model is designed, starting with several graph convolution layers to extract the high-level structure features, followed by a concatenation layer to fuse the features into one topology representation vector, and ending with a Multilayer Perceptron (MLP) to learn the relationship between inputs and targets. Seeing that GCN can be viewed as a generalization of CNN onto non-Euclidean data structures [48], it is our belief that it would be more suited for the task of network reliability estimation, which is also corroborated by the experiment results in later sections.

The main contribution of our proposed model is three-fold:

- (1) Several network k -terminal reliability datasets are constructed, via the exact contraction-deletion method. Compared to the approximative simulation method used in previous studies, our approach can provide more precise training data to the estimation model.
- (2) Seeing that the network topology is essentially non-Euclidean data, the GCN is taken advantage of to extract the structure feature. To the best of our knowledge, our proposed model is the first to utilize GCN in a network reliability estimation problem.
- (3) Experiment results show that our proposed GCN model outperforms the existing estimation models, with an average of 59.60% improvement of Root Mean Squared Error (RMSE) on homogeneous datasets, and 57.52% on heterogeneous datasets.

The remaining part of this paper is as follows: Section 2 introduces some notations of the graph and the general pipeline of GCN; in Section 3 the proposed framework for k -terminal network reliability is described in details; Section 4 reports the experiment results of our proposed GCN model on several datasets, comparative analysis with other

neural network-based models is also undertaken to demonstrate the advantages of our proposed model; finally, Section 5 marks the end of this paper with some concluding remarks.

2. Graph Convolution Neural Network

Many real-world systems from various fields of research can be represented by graphs, including communication networks, social networks [50, 51], protein-protein interaction networks [52], and chemistry molecular graphs [53]. In other words, graphs can be viewed as a special kind of data structure that models the pairwise relationships (edges) between a set of objects (nodes).

In essence, graphs are non-Euclidean data structures. Thus, convolutional neural networks that have outstanding performances at the task of processing Euclidean data structures like images and videos, need to be generalized before they can be directly applied to graphs [48]. From this point of view, GCN can be seen as a generalization of CNN to non-Euclidean data structure, while maintaining the three advantages of CNN, namely, local connection, shared weights, and use of multiple layers. This section briefly introduces some preliminaries of GCN.

A graph is defined as $G = (V, L)$, where $V = \{v_i | i = 1: n\}$ is a set of nodes and $L = \{(v_i, v_j) | i, j = 1: n\}$ a set of links. The adjacency matrix of a graph is defined as a $n \times n$ matrix $A = (a_{ij})_{n \times n}$ where $a_{ij} = 1$ if there exists a link from node v_i to node v_j . A graph is said to be undirected if its adjacency matrix is symmetric, i.e., $a_{ij} = a_{ji}, \forall i, j = 1: n$, otherwise it is directed. Since the focus of our work is communication networks where information flows both ways, in this paper we assume the graphs are undirected.

The general pipeline of the GCN model is shown in Figure 1. Firstly, for each graph $G = (V, L)$, the input layer processes three types of information: node features embedded as p -dimensional hidden features $x_i^{d=0}$ ($i = 1: n$) and edge features embedded as q -dimensional hidden features $e_{ij}^{d=0}$ ($\forall (v_i, v_j) \in L$), as well as the graph itself.

Then, each GCN layer computes p -dimensional representation features for each node by aggregating information passed from their neighbouring nodes and edges. Let us suppose x_i^d and e_{ij}^d is the hidden feature vector associated with node v_i and edge (v_i, v_j) at GCN layer d , respectively. Node v_i 's hidden feature vector x_i^{d+1} at the next layer $d + 1$ is updated by first aggregating the hidden feature vectors of the neighbouring nodes and edges of v_i , then applying a non-linear transformation to the aggregated information and the hidden feature vector x_i^d at layer d . Similarly, the hidden feature vector e_{ij}^{d+1} of edge (v_i, v_j) at layer $d + 1$ can also be updated with the hidden features x_i^d, x_j^d and e_{ij}^d . Generally speaking, the updating of hidden features at layer $d + 1$ is

$$\begin{aligned} x_i^{d+1} &= f^x(x_i^d, g^x(\{x_j^d, e_{ij}^d: j \in N_i\})), \\ e_{ij}^{d+1} &= f^e(e_{ij}^d, g^e(x_i^d, x_j^d)), \end{aligned} \quad (1)$$

where $N_i = \{v_j: (v_i, v_j) \in L\}$ denotes the neighbourhood of node v_i , $g^x()$ and $g^e()$ are the aggregation functions, $f^x()$ and $f^e()$ are the transformation functions of node and edge,

respectively. In each GCN layer, by aggregating the neighbouring nodes and edges' information, the updated hidden feature can capture the local structures of the graph. For instance, for the central node in Figure 1 (coloured in green), at the first GCN layer it learns from its immediate neighbouring nodes and edges (coloured in orange). Then, at the second GCN layer, the central node can learn from nodes and edges 2 hops away from itself (coloured in blue), because at the first layer, the central nodes' neighbouring nodes have also learned from their respective neighbours. This way, by stacking D GCN layers, the node representation can implicitly learn from its D -hop neighbourhood.

Next, the hidden features of nodes after D GCN layers are fused into one feature vector. Here the fusion procedures differ from various types of tasks. For classification or regression tasks at the node level, the hidden feature of each node can be taken directly as the feature vector. For classification or regression tasks at the edge or graph level, hidden features of multiple nodes need to be aggregated or concatenated to form the feature vector. A such feature vector is then passed along to an MLP to compute task-dependent outputs. Finally, the output of the task-specific MLP can be fed to a loss function. Then, via back propagation, the parameters of the GCN model can be trained in an end-to-end fashion.

3. Proposed Framework

As illustrated in Figure 2, our proposed framework for communication network reliability mainly contains the following procedures:

- Step 1. Generate a set of network topologies
- Step 2. Generate a link reliability vector for each network topology
- Step 3. Compute the exact network reliability via the contraction-deletion method
- Step 4. Convert the network topologies and corresponding link reliability vector into input features, with the computed network reliability from Step 3 as a target
- Step 5. Design the architecture of the GCN model
- Step 6. Split the dataset, then train and evaluate the GCN model

In this section, the details of the proposed framework are explained in two subsections: the dataset generation process (Step 1 to Step 3), and the reliability estimation model based on GCN (Step 4 to Step 6).

3.1. Dataset Generation. To train a machine learning model that could capture the highly-complex relationship between network topology and reliability, it is of great importance to obtain a proper dataset with a sufficient sample size. To this end, we first generate a set of network topologies. For a network with node set of size $|V| = n$, an adjacency matrix $A_{n \times n}$ can represent the network topology, where $a_{ij} = 1$ indicates a link between node v_i and node v_j , and $a_{ij} = 0$ otherwise. Since it is assumed that the communication

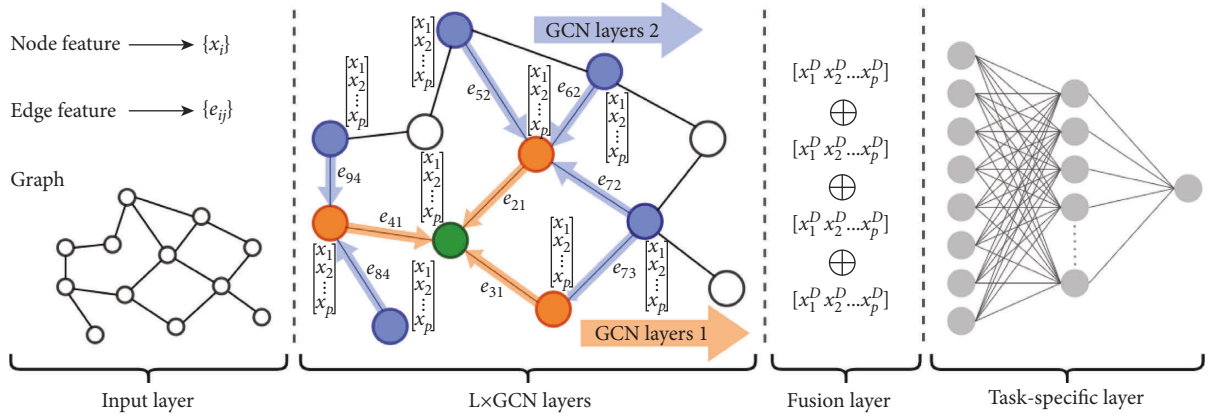


FIGURE 1: The general pipeline of Graph convolution neural network models.

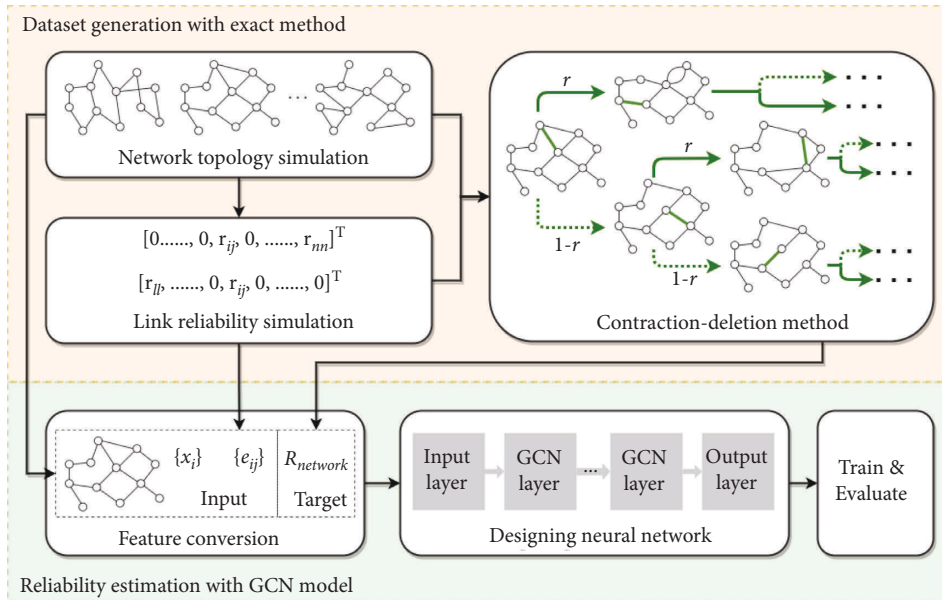


FIGURE 2: The proposed framework for communication network reliability estimation.

networks are undirected and acyclic [36], only $n(n-1)/2$ entries in the upper triangle of the adjacency matrix are randomly generated, while the rest of the entries in the adjacency matrix are determined by setting $a_{ji} = a_{ij}$ ($j > i$). What's more, a connectivity test is also needed to see if the nodes of interest in the set K are connected to one another. If so, the network topology can be added to the dataset, otherwise a new topology needs to be generated. Such procedure repeats until sufficient dataset size is satisfied.

Next, for each network topology, a $n(n-1)/2$ -dimensional link reliability vector r is generated. If a link is present in some network topology, then a reliability value in the range of $(0, 1)$ is assigned to the corresponding entry in r . Here, for homogeneous networks, the entries in r are always equal to each other; while in heterogeneous networks, the entries in r may take different values.

With network topology and link reliability vector, the k -terminal reliability can be computed. To improve the precision of the estimation model, instead of the Monte

Carlo simulation method as presented in [33–37], in this paper we use the exact method to calculate the target reliability value. One of the most widely-used exact methods for network k -terminal reliability is the contraction-deletion method [54–57]. It stemmed from a simple yet elegant observation that for an undirected network G , the reliability is given by the following equation:

$$R(G) = r_l R(G * l) + (1 - r_l) R(G \setminus l), \quad (2)$$

where l is a random edge in G , r_l is the reliability of edge l , $G * l$ and $G \setminus l$ is the new network derived from G by contracting and deleting edge l respectively. This way, the network topology can be recursively simplified until it is reduced to a singleton or a self-loop. Table 1.

The pseudocode of the contraction-deletion method [55] is presented in Table 1. The contraction-deletion method requires the network topology, link reliability vector, and a specified set of nodes to compute the k -terminal reliability.

TABLE 1: The pseudo-code of contraction-deletion method.

Algorithm	ConDel(G)
Input	$G = (V, L, K), r$
1	if $ K = 1$:
2	Return 1
3	Else:
4	Randomly select an edge $l = (v_i, v_j)$ from L .
5	/*Contraction*/
6	$L_1 \leftarrow L$
7	For $k = 1: n$:
8	if $(v_i, v_k) \in L$:
9	$L_1 \leftarrow L_1 \setminus (v_i, v_k) \cup \{(v_j, v_k)\}$
10	$V_1 \leftarrow V \setminus v_i$
11	if $v_i \in K$:
12	$K_1 \leftarrow K \setminus \{v_i\} \cup \{v_j\}$
13	$G_1 \leftarrow (V_1, L_1, K_1)$
14	/*Deletion*/
15	$L_2 \leftarrow L \setminus (v_i, v_j)$
16	$G_2 \leftarrow (V, L_2, K)$
17	if nodes in K is connected in G_2 :
18	Return $r_1 \text{ConDel}(G_1) + (1 - r_1) \text{ConDel}(G_2)$
19	Else:
20	Return $r_1 \text{ConDel}(G_1)$.

For simplicity, here we denote the graph by a triplet (V, L, K) . To begin with, the algorithm randomly selects an edge $l = (v_i, v_j)$ from the edge set L (line 4). Denote the graphs after contraction and deletion as G_1 and G_2 , respectively; (line 13 and line 16). G_1 is obtained by first replacing node v_i with node v_j for all the other edges on node v_i (line 6 to line 9), then removing node v_i from the set V (line 10). If node v_i is in K , then node v_j also needs to be in K in the contracted graph (line 11 to line 12). G_2 is obtained by directly deleting edge l (line 15). Next, a connectivity test is performed to determine whether the nodes in the K are joined by a path of edges in G_2 . If so, the network reliability is calculated as $r_1 R(G_1) + (1 - r_1) R(G_2)$ (line 17 to line 18), otherwise $r_1 R(G_1)$ (line 19 to line 20). This procedure repeats recursively until only one node in K remains, at this time the k-terminal reliability of the reduced network is naturally 1 (line 1 to line 2).

3.2. GCN Model. With the network reliability dataset generated via the contraction-deletion method, the next step in our proposed framework is to convert it into (input, target) pair for the GCN model. As mentioned in Section 2, the input to the GCN model contains three parts: node features, edge features, and graphs. In our proposed framework, the dimensions of the node feature and edge feature are set to be 1, i.e., $p = q = 1$. More specifically, the node feature is the degree of centrality of each node:

$$x_i = \sum_{j=1}^n a_{ij}, \forall i = 1: n, \quad (3)$$

the edge feature is the reliability value of each edge:

$$e_{ij} = r_{ij}, \forall i = 1: n, \quad (4)$$

and graph is the network topology. The target of the GCN model is the k-terminal network reliability R obtained via the contraction-deletion method.

The proposed GCN model for network reliability estimation is demonstrated in Figure 3. After the network information has been converted into an input form, several GCN layers update the node features by passing messages among the nodes' neighbours. To be more specific, the GCN operator in [58] is utilized in this paper, because it is easy to use and achieves state-of-the-art performances on many learning tasks. Let x_i^d be the node feature associated with node v_i at layer d , e_{ij}^d be the edge feature of (v_i, v_j) at layer d , a GCN layer updates the node and edge feature vector as follows:

$$x_i^{d+1} = \text{ReLU} \left(W_1^d \sum_{j \in N_i} e_{ji}^d \cdot x_j^d + W_2^d x_i^d \right), \quad (5)$$

$$e_{ij}^{d+1} = e_{ij}^d,$$

where $\text{ReLU}(\cdot)$ is the activation function, W_1^d and W_2^d are the weight matrices to be trained.

Several GCN layers are stacked to learn the local structures of the network. Afterwards, the final node features $\{x_i^D | i = 1: n\}$ are concatenated to form a graph-level representation vector, which is then fed to a fully-connected layer and an output layer to estimate the network reliability.

4. Experiment

To demonstrate the practicality of our proposed model, a case study of communication network reliability is carried out. Comparative analysis is also undertaken to demonstrate our proposed model's advantages to existing approaches. All experiments are run on an Intel Core i7-10510U CPU @ 1.80 GHz processor with an NVIDIA GeForce MX250 GPU.

4.1. Settings. As described in Section 3.2, several network reliability datasets are generated via the contraction-deletion method, the details of which are listed in Table 2. Based on previous studies [33–37], for networks with node set size of 10, 20, and 30, the dataset sizes are set to be 3000, 5000, and 8000, respectively. Some sample network topologies are illustrated in Figure 4, Figure 5 and Figure 6.

Based on link reliability, the datasets can be categorized into two types. For homogeneous datasets, the entries in the reliability vector takes a value in $\{0.8, 0.85, 0.9, 0.95, 0.99\}$, but are always equal to each other for the same sample. For example, the reliabilities of all links in the sample network depicted in Figure 4(a) is always 0.9, the same goes for sample networks in Figure 5(a) and Figure 6(a). While for the heterogeneous dataset, the entries of the reliability vector of one sample may take various values, as shown in Figure 4(b), Figure 5(b), and Figure 6(b). Take Figure 4(b) for instance, the reliability of link (v_8, v_2) is 0.85, whereas the reliability of link (v_2, v_6) is 0.9.

Since 2-terminal and all-terminal network reliability problems can be viewed as special cases to k-terminal reliability, our proposed framework can also be applied to

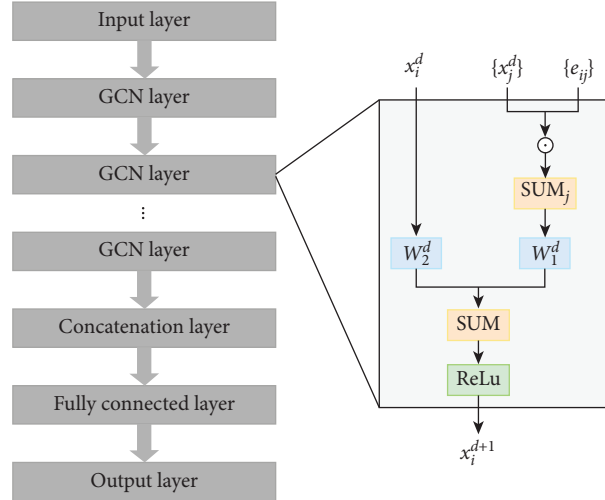
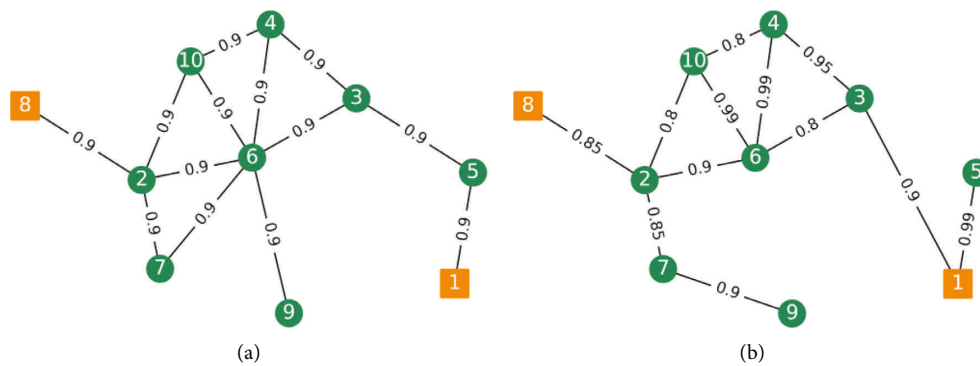


FIGURE 3: The graph convolution neural network model for network reliability estimation.

TABLE 2: Descriptions of network k-terminal reliability datasets.

Dataset no.	Network type	$ V $	Dataset size	$ K $
1	Homogeneous	10	3000	2
2				10
3				4
4				20
5				5
6				30
7	Heterogeneous	10	3000	2
8				10
9				4
10				20
11				5
12				30

FIGURE 4: Sample network topology with 10 nodes. (a) Homogeneous network with $|V| = 10$, (b) Heterogeneous network with $|V| = 10$.

them as well. Therefore, the sizes of K in the generated datasets range from 2 to $|V|$. For the dataset with $|K| < |V|$, the nodes in K are represented by orange squares in Figure 4, Figures 5, and 6, while other nodes are represented by green circles. Take Figure 4(a) for instance, in dataset 1, K is set to be $\{v_1, v_8\}$; while in dataset 2, the network reliability is computed for all 10 nodes, i.e., $K = \{v_i | i = 1: 10\}$.

To demonstrate the calculation process of the contraction-deletion method described in Section 3.1, here the sample network in Figure 4(a) is taken as an illustrative example. Because there are only two nodes v_1 and v_8 in set K , the k-terminal reliability is reduced to 2-terminal reliability, which is defined as the probability that a working communication path exists between node v_1 and node v_8 . As

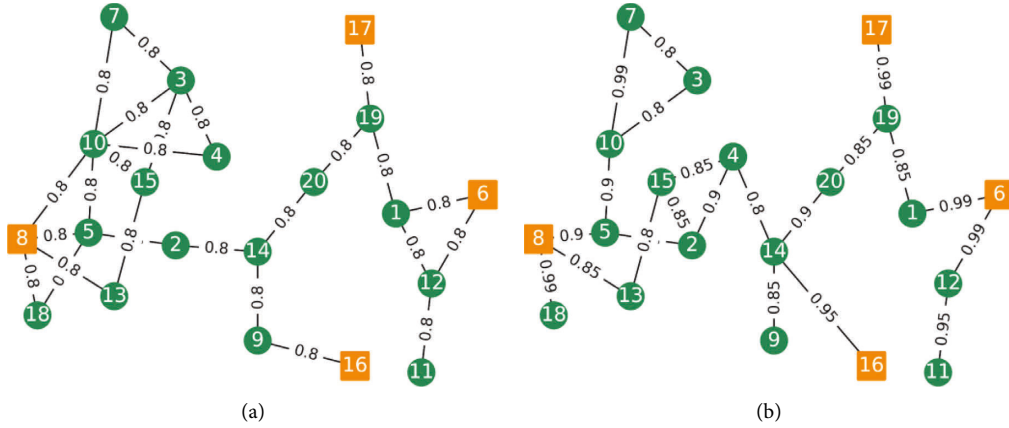


FIGURE 5: Sample network topology with 20 nodes. (a) Homogeneous network with $|V| = 20$, (b) Heterogeneous network with $|V| = 20$.

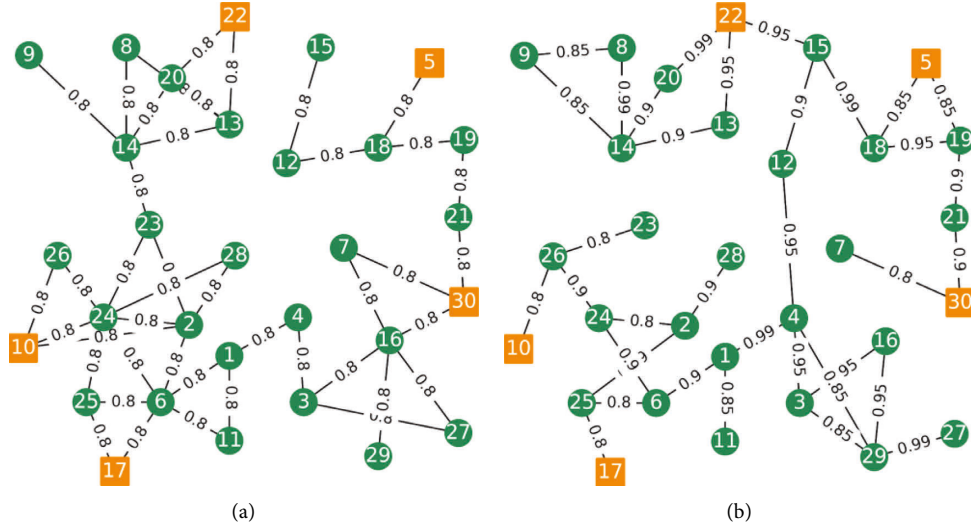


FIGURE 6: Sample network topology with 30 nodes. (a) Homogeneous network with $|V| = 30$, (b) Heterogeneous network with $|V| = 30$.

shown in Figure 7, the link (v_5, v_1) is chosen first. If link (v_5, v_1) is contracted, then node v_5 is deleted from the node set, and the edge (v_3, v_5) is replaced by (v_3, v_1) , the resulting network topology is as shown in Figure 7(b). If link (v_5, v_1) is deleted, from Figure 7(c), it can be seen that the nodes in set K , i.e., node v_1 and v_3 , are no longer connected. Therefore, according to the contraction-deletion algorithm, the 2-terminal reliability can be calculated as $r_{(v_5, v_1)}R(G * (v_5, v_1)) = 0.9 \times R(G^{(b)})$, where $G^{(b)}$ denotes the network topology in Figure 7(b). This process repeats recursively, and the network topology is gradually simplified until only one node remains in the set K . For the compactness of the presentation, the rest of the calculation process are omitted.

With the network reliability datasets, the Root Mean Squared Error (RMSE) is selected as a loss function to measure the performance of estimation models:

$$\text{RMSE} = \sqrt{\frac{1}{M} \sum_{m=1}^M (y_m^{\text{pred}} - y_m)^2}, \quad (6)$$

where y_m is the network reliability of dataset's m -th sample obtained via contraction-deletion method as described in Section 3.1, y_m^{pred} is the estimation model's prediction, and M is the size of the dataset. To avoid the problem of overfitting, the RMSE of the estimation model on both the training and testing sets are considered, denoted as "Train RMSE" and "Test RMSE" in the remaining part of this paper.

Comparative experiments are undertaken for 2 ANN models, the CNN model and our proposed model. To avoid misunderstanding and distinguish the ANN models proposed in [33–36], for the rest of the paper they are referred as "ANN1" and "ANN2", respectively.

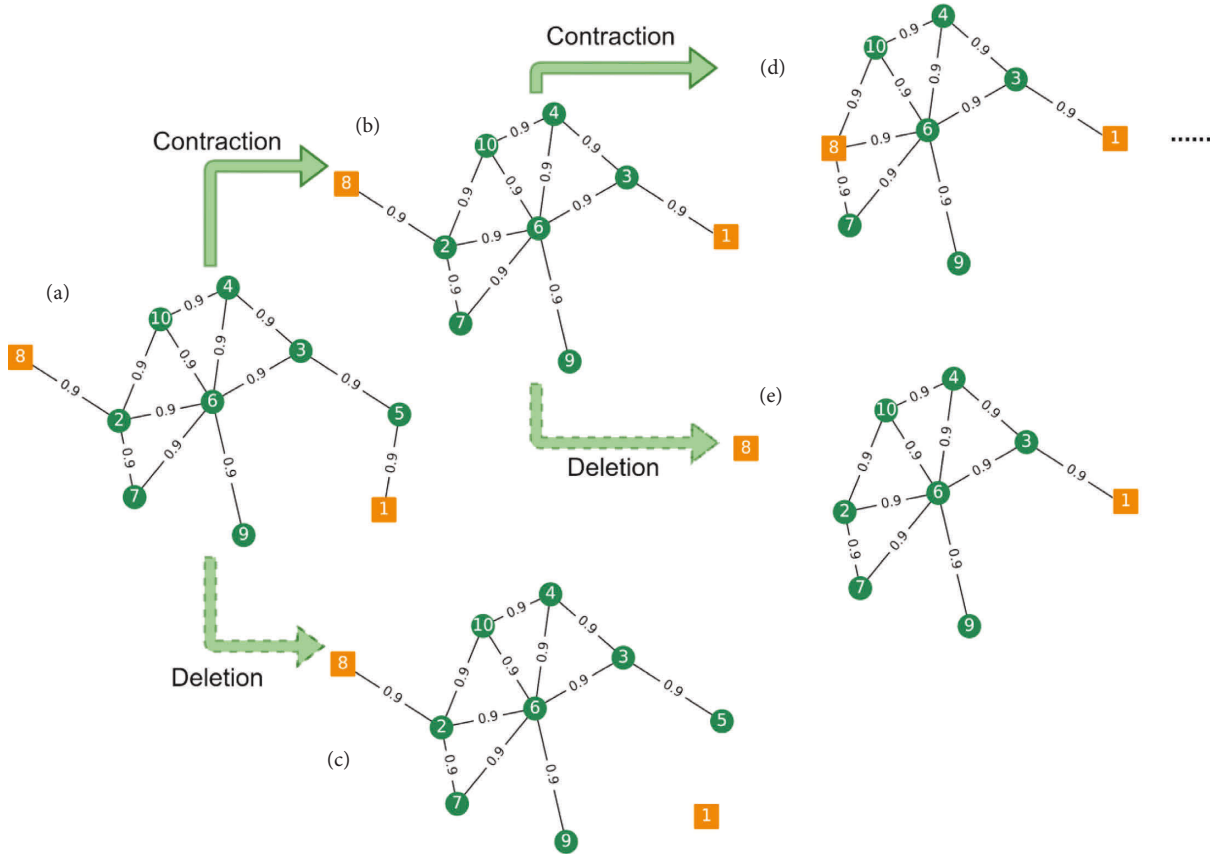


FIGURE 7: Illustrative process of contraction-deletion method for the sample network in Figure 4a.

The same learning rate decay strategy is deployed for all estimation models. The initial learning rate is set to be 10^{-4} . If the validation loss does not improve for 10 epochs, the learning rate is reduced by half. The training process is terminated if one of the following two conditions are met: the learning rate decays below 10^{-7} , or the epoch reaches the maximal number of 1000.

To make a fair comparison with existing estimation models, the number of GCN layers is set to be $D = 4$ with in/out channels configuration of $(1, 8)$, $(8, 16)$, $(16, 16)$, and $(16, 16)$, respectively. The parameter configurations of other existing estimation models are as reported in the literature. 10-fold cross-validation is used to train and evaluate the estimation models, and the mean values are reported as the experiment results.

4.2. Result Analysis. Seeing that some of the existing estimation models can only be applied to homogeneous networks, here the experiment results on homogeneous datasets and heterogeneous dataset are reported separately.

4.2.1. Homogeneous Network. The experiment results in Table 3 compare 2 ANN models, CNN model, and our proposed model on 6 homogeneous datasets (dataset 1 to 6), where the bold font indicates the optimal result and the italic indicates the suboptimal result.

It can be seen that our proposed model outperforms three previous models on all of the homogeneous datasets. Compared to existing models, our proposed GCN model has an average of 59.60% improvement on RMSE. The least significant improvement is on dataset 1, where compared to the best result out of the other three models (ANN1) our GCN model reduces the RMSE by 26.44%; while the most noticeable improvement is on dataset 6, where the RSME of our GCN model is 83.26% lower compared to the previous best result (ANN1).

To better visualize the estimation model performances, here we take the models trained in the first fold on all-terminal reliability datasets (dataset 2, 4, and 6) as show-cases. All models are trained 5 times and the one with the lowest RMSE is selected. Their estimation results are compared on 50 sample networks, as illustrated in Figure 8, Figure 9, and Figure 10. Seeing that the contraction-deletion method is an exact method, the reliability calculated via contraction-deletion is also depicted as the actual reliability. It is easy to see that the deviation between the estimation results obtained via our model and the actual reliability values is the lowest among all estimation models.

There are several factors that may possibly contribute to our proposed model's advantage over previous estimation models. To begin with, the network topology is, by its nature, rich in spatially local patterns. Simply put, the network topology is composed of a set of nodes and a set of links that represents the pairwise relationships or interactions between

TABLE 3: Experiment results of reliability estimation on homogeneous datasets.

	Model	#Param	Train RMSE	Test RMSE	Epochs	Train time	
$ V = 10$	Dataset 1 $ K = 2$	ANN1	45057	0.0228	0.0503	440.00	46.16
		ANN2	36865	0.0501	0.0599	640.33	100.61
		CNN	26977	0.0249	0.0574	212.00	37.22
		GCN	75593	0.0273	0.0370	353.33	168.66
	Dataset 2 $ K = 10$	ANN1	45057	0.0152	0.0453	739.56	129.72
		ANN2	36865	0.0441	0.0532	815.00	132.05
		CNN	26977	0.0258	0.0556	169.00	52.37
		GCN	75593	0.0127	0.0156	197.33	75.66
$ V = 20$	Dataset 3 $ K = 4$	ANN1	82177	0.0182	0.0739	622.56	200.43
		ANN2	39452	0.0706	0.1097	918.67	181.47
		CNN	273505	0.0305	0.0572	224.00	167.87
		GCN	116553	0.0231	0.0256	305.11	244.46
	Dataset 4 $ K = 20$	ANN1	82177	0.0185	0.0919	622.78	154.54
		ANN2	39452	0.0997	0.1361	928.22	204.19
		CNN	273505	0.0323	0.0665	316.22	195.51
		GCN	116553	0.0071	0.0149	244.00	136.01
$ V = 30$	Dataset 5 $ K = 5$	ANN1	144897	0.0216	0.1037	412.33	132.44
		ANN2	41985	0.0647	0.0777	904.89	248.71
		CNN	2472097	0.0537	0.0821	113.22	108.84
		GCN	157513	0.0047	0.0157	212.78	255.41
	Dataset 6 $ K = 30$	ANN1	144897	0.0223	0.1311	202.11	61.76
		ANN2	41985	0.0670	0.0945	887.44	232.95
		CNN	2472097	0.0348	0.0466	125.78	116.17
		GCN	157513	0.0043	0.0078	140.33	131.24

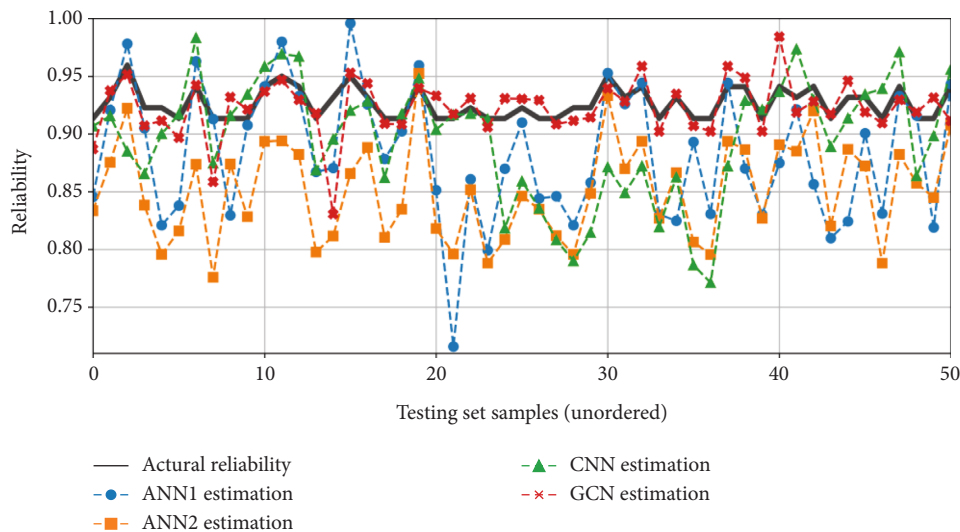


FIGURE 8: All-terminal reliability estimation result (10-node homogeneous networks).

these nodes, therefore the neighbouring nodes and edges are highly correlated. The estimation models based on ANN ignore this type of spatial locality, processing the nodes close to each other in the same way as the nodes far apart from one another. While the CNN model and our proposed GCN model can exploit said spatial locality by utilizing certain types of convolution layer to extract local conjunctions of features [48]. By stacking several convolution layers, the lower-level local features can be composed into a high-dimensional graph-level representation feature.

Besides, as mentioned before, the network topology represented by the graph is intrinsically non-Euclidean. In the CNN model, the network topology is encoded and processed as a $n \times n$ pixel image determined by its adjacency matrix. This way, in the CNN model when the convolution layer extracts local structure patterns, the neighbouring nodes of v_i is defined as those nodes with precedent and subsequent indices, i.e., v_{i-1} and v_{i+1} , rather than the nodes with links connecting to v_i . This encoding scheme may be problematic because the nodes with consecutive indices

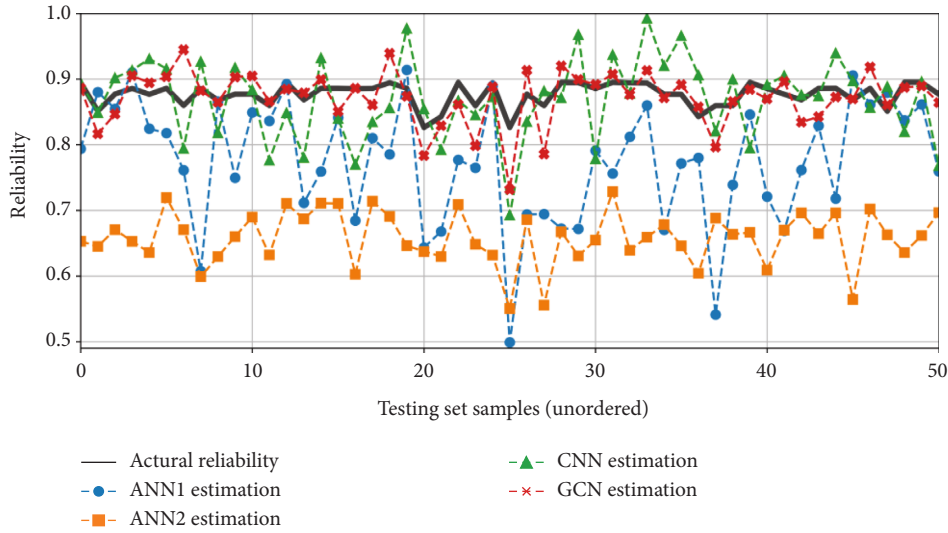


FIGURE 9: All-terminal reliability estimation result (20-node homogeneous networks).

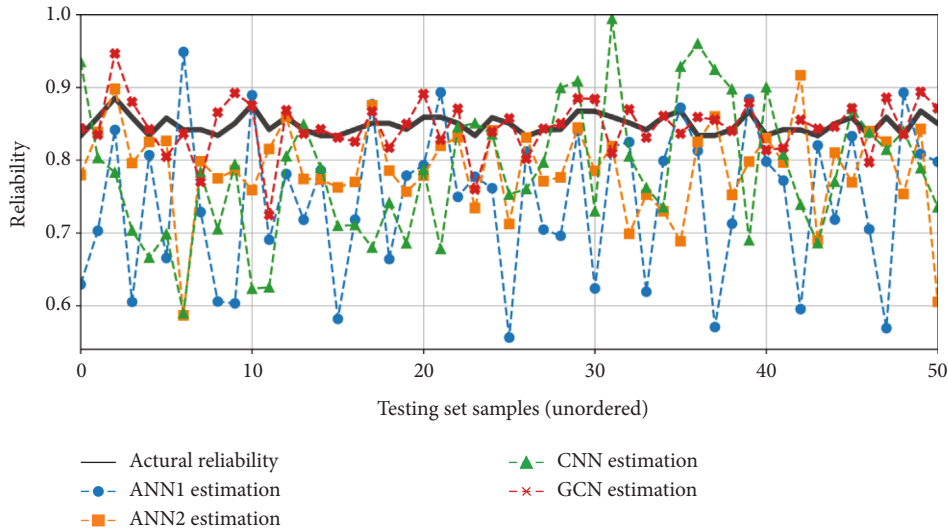


FIGURE 10: All-terminal reliability estimation result (30-node homogeneous networks).

could be many hops away from each other in the actual network topology. For example, in Figure 4(a), node v_1 is 4 hops away from node v_2 , and node v_8 is 3 hops away from node v_9 . Moreover, if the node set is reindexed, the input image of the CNN model can change drastically. Seeing that GCN is a generalization of CNN on non-Euclidean data structures, the representation feature learned via our proposed GCN model has the merits of higher rationality and invariance under node reindexing compared to the CNN model, while maintaining the advantage over ANN models brought by convolution operator.

Although the parameter budget of our proposed GCN model is more expensive than most other models, it can be seen from Table 3 that the training time does not grow dramatically. Compared to the cumbersome computations of the exact methods, relatively speaking, our estimation models are quite effective. For instance, the total

computation time of the contraction-deletion method for dataset 6 is 84911 seconds, that is an average of 10.61 seconds per 30-node network. While for our proposed GCN model, once properly trained, it can give an estimation result with high precision immediately. In addition, it should also be noticed that the parameter budget of the CNN model nearly grows tenfold as the node set size grows from 10 to 20 and from 20 to 30, while the parameter budget of our proposed GCN model only increases by 54.57% and 35.30%, respectively. Thus, in terms of training parameter budget, our proposed model is more scalable to larger networks than the CNN model.

4.2.2. Heterogeneous Network. Comparative experiments are also undertaken for estimation models on 6 heterogeneous datasets (dataset 7 to 12), the results of which are listed

TABLE 4: Experiment results of reliability estimation on heterogeneous datasets.

		Model	#Param	Train RMSE	Test RMSE	Epoch	Train time
V = 10	Dataset 7 K = 2	ANN1	45057	0.0561	0.0812	266.78	18.32
		GCN	75593	0.0387	0.0505	229.78	82.24
	Dataset 8 K = 10	ANN1	45057	0.0304	0.0693	535.89	38.92
		GCN	75593	0.0169	0.0278	205.44	70.21
V = 20	Dataset 9 K = 4	ANN1	82177	0.0416	0.0860	353.22	49.20
		GCN	116553	0.0295	0.0412	410.00	444.87
	Dataset 10 K = 20	ANN1	82177	0.0255	0.0601	307.11	50.00
		GCN	116553	0.0125	0.0256	300.33	187.61
V = 30	Dataset 11 K = 5	ANN1	144897	0.0389	0.0846	113.89	28.56
		GCN	157513	0.0089	0.0286	109.56	126.37
	Dataset 12 K = 30	ANN1	144897	0.0206	0.0457	111.56	31.10
		GCN	157513	0.0061	0.0129	107.44	106.82

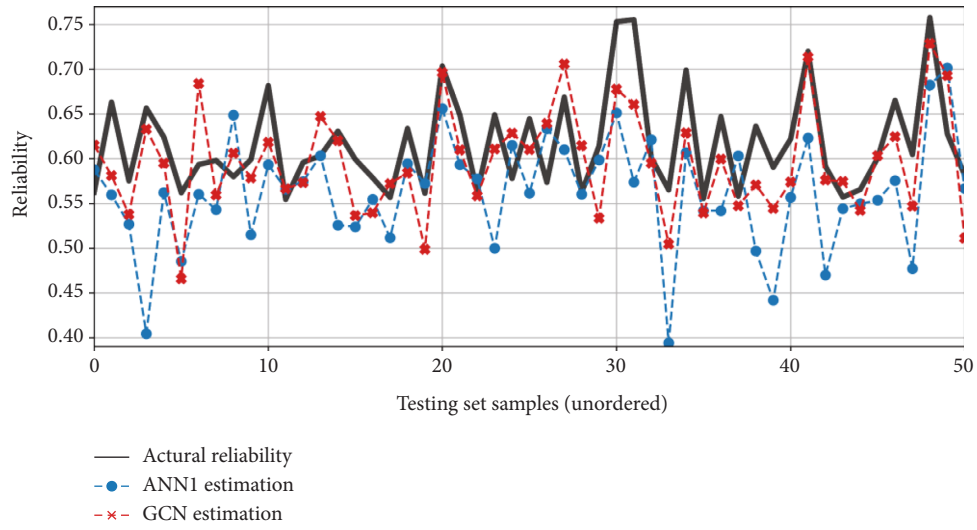


FIGURE 11: All-terminal reliability estimation result (10-node heterogeneous networks).

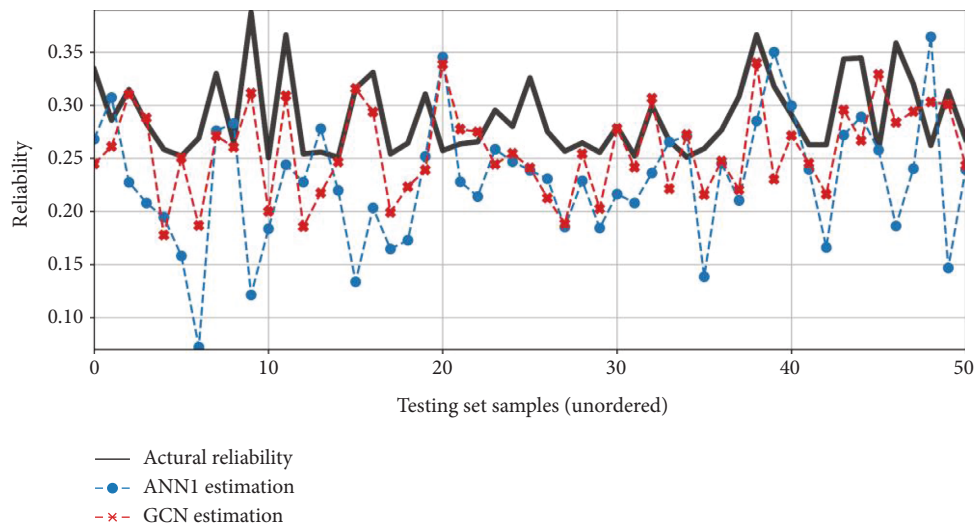


FIGURE 12: All-terminal reliability estimation result (20-node heterogeneous networks).

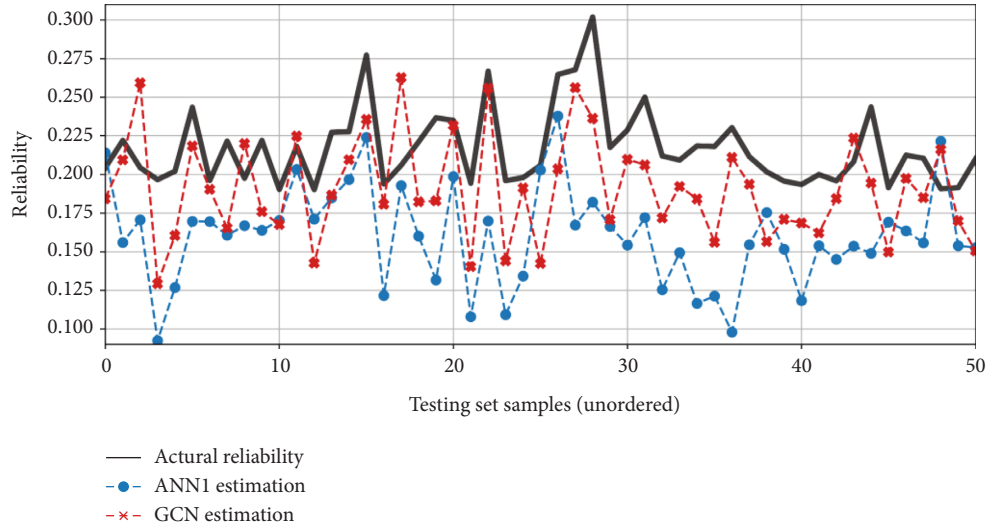


FIGURE 13: All-terminal reliability estimation result (30-node heterogeneous networks).

in Table 4. Because the encoding method in the CNN model and the ANN2 model can only be applied to homogeneous networks, here only the ANN1 model are investigated and compared to our proposed GCN model.

Like before, our proposed GCN model consistently outperforms the ANN1 model across all 6 heterogeneous datasets, with an average of 57.52% improvement on RMSE. The best case is dataset 12, where our proposed model reduces the RMSE by 71.77% compared to the ANN1 model, while the least significant case is dataset 7, where our proposed GCN model's RMSE is 37.81% lower than that of the ANN1 model.

For heterogeneous datasets, we also make the models trained in the first fold on all-terminal reliability datasets (datasets 8, 10, and 12) as illustrative examples. As before, the best-trained model in the 5 runs is selected for the ANN1 model and our proposed GCN model. Figure 11, Figure 12, and Figure 13 compare their estimation results to the actual reliability value on 50 sample networks. It is safe to say that compare to the ANN1 model, our proposed GCN model captures the relationship between the network topologies and reliability values more effectively and accurately.

5. Concluding Remarks

This paper proposes a general framework for the k -terminal estimation problem of communication networks. The analytic contraction-deletion method is made use of to compute the k -terminal reliability for sample networks. Once the dataset with sufficient sample size is obtained, an estimation model based on GCN can be trained. Because the graph convolution layers are capable to extract high-level feature from network topology, the features learned with the GCN model are more effective in representing the network structure, which then results in higher precision. Comparative experiments on communication networks with node set size of 10, 20, and 30 is carried out. The results show that for the 6 homogeneous networks, our proposed GCN model

reduces the RMSE by an average of 59.60%; while for the 6 heterogeneous networks, an average of 57.52% decrease in RMSE is witnessed comparing our proposed model to previous research studies.

However, there are still several improvements that could be applied to our model, which will be the focus of our future work. To begin with, it has been recognized that the performances of neural network-based estimation model depend heavily on the amount of data available. To this end, several topology reduction techniques like degree-1 or degree-2 node reduction can be applied in the company with a contraction-deletion method to accelerate the dataset generation process. Besides, in our proposed model, the dimensions of node and edge features are set to be 1. For edge features, if other information of the communication links, for instance, bandwidth or transmission delay, can also be incorporated, the model may be applied to the task of mission reliability estimation. Whereas for node feature, other than degree centrality, there are several attributes that can also be taken into consideration, such as betweenness centrality, eigenvector centrality, or clustering coefficient, which may further improve the precision of the estimation model. What's more, the development of GCN is a well-studied topic in the field of machine learning. The application of more sophisticated and advanced GCN models [59–62] to the estimation of network reliability would also be a direction of our future work.

Data Availability

The communication network reliability data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

- [1] J. Liu, Z. Zhao, J. Ji, and M. Hu, "Research and application of wireless sensor network technology in power transmission and distribution system," *Intelligent and Converged Networks*, vol. 1, no. 2, pp. 199–220, 2020.
- [2] S. Alam, M. F. Sohail, S. A. Ghauri, and I. M. Qureshi, "Naveed Aqdas, Cognitive radio based smart grid communication network," *Renewable and Sustainable Energy Reviews*, vol. 72, pp. 535–548, 2017.
- [3] A. Zaballos, A. Vallejo, and J. M. Selga, "Heterogeneous communication architecture for the smart grid," *IEEE network*, vol. 25, no. 5, pp. 30–37, 2011.
- [4] S. Azmani, N. Juliana, A. M. Idrose, N. A. Amin, and A. S. M. Saudi, "Challenges of communication system during emergency disaster response in Malaysia: a review," *Journal of Fundamental and Applied Sciences*, vol. 9, no. 4S, pp. 890–904, 2018.
- [5] P. Lieser, F. Alvarez, P. Gardner-Stephen, M. Hollick, and D. Boehnstedt, "Architecture for responsive emergency communications networks," in *Proceedings of the 2017 IEEE Global Humanitarian Technology Conference (GHTC)*, IEEE, San Jose, CA, USA, October 2017.
- [6] D. Avagnina, F. Dovis, A. Ghiglione, and P. Mulassano, "Wireless networks based on high-altitude platforms for the provision of integrated navigation/communication services," *IEEE Communications Magazine*, vol. 40, no. 2, pp. 119–125, 2002.
- [7] W. Ahmad, O. Hasan, U. Pervez, and J. Qadir, "Reliability modeling and analysis of communication networks," *Journal of Network and Computer Applications*, vol. 78, pp. 191–215, 2017.
- [8] A. Rausand and A. Hoyland, *System Reliability Theory: Models, Statistical Methods, and Applications*, Vol. 396, John Wiley & Sons, Hoboken, NJ, USA, 2003.
- [9] D. W. Coit and E. Zio, "The evolution of system reliability optimization," *Reliability Engineering & System Safety*, vol. 192, Article ID 106259, 2019.
- [10] L. King, G. Levitin, and C. Wang, *Dynamic System Reliability: Modeling and Analysis of Dynamic and Dependent Behaviors*, John Wiley & Sons, Hoboken, NJ, USA, 2019.
- [11] H. Pérez-Rosés, "Sixty years of network reliability," *Mathematics in Computer Science*, vol. 12, no. 3, pp. 275–293, 2018.
- [12] M. A. Nabian and H. Meidani, "Deep learning for accelerated seismic reliability analysis of transportation networks," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 6, pp. 443–458, 2018.
- [13] I. B. Gertsbakh and Y. Shpungin, *Models of Network Reliability: Analysis, Combinatorics, and Monte Carlo*, CRC Press, Boca Raton, FL, USA, 2016.
- [14] J. Lu, W. Li, and W. Yan, "State enumeration technique combined with a labeling bus set approach for reliability evaluation of substation configuration in power systems," *Electric Power Systems Research*, vol. 77, no. 5–6, pp. 401–406, 2007.
- [15] G. Hardy, C. Lucet, and N. Limnios, "K-terminal network reliability measures with binary decision diagrams," *IEEE Transactions on Reliability*, vol. 56, no. 3, pp. 506–515, 2007.
- [16] Y. Lei, K. Hou, Y. Wang et al., "A new reliability assessment approach for integrated energy systems: using hierarchical decoupling optimization framework and impact-increment based state enumeration method," *Applied Energy*, vol. 210, pp. 1237–1250, 2018.
- [17] Y.-F. Niu and X.-Z. Xu, "A new solution algorithm for the multistate minimal cut problem," *IEEE Transactions on Reliability*, vol. 69, no. 3, pp. 1064–1076, 2020.
- [18] M. Kvassay, V. Levashenko, and E. Zaitseva, "Analysis of minimal cut and path sets based on direct partial Boolean derivatives," *Proceedings of the Institution of Mechanical Engineers - Part O: Journal of Risk and Reliability*, vol. 230, no. 2, pp. 147–161, 2016.
- [19] W.-C. Yeh, "An improved sum-of-disjoint-products technique for symbolic multi-state flow network reliability," *IEEE Transactions on Reliability*, vol. 64, no. 4, pp. 1185–1193, 2015.
- [20] S. K. Chaturvedi, G. Khanna, and S. Soh, "Reliability evaluation of time evolving Delay Tolerant Networks based on Sum-of-Disjoint products," *Reliability Engineering & System Safety*, vol. 171, pp. 136–151, 2018.
- [21] P. Caçcaval and S.-A. Floria, "SDP algorithm for network reliability evaluation," in *Proceedings of the 2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, IEEE, Gdynia, Poland, July 2017.
- [22] A. Satyanarayana and M. K. Chang, "Network reliability and the factoring theorem," *Networks*, vol. 13, no. 1, pp. 107–120, 1983.
- [23] D. A. Migov and A. S. Rodionov, "Parallel implementation of the factoring method for network reliability calculation," in *International Conference on Computational Science and its Applications*, Springer, Cham, 2014.
- [24] S. N. Nesterov and D. A. Migov, "Parallel calculation of diameter constrained network reliability," in *International Conference on Parallel Computing Technologies*, Springer, Cham, 2017.
- [25] S. Reed, M. Löfstrand, and J. Andrews, "An efficient algorithm for computing exact system and survival signatures of K-terminal network reliability," *Reliability Engineering & System Safety*, vol. 185, pp. 429–439, 2019.
- [26] M. O. Ball, "Complexity of network reliability computations," *Networks*, vol. 10, no. 2, pp. 153–165, 1980.
- [27] W.-C. Yeh, "A quick BAT for evaluating the reliability of binary-state networks," *Reliability Engineering & System Safety*, vol. 216, Article ID 107917, 2021.
- [28] X. G. Chen, "A novel reliability estimation method of complex network based on Monte Carlo," *Cluster Computing*, vol. 20, no. 2, pp. 1063–1073, 2017.
- [29] S. A. Fernandez, A. A. Juan, J. de Armas Adrian, D. G. E. Silva, and D. R. Terrén, "Metaheuristics in telecommunication systems: network design, routing, and allocation problems," *IEEE Systems Journal*, vol. 12, no. 4, pp. 3948–3957, 2018.
- [30] S. Bisht and S. Singh, "Signature reliability of binary state node in complex bridge networks using universal generating function," *International Journal of Quality & Reliability Management*, vol. 36, no. 2, 2019.
- [31] V. Gaur, O. P. Yadav, G. Soni, and A. P. S. Rathore, "A review of metrics, algorithms and methodologies for network reliability," in *Proceedings of the 2019 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, IEEE, Macao, China, December 2019.
- [32] V. Gaur, O. P. Yadav, G. Soni, and A. P. S. Rathore, "A literature review on network reliability analysis and its engineering applications," *Proceedings of the Institution of Mechanical Engineers - Part O: Journal of Risk and Reliability*, vol. 235, no. 2, pp. 167–181, 2020.
- [33] F. He and H. Qi, "A method of estimating network reliability using an artificial neural network," in *Proceedings of the 2008 IEEE Pacific-Asia Workshop on Computational Intelligence*

- and Industrial Application, pp. 57–60, IEEE, Wuhan, China, December 2008.
- [34] C. Srivaree-Ratana, A. Konak, and A. E. Smith, “Estimation of all-terminal network reliability using an artificial neural network,” *Computers & Operations Research*, vol. 29, no. 7, pp. 849–868, 2002.
- [35] F. Altıparmak, B. Dengiz, and A. E. Smith, “Reliability estimation of computer communication networks: Ann models,” in *Proceedings of the Eighth IEEE Symposium on Computers and Communications. ISCC*, IEEE, Kemer-Antalya, Turkey, July 2003.
- [36] F. Altıparmak, B. Dengiz, and A. E. Smith, “A general neural network model for estimating Telecommunications network reliability,” *IEEE Transactions on Reliability*, vol. 58, no. 1, pp. 2–9, 2009.
- [37] A. Davila-Frias and O. P. Yadav, “All-terminal network reliability estimation using convolutional neural networks,” *Proceedings of the Institution of Mechanical Engineers - Part O: Journal of Risk and Reliability*, vol. 236, no. 4, pp. 584–597, 2020.
- [38] N. Widiastuti, “Convolution neural network for text mining and natural language processing,” *IOP Conference Series: Materials Science and Engineering*, vol. 662, Article ID 052010, 2019.
- [39] M. R. Hossain, M. M. Hoque, N. Siddique, and I. H. Sarker, “Bengali text document categorization based on very deep convolution neural network,” *Expert Systems with Applications*, vol. 184, Article ID 115394, 2021.
- [40] A. A. M. Al-Saffar, H. Tao, and M. A. Talab, “Review of deep convolution neural network in image classification,” in *Proceedings of the 2017 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET)*, IEEE, Jakarta, Indonesia, January 2017.
- [41] M. Tripathi, “Analysis of convolutional neural network based image classification techniques,” *Journal of Innovative Image Processing (JIIP)*, vol. 3, no. 2, pp. 100–117, 2021.
- [42] D. N. Le, V. S. Parvathy, D. Gupta, A. Khanna, J. J. P. C. Rodrigues, and K. Shankar, “IoT enabled depthwise separable convolution neural network with deep support vector machine for COVID-19 diagnosis and classification,” *International journal of machine learning and cybernetics*, vol. 12, no. 11, pp. 3235–3248, 2021.
- [43] D. T. S. Kumar, “Video based traffic forecasting using convolution neural network model and transfer learning techniques,” *Journal of Innovative Image Processing (JIIP)*, vol. 2, no. 3, pp. 128–134, 2020.
- [44] N. Kousik, Y. Natarajan, R. Arshath Raja, S. Kallam, R. Patan, and A. H. Gandomi, “Improved salient object detection using hybrid Convolution Recurrent Neural Network,” *Expert Systems with Applications*, vol. 166, Article ID 114064, 2021.
- [45] K. Guo, Y. Hu, Z. Qian et al., “Optimized graph convolution recurrent neural network for traffic prediction,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 2, pp. 1138–1149, 2021.
- [46] X. Geng, Y. Li, L. Wang et al., “Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 3656–3663, 2019.
- [47] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, “Traffic graph convolutional recurrent neural network: a deep learning framework for network-scale traffic learning and forecasting,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4883–4894, 2020.
- [48] J. Zhou, G. Cui, S. Hu et al., “Graph neural networks: a review of methods and applications,” *AI Open*, vol. 1, pp. 57–81, 2020.
- [49] E. Ranjan, S. Sanyal, and P. Talukdar, “Asap: Adaptive structure aware pooling for learning hierarchical graph representations,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, pp. 5470–5477, 2020.
- [50] S. P. Borgatti and B. Ofem, “Social network theory and analysis,” *Social network theory and educational change*, vol. 17, p. 29, 2010.
- [51] Z. X. Zhang, Y. W. Xu, W. N. Hao, and X. H. Yu, “A signed network analysis-based Consensus reaching process in Group decision making,” *Applied Soft Computing*, vol. 100, Article ID 106926, 2021.
- [52] B. Neyshabur, A. Khadem, S. Hashemifar, and S. S. Arab, “NETAL: a new graph-based method for global alignment of protein–protein interaction networks,” *Bioinformatics*, vol. 29, no. 13, pp. 1654–1662, 2013.
- [53] W. Jin, R. Barzilay, and T. Jaakkola, “Junction tree variational autoencoder for molecular graph generation,” in *Proceedings of the International conference on machine learning*, PMLR, July 2018.
- [54] Y.-J. Tan, J. Wu, and H. Z. Deng, “Evaluation method for node importance based on node contraction in complex networks,” *Systems Engineering-Theory & Practice*, vol. 11, no. 11, pp. 79–83, 2006.
- [55] L. B. Page and J. E. Perry, “A practical implementation of the factoring theorem for network reliability,” *IEEE Transactions on Reliability*, vol. 37, no. 3, pp. 259–267, 1988.
- [56] L. B. Page and J. E. Perry, “Reliability polynomials and link importance in networks,” *IEEE Transactions on Reliability*, vol. 43, no. 1, pp. 51–58, 1994.
- [57] S. C. F. Zainabu, *Comparison of Old and New Algorithms for S, T-Network Reliability*, Stellenbosch University, Stellenbosch, 2020.
- [58] C. Morris, M. Ritzert, M. Fey et al., “Weisfeiler and leman go neural: higher-order graph neural networks,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 4602–4609, 2019.
- [59] A. Kazi, L. Cosmo, S.-A. Ahmadi, N. Navab, and M. Bronstein, “Differentiable graph module (dgm) for graph convolutional networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, <https://ieeexplore.ieee.org/abstract/document/9763421>, 2022.
- [60] B. Liang, H. Su, L. Gui, E. Cambria, and R. Xu, “Aspect-based sentiment analysis via affective knowledge enhanced graph convolutional networks,” *Knowledge-Based Systems*, vol. 235, Article ID 107643, 2022.
- [61] D. Bo, X. Wang, C. Shi, and H. Shen, “Beyond low-frequency information in graph convolutional networks,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 5, pp. 3950–3957, 2021.
- [62] R. Li, H. Chen, F. Feng, Z. Ma, X. Wang, and E. Hovy, “Dual graph convolutional networks for aspect-based sentiment analysis,” in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, vol. 1, Long Papers, 2021.