IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 14 December 2025, accepted 19 December 2025, date of publication 23 December 2025,
date of current version 31 December 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3647668

RESEARCH ARTICLE

GCN-RP: Graph Convolutional Network-Based Predictor for Network Connectivity Resilience

AYBIKE ŞIMŞEK
Faculty of Engineering and Architecture, Department of Computer Engineering, Sinop University, 57000 Sinop, Türkiye
e-mail: asimsek@sinop.edu.tr

ABSTRACT Evaluating the connectivity resilience of real-world networks through attack simulations is a time-consuming process. This study proposes a method that trains Graph Convolutional Networks (GCNs) on synthetic graphs—generated from real graphs—to predict the resilience curves (Normalized Largest Connected Component, NLCC) of real networks. The method was tested on six real-world interaction networks under degree-based and random attack strategies. Results show that GCN achieves Mean Absolute Error (MAE) values between 0.03 and 0.23, with higher accuracy under random attacks. Additionally, the entire process—including synthetic graph generation, attack simulations, and GCN training—takes approximately five times less time than direct simulations on real graphs with similar size and density. These findings demonstrate that the proposed approach offers both accurate and efficient resilience estimation.

INDEX TERMS Complex networks, graph neural networks, network resilience, robustness prediction.

I. INTRODUCTION

Complex networks serve as a key tool to describe the structure and dynamics of various natural and social systems. The theory of complex networks offers significant advantages to a variety of disciplines, including mathematics, biology, computer science, social sciences, transportation, and communications. An understanding of the fundamental properties of complex networks is essential for comprehending their functions and roles within a system. On the other hand, systems that possess complex network characteristics can be vulnerable to attacks, which may result in partial or total loss of functionality. Such attacks typically involve the removal of nodes or connections within the network.

The concept of "Network Resilience", defined as the resistance of complex networks to attacks or failures, is of critical importance in understanding the response and structural properties of networks to change. Network resilience has become widely used in many different fields, including ecology, engineering, social sciences, and economics [1], [2], [3], [4], [5]. Therefore, the definitions and measurement methods of network resilience also differ across disciplines, and these differences have led to the emergence of various approaches to assessing the concept. Correct understanding and effective assessment of network resilience is critical to the sustainability and reliability of complex systems. Network resilience is the ability of systems to withstand and recover from disruptions and attacks. Resilience in this context entails fundamental capabilities such as minimizing network performance degradation following an attack, enhancing recovery efficiency within a short timeframe, and ensuring system reliability.

Network resilience studies are relevant to various types of networks, including biological, social, informational, and technological networks. Network resilience research spans various fields, including ecosystem sustainability, transportation and communication infrastructure reliability, social community recovery, and information security. Enhancing network resilience and management improves the design, operation, performance, and security of complex systems.

Recent studies concentrate on understanding, measuring, and enhancing the resistance of networks against attacks by analyzing the resilience of complex networks using various methods and metrics.

Al Musawi et al. [6] conducted an analysis of the vulnerabilities present in complex networks by examining topological properties such as modularity, diversity, clustering coefficient, and diameter. Their research concluded that networks with high modularity and low diversity are particularly susceptible to targeted attacks. These findings

The associate editor coordinating the review of this manuscript and approving it for publication was Alba Amato.

underscore the necessity of developing adaptive resilience models to safeguard critical infrastructures.

Nguyen et al. [7] examined the link between social network resilience and modularity using node attacks based on betweenness and degree centrality. Networks with high modularity were found to be more vulnerable to betweenness centrality attacks, especially in Barabasi–Albert model networks and real social networks, showing sudden disconnections indicative of first-degree percolation transitions.

Gao et al. [8] created analytical tools to evaluate the resilience of complex multidimensional systems. By analyzing system dynamics and topology independently, they demonstrated that the resilience of multidimensional ecological and technological networks can be effectively measured with one-dimensional parameters, and that this method reveals a consistent resilience pattern.

John et al. [9] investigated the impact of weight thresholding by eliminating strong connections on the resilience of real networks. Their findings indicate that network sparsification maintains overall robustness but diminishes the efficiency of targeted attack strategies by significantly altering node centrality rankings.

Zheng et al. [10] developed greedy and simulated annealing algorithms to enhance network resilience. They found that the simulated annealing method, which optimizes topology by random edge reconnection, is more effective.

Yamashita and Ohsaki [11] created a spectral metric for node resilience against random disruptions. They validated it through simulations, showing its effectiveness in predicting resilience in scale-independent networks.

Matoba et al. [12] found that graph coarsening techniques better maintain network resilience compared to sampling methods. They highlighted that these techniques preserve robustness even with significant size reductions.

These studies highlight the complexity of network resilience, indicating the significance of modularity, centrality measures, spectral measurements, effective reduction strategies, and additional spare capacity in preserving structural integrity against targeted and random attacks. Recent research aims to understand, measure, and enhance the resilience of networks by examining complex networks with various methods and measures.

Grassia et al. [13] employed a machine learning-based methodology to fragment networks within complex systems. Their research demonstrated that models trained on small-scale synthetic networks were equally effective when applied to large-scale social and infrastructure networks. The machine learning-based approach outperformed human-based heuristics by an average of 12% and proved effective in predicting early warning signals prior to system fragmentation.

Pan and Wang [14] revealed that their proposed stress-capacity balanced model showed various recovery strategies improved different performance indicators but not all aspects simultaneously. They highlighted the im-portance of evaluation criteria and timing in network recovery strategies.

Lou et al. [15] have conducted a series of studies in the field of network resilience. They created a fast and accurate method to estimate network connectivity durability using Convolutional Neural Networks (CNN). They converted the network's adjacency matrix into a grayscale image for CNN processing, achieving high accuracy in less time than traditional methods in both synthetic and real network experiments. In one of their studies [16], they employed a CNN-based approach to assess network controllability robustness by converting adjacency matrices into images. The CNN models had lower average error rates and were significantly faster than traditional methods.

Lou et al. [17] introduced a knowledge-based CNN approach to predict the controllability robustness of networks. This method showed higher accuracy than traditional single CNN models and surpassed conventional spectral methods and network heterogeneity measures. Specifically, the prediction accuracy improved by classifying the training data using network topology information. Lastly, Lou et al. [18] introduced a new method called mCNN-RP that utilizes multiple convolutional neural networks (CNN) to estimate the connectivity resilience of complex networks subjected to various attacks efficiently and accurately. They developed a CNN classifier that categorizes networks based on prior knowledge of network topology and employs CNN-based estimators specific to each category. Experiments on both synthetic and real-world networks demonstrated that the mCNN-RP method performs better than the single CNN-based method (CNN-RP), with lower error rates. The primary success metric was the estimation accuracy of connectivity resilience curves, assessed by comparing the estimated changes in the size of the remaining Largest Connected Component (LCC) during attacks with the actual values.

Qi and Mei [19] reviewed resilience definitions and measurement methods for complex networks, categorizing them into quantitative and qualitative. They discussed applications in biological, social, technological, and information networks, highlighting machine learning's effectiveness, especially CNN methods, in resilience analysis. They emphasized the importance of topological and dynamic features in network resilience.

In conclusion, current research primarily focuses on evaluating network resilience under various types of attacks utilizing deep learning techniques, such as Convolutional Neural Networks (CNNs). These studies predominantly define network resilience as the preservation of the largest connected component (LCC). Additionally, it is observed that synthetic graphs, including Erdös-Rényi and Watts-Strogatz small-world networks, are commonly employed for training CNNs. The trained CNN models are subsequently used to make predictions on real-world networks [18].

Table 1 presents a comparative overview of recent studies published over the past decade on network robustness

TABLE 1. Summary of related work on network robustness prediction.

Authors	Year	Dataset Type	Method(s)	Target	Key Findings
Y. Lou et al. [18]	2023	Real-world & synthetic	CNN-RP, mCNN-RP, Spectral Heuristics	Connectivity robustness (NLCC)	mCNN-RP outperforms other models in average rank error and top/bottom 10% detection
Y. Lou et al. [20]	2022	Synthetic	CNN-RP Null/Confusion Masking	Connectivity robustness (NLCC)	Robust to missing edges; masking improves MAE
Y. Lou et al. [16]	2021	Synthetic & Real-world	CNN-RP, Spectral heuristics	Connectedness robustness (LCC)	CNN-RP faster than simulation, low error
Y. Lou et al. [17]	2022	Synthetic	CNN-RP	Controllability robustness	Strong generalization across topologies
W. Wu et al. [21]	2024	Real & Synthetic	GIN-MAS, GCN, MLP, LR, RF, DT	connectivity, controllability, thresholds	Best performance on multitasks prediction of four robustness metrics
J. Wang et al. [5]	2023	Synthetic	Mean-field theory, supply chain network (SCNE model)	Robustness	Using interdisciplinary knowledge from supply chain management and complex network theory
Z. Xu et al. [22]	2025	Transport Network	TD3-based DRL	Recovery resilience	Outperforms heuristic baselines in all custom resilience metrics
W. Lei et al. [23]	2017	Synthetic	Spectral metrics	Controllability robustness	Properly tuning high-load edge capacities can reduce cost without harming controllability robustness.
M. Grassia et al. [13]	2021	Synthetic & Real	MLP, GAT	Early warning of collapse	Predicts critical phase transitions before collapse
X. Pan and H. Wang [14]	2018	Weighted Networks	Simulation Heuristics	Recovery resilience	Edge weights affect resilience; weighted paths aid recovery
Z. Yu et al. [24]	2024	Synthetic & Real	Network Robustness Learning via the proposed Graph Transformer (NRL-GT)	Connectivity robustness	NRL-GT outperforms CNN-based models both in terms of controllability robustness learning and connectivity robustness learning.

prediction and resilience estimation. The table includes 11 representative works that address various types of robustness—such as connectivity, controllability, and recovery—across different network types including synthetic, real-world, and weighted networks. Each study is categorized by the dataset used, methodology applied, the specific robustness target, and the performance metrics reported. Notably, the most frequently used metrics include MAE, rank error, AUC, and runtime. This summary highlights the increasing use of deep learning-based models, especially CNNs and GNNs, as well as hybrid and knowledge-based approaches. It also shows the diversity of robustness targets being addressed, from NLCC-based connectivity to early-warning signals of collapse, reflecting the evolving landscape and methodological innovation in this field.

This study addresses a key challenge in the field of network resilience prediction: the time-consuming nature of attack simulations required to assess a network's robustness [25]. While previous works have employed machine learning models—particularly CNN and GNN-based approaches—to estimate robustness, they often rely on simulations performed directly on real-world networks or lack generalizability across diverse network types. The main research gap lies in the limited exploration of using only synthetic graphs—generated based on real-world degree distributions—to train deep learning models that generalize to real networks. To address this, we pose two research questions: (1) Can a GCN model trained on synthetic graphs generated from the degree distribution of a single real-world network accurately predict the Normalized Largest Connected Component (NLCC) curve of the original graph? (2) Can a GCN model trained on synthetic graphs generated from multiple similar real-world networks generalize to other graphs of the same type?

Recent studies have demonstrated that advanced GCN architectures, such as the Adaptive Symbiotic Graph Convolutional Network (ASGCN), can effectively capture multi-scale topological and attribute-based information through co-evolutionary learning mechanisms [26]. These findings support our assumption that learning-based models, when provided with appropriate structural features, can generalize resilience characteristics across different network types.

Our approach utilizes a Graph Convolutional Network (GCN) that receives as input the adjacency matrix along with node-level centrality features (degree, closeness, clustering coefficient, and eigenvector). The main contributions of this study are as follows:

- The paper shows that a GCN trained on synthetic graphs generated from the degree distribution of a single real-world network can accurately predict the NLCC of the original network.
- It is demonstrated that training on synthetic graphs derived from multiple similar real-world networks allows for generalization across networks of the same type.

• The proposed method is shown to significantly reduce computational time by eliminating the need for direct simulations on real graphs.

•The paper presents an extensive experimental analysis validating the proposed approach across various network types and attack strategies.

The rest of this paper is structured as follows: Section II introduces network resilience and basic definitions; Section III presents the materials and methodology; Section IV discusses the experimental results; and Section V concludes the study.

Table 2 provides the abbreviations and their corresponding full forms used throughout the manuscript, listed in alphabetical order to enhance clarity and readability.

TABLE 2. List of abbreviations used in the manuscript.

Abbreviation	Full Term
ASGCN	Adaptive Symbiotic Graph Convolutional Network
AUC	Area Under the Curve
CC	Closeness Centrality
CNN	Convolutional Neural Network
CO	Clustering Coefficient Centrality
DC	Degree Centrality
DRL	Deep Reinforcement Learning
EC	Eigenvector Centrality
ER	Erdös–Rényi Random Graph
GAT	Graph Attention Network
GCN	Graph Convolutional Network
LCC	Largest Connected Component
MAE	Mean Absolute Error
MLP	Multi-Layer Perceptron
NLCC	Normalized Largest Connected Component
QS	q-snapback Graph
ReLU	Rectified Linear Unit Activation
SW	Small World Graph
TD3	Twin Delayed Deep Deterministic Policy Gradient

II. NETWORK RESILIENCE AND BASIC DEFINITIONS

Network connectivity resilience refers to how long and to what extent networks can maintain their integrity under various attacks or random failures [19], [27]. Connectivity is a critical property for the functioning of a network and represents the capacity for connections to exist between any two nodes of the network. This capacity is vital for system functionality. Fragmentation, which occurs during the systematic removal of nodes in a network (e.g. because of an attack or a fault), is usually evaluated on the Largest Connected Component (LCC) of the network [28], [29], [30]. This evaluation process is as follows: First, an initially fully connected network can be broken into several subcomponents because of the removal of nodes. Each subcomponent is connected within itself but is separated from the other subcomponents. The largest of these components (LCC) that emerge during the attack is considered the most important fragment in terms of the overall functionality of the network.

A. LCC BASED RESILIENCE MEASUREMENT

As nodes are sequentially removed under a given attack strategy, the ratio of the number of nodes in the remaining largest connected component of the network to the current total number of nodes in the network is calculated at each step. This ratio is called the Normalized Largest Connected Component (NLCC) and is expressed as: (1) [16]:

$$NLCC(i) = \frac{LCC(i)}{N - i} \tag{1}$$

Here, i=0,1,2,3…,N-1 refers to number of removed nodes; LCC(i) refers to the number of nodes remaining in the largest connected component after removing i nodes; NLCC(i) refers to the ratio of nodes in the largest connected component remaining after removing i nodes from the network; N refers to the total number of nodes present in the network before the attacks. Thus, an attempt is made to estimate the NLCC curve, which consists of the NLCC values obtained starting from when no node was removed from the network until there was only one node left in the network. It is worth noting here that while the LCC curve should show a monotonic non-increasing property as each node is removed, the NLCC may not show this property.

B. ERROR MEASUREMENT

Let the true NLCC created with attack simulations be denoted as At=[At(0), At(1),…, At(N-1)], and the predicted NLCC by the model be denoted as Ap=[Ap(0), Ap(1),…, Ap(N-1)]. At and Ap are used to visualize the difference between the true and predicted values. The prediction error is calculated by (2) [18].

$$\varepsilon = \frac{1}{N} \left| A_t - A_p \right| \tag{2}$$

So, ε is the mean absolute error of the prediction.

C. ERROR MEASUREMENT

During the training of GCN, the degree, closeness, eigenvector, and clustering coefficient centrality measures were provided as input along with the adjacency matrix. This section will explain these measures.

Degree centrality (dc) measures a node's local connectivity and is used to identify highly connected nodes in a network [31]. It is calculated as (3).

$$\mathrm{dc(v)} = \frac{\mathrm{deg(v)}}{\mathrm{N} - 1} \tag{3}$$

Here, deg(v) is the degree of node v; N is the number of total nodes on the network.

Closeness centrality (cc) measures how quickly a node can reach all other nodes in a network. It is the reciprocal of the sum of the shortest path distances from a node to all others. A higher cc value means the node is more central and interacts quickly with other nodes [31]. Formally, it is calculated (4).

$$cc\,(\mathrm{v}) = \frac{\mathrm{N} - 1}{\sum_{\mathrm{u} \neq \mathrm{v}} \mathrm{sp(v, u)}} \tag{4}$$

Here, N is the number of total nodes on the network; sp(v,u) is the length of shortest path between the node v and node u; $\sum_{u\neq v} sp(v, u)$ is the total sum of the shortest paths from node v to all other nodes in the network.

Eigenvector centrality (ec) quantifies a node's relative importance by considering its connections to other nodes with high scores [31]. It is calculated as (5).

$$ec(v) = \frac{1}{\lambda} \sum_{u \in Ngh(v)} e_u \tag{5}$$

Here, e_u is the relative centrality score of the node u; Ngh(v) is the set of neighbors of node v; λ is a constant.

The clustering coefficient centrality (co) measures how closely connected the nodes in a network are to each other, indicating the local density of the network. The clustering coefficient is calculated by dividing the actual number of connections between a node's neighbors by the number of possible connections [31]. It is calculated as (6).

$$co(v) = \frac{2n_v}{k_v\,(k_v - 1)} \tag{6}$$

Here, n_v is the actual number of connections between neighbors of node v; k_v is the total number of neighbors of node v.

III. MATERIALS AND METHODS

This study tackles the network resilience problem in three stages. Initially, three types of synthetic network models were analyzed: the Erdös–Rényi (ER) random graph [32], the Small World (SW) graph [33], and the q-snapback (QS) graph [34]. Attack simulations were performed on synthetic graphs, followed by training and testing of the GCN model. During the second stage, synthetic graphs were generated utilizing the degree distribution data from a single real graph, and attack simulations were subsequently conducted on these synthetic graphs. The model trained on these synthetic graphs was subsequently evaluated using the real graph. Finally, synthetic graphs were generated utilizing degree distribution data from various real graphs of the same type, and the model trained on these synthetic graphs was subsequently evaluated using the real graphs. The following Python modules were used throughout the study: NetworkX [35], PyTorch [36], Pandas [37], [38], NumPy [39], Matplotlib [40].

This section will elaborate on the specifics of these three stages. Prior to that, a brief overview of the processes common to all stages will be presented as follows.

Process Steps

a. Create synthetic graphs.

b. Derive NLCCs using random and degree attack simulations.

c. Compute degree, closeness, clustering coefficient, and eigenvector centrality for each graph.

d. Train the GCN using the adjacency matrix and centrality measures as inputs, with NLCC as the target.

e. Generate predictions using the trained GCN.

A. GENERATION OF ER, SW, AND QS SYNTHETIC GRAPHS, ATTACK SIMULATIONS

For each type of synthetic graph, 1000 graphs with 1000 nodes were generated, resulting in a total of 3000 synthetic graphs. For ER graphs, the probability value was set at p=0.01. For SW graphs, the probability value was set at p=0.1, with each node connected to its k nearest neighbors (k=4) in a ring topology. Lastly, for QS graphs, the probability value was set at q=0.1.

Simulations were conducted using degree attack (where the node with the highest degree centrality was removed) and random attack (where nodes were selected randomly for removal). The NLCC was calculated by removing one node at a time, starting from an intact network. This generated 6 datasets: ER, SW, and QS graphs under both attack types.

Finally, a separate GCN model was trained for each dataset. To evaluate performance, each dataset was partitioned into 80% training and 20% testing subsets.

B. GENERAL GCN ARCHITECTURE AND TRAINING SETUP

To ensure reproducibility and clarify the design choices, we provide detailed information about the GCN architecture and training configuration used in this study.

The proposed GCN-based robustness predictor consists of four graph convolutional layers followed by a fully connected prediction head.

Graph Convolutional Layers: The first GCN layer maps the input node feature vectors of dimension input_dim to a 128-dimensional latent space. The second and third GCN layers preserve this dimensionality, while the fourth GCN layer projects the representation to a hidden_dim space (set to 128 in all experiments). Each GCN layer is followed by a ReLU activation function to introduce nonlinearity and a dropout layer with probability 0.3 to mitigate overfitting.

Graph-Level Pooling: After the final graph convolution, global mean pooling is applied to aggregate node-level representations into a single graph-level embedding. This pooled representation captures the overall structural characteristics of the network.

Fully Connected Layers: The pooled graph embedding is passed through a sequence of four fully connected layers with the following dimensions:

$128 \rightarrow 256 \rightarrow 128 \rightarrow 64 \rightarrow 1000$.

ReLU activation is applied after each hidden layer, and dropout with a rate of 0.3 is used consistently.

The final layer outputs a vector of length output_len, corresponding to the predicted NLCC curve.

Optimizer: The model is trained using the Adam optimizer with a learning rate of 0.001, which provides stable convergence across all datasets.

Loss Function: Mean Squared Error (MSE) loss was used between the predicted NLCC curve and the ground truth curve.

Training Schedule: The maximum number of training epochs is set to 50, with early stopping enabled to prevent

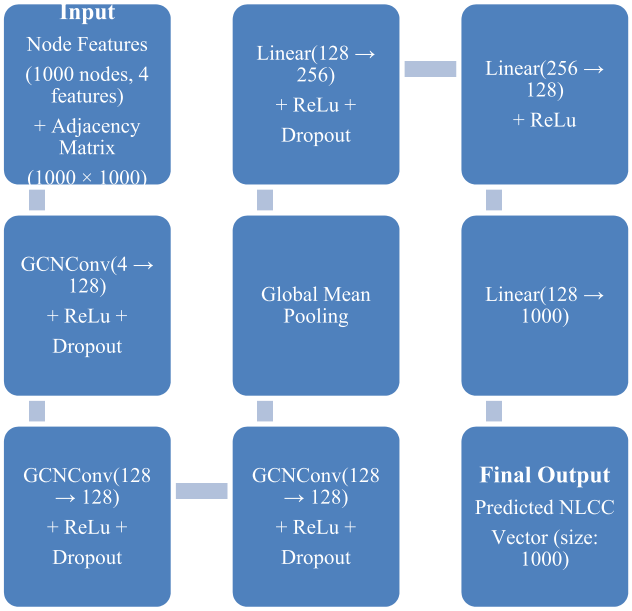


FIGURE 1. GCN model for the synthetic graphs.

overfitting. Early stopping is triggered if the validation loss does not improve for 5 consecutive epochs (patience = 5).

The best-performing model (based on validation loss) is saved during training.

Batch Size: Training is performed with a batch size of 32 graphs per batch.

Input Features and Target: Each node is represented by a feature vector consisting of structural centrality measures, including degree, closeness, clustering coefficient, and eigenvector centrality.

The target output is the Normalized Largest Connected Component (NLCC) curve obtained from attack simulations.

This configuration was selected to balance expressive power and generalization ability while maintaining computational efficiency. The same training protocol and loss function were used consistently across all experiments and baseline models to ensure a fair comparison. The overall architecture of the implemented GCN models is illustrated in Figure 1.

C. ATTACK SIMULATIONS FOR A SINGLE REALWORLD GRAPH, GCN MODEL

This study suggests that a GCN model trained on synthetic graphs—generated based on the degree distribution of a real-world graph—can effectively estimate the NLCC of the corresponding real network. Generating synthetic graphs with fewer nodes yet similar degree distributions, running simulations on these graphs to obtain NLCC curves, and subsequently training the model on them is typically more time-efficient than conducting simulations directly on real graphs. A detailed discussion of this approach is provided in the Results and Discussion section. For this purpose, the following graphs were employed: bio-WormNet-v3, power-bcspwr10 [41], musae-github, musae-facebook [42]. A total of 1000 synthetic graphs, each consisting of 1000 nodes, were generated from each of these graphs. Detailed information about these graphs is provided in Table 3.

A key challenge lies in generating a synthetic graph that closely resembles the original real-world graph. The procedure used for this synthetic graph generation is outlined in Algorithm 1. Given that a graph's degree distribution encapsulates essential structural information, the configuration model was employed to generate synthetic graphs based on the degree distributions of the real-world networks.

TABLE 3. Network Descriptions. The abbreviations BWN, p10, GH, and FB respectively refer to the bio-WormNet-v3, power-bcspwr10, musae-github, and musae-facebook networks. The last two columns represent the diameter and density, respectively. [Note: For the disconnected graph bio-WormNet-v3, all values are calculated based on its initial largest connected component (LCC)].

	$\|V\|$	$\|E\|$	$\langle k \rangle$	Max k	Min k	d	ρ
BWN	16.2K	762.8K	93	1.3K	1	12	0.005
P10	5.3K	13.5K	5	15	3	49	0.001
GH	37.7K	289K	15	9.5K	1	11	0.004
FB	22.5K	171K	15	709	1	15	0.007

Algorithm 1 Generating Synthetic Graphs From a Single Real Network

1. Load the input graph.
2. Compute the degree of each node.
3. Generate a normalized degree histogram.
4. Extract degree values (0, 1, 2,…, k) and their associated probabilities from the histogram.
5. Sample degree values for the desired number of nodes (1,000 in our case) according to the derived probabilities.
6. If the total degree sum is odd, increment it by 1 to ensure it is even (a requirement of the configuration model) [28]
7. Remove self-loops and multiple edges from the generated graph.
8. Save the resulting synthetic graph.
9. Repeat steps 5–8 until the desired number of synthetic graphs is generated (1,000 in our case).

In this study, both degree-based and random attack strategies were employed, resulting in a total of eight datasets—two per graph. The GCN models trained on these synthetic datasets were subsequently evaluated on their corresponding real graphs, with separate testing conducted for each case. The overall architecture of the GCN models is illustrated in Figure 2. It is important to note that the output size of the model is fixed at 1,000. However, the real-world networks used in this study contain more than 1,000 nodes, resulting in NLCC curves with a greater number of data points. To address this mismatch during the model's testing phase, linear interpolation was applied to downsample the NLCC curves of the real networks, reducing them to 1,000 data points.

D. ATTACK SIMULATIONS FOR MULTIPLE REALWORLD GRAPHS, GCN MODEL

Our final proposition is that a GCN model trained on synthetic graphs—constructed using the aggregated degree

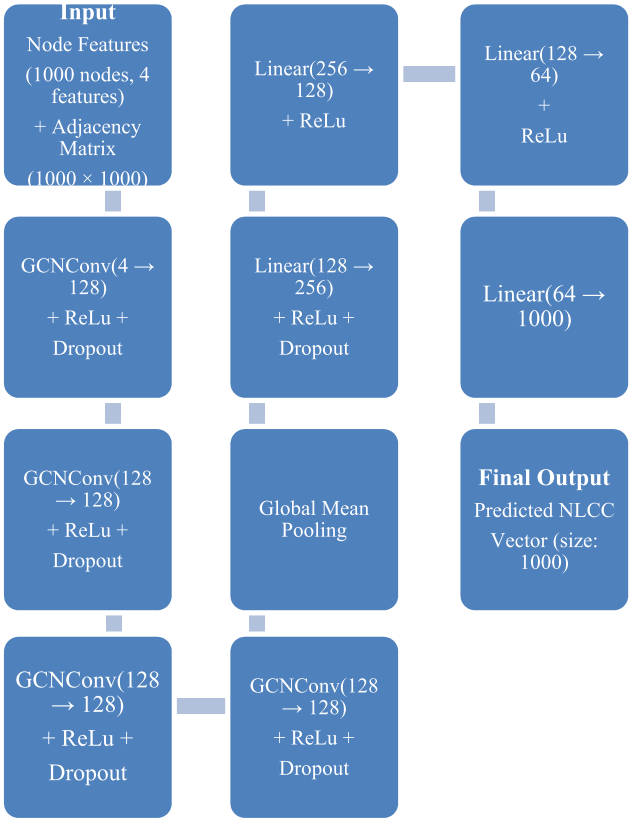


FIGURE 2. GCN model applied to a single network.

Algorithm 2 Generating Synthetic Graphs From Multiple Networks

1. Load the input graphs.
2. Compute the degree of each node in every graph.
3. Generate a normalized degree histogram for each graph.
4. Since the resulting histograms may differ in length (due to varying graph sizes), pad each histogram with zeros to match the length of the longest one.
5. Compute the element-wise average of the padded histograms to obtain a single representative histogram.
6. Extract the degree values (0, 1, 2,. . . , k) and their associated probabilities from the averaged histogram.
7. Sample a degree sequence of the desired size (1,000 nodes in our case) based on these probabilities.
8. If the total degree sum is odd, increment it by 1 to ensure it is even (a requirement of the configuration model) [28]
9. Remove any self-loops and multiple edges in the resulting graph.
10. Save the generated graph.
11. Repeat steps 7–10 until the desired number of synthetic graphs has been generated (1,000 in our case).

distributions of multiple, structurally similar real-world networks—can effectively estimate the NLCC of those real networks. Notably, this approach closely mirrors the procedure outlined in the previous section, with the key distinction occurring during the synthetic graph generation phase. Here, instead of using individual degree distributions, the average degree distribution computed across all real graphs is utilized. The procedure for generating these synthetic graphs is detailed below.

TABLE 4. Network Descriptions. RO, HU, HR, REA, WIK, and ENR denote the soc-gemsec-RO, soc-gemsec-HU, soc-gemsec-HR, ia-reality, ia-frwikinews-user-edits, and ia-enron-large networks, respectively.

	$\|V\|$	$\|E\|$	$\langle k \rangle$	Max k	Min k	d	ρ
RO	41.7K	125.8K	6	112	1	19	0.0001
HU	47.5K	222.8K	9	112	1	14	0.0002
HR	54.5K	498.2K	18	420	1	12	0.0003
REA	6.8K	7.7K	2	261	1	8	0.0003
WIK	25K	68.7K	5	11K	1	7	0.0002
ENR	33.7K	180.8K	11	1.4K	1	13	0.0003

For this purpose, two categories of networks were utilized. The first category comprises social networks, including soc-gemsec-RO, soc-gemsec-HU, and soc-gemsec-HR [41]. The second category consists of interaction networks, namely ia-reality, ia-frwikinews-user-edits, and ia-enron-large [41]. Detailed information about these graphs is provided in Table 4.

GCN models were trained on synthetic graphs generated from each network group using the procedure described in Algorithm 2. Two types of attacks—degree-based and random—were employed, resulting in a total of four datasets, with two datasets generated per group. Each of the four GCN models trained on these datasets was then used to produce individual predictions for every graph within the respective groups.

E. BASELINE MODELS FOR COMPARISON - MULTILAYER PERCEPTRON (MLP) AND LINEAR REGRESSION (LR)

To evaluate the effectiveness of the GCN-based approach, two baseline models were also implemented: a Multi-Layer Perceptron (MLP) and a Linear Regression model. Both models were trained using the same input features as the GCN model. The target was the NLCC curve under attack.

The MLP model consists of a single hidden layer with 128 units and ReLU activation, followed by an output layer matching the length of the NLCC vector. The Adam optimizer was used with a learning rate of 0.001, mean squared error (MSE) was adopted as the loss function, and early stopping was applied with a patience of 5 epochs. Although the maximum number of epochs was set to 500, training typically terminated earlier due to the early stopping mechanism.

The Linear Regression model uses a single fully connected layer without any activation function, directly mapping the input feature vector to the NLCC output. It was trained using the same optimizer and loss function for consistency.

These two models serve as structure-agnostic baselines, allowing us to assess the added value of incorporating graph structure via GCN.

IV. EXPERIMENTAL RESULTS

The experimental results are presented in the same order as outlined in the previous section: beginning with fully synthetic graphs, followed by single real-world graphs, and concluding with multiple real-world graphs of the same type. As previously described, the primary evaluation metric is the error in estimating the NLCC curve. Errors smaller than the standard deviation of the ground truth values are considered successful and are highlighted in bold within the result tables. In the subsections that follow, both the numerical error values and visual comparisons of the actual versus estimated NLCC curves are provided.

A. PERFORMANCE ANALYSIS ON SYNTHETIC ER, SW, AND QS NETWORKS

Figure 3 presents the error plots corresponding to the GCN model's predictions under degree-based and random attacks for ER, QS, and SW graph types. As 20% of the 1,000 generated graphs were allocated for testing, each plot reflects the average performance across 200 test graphs. The shaded regions in the plots represent standard deviation intervals.

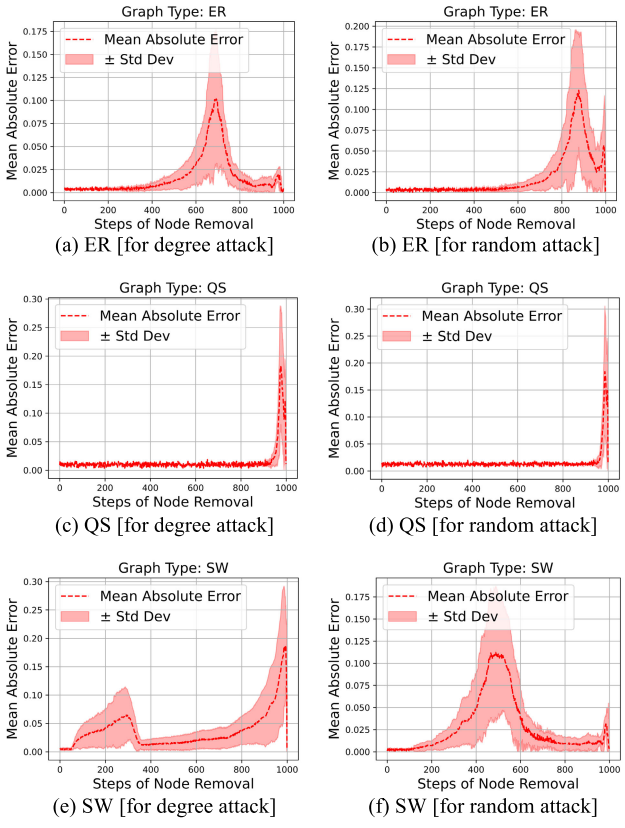


FIGURE 3. Prediction error plots for ER, QS, and SW graph types under both attack strategies. Subfigures (a–b) correspond to ER, (c–d) to WS, (e–f) to SW. Subfigures [a, c,e] illustrate results under degree-based attacks, while [b,d, f] correspond to random attacks.

Overall, the GCN model consistently achieves predictions with near-zero error and demonstrates high accuracy across the different graph types. Additionally, Figure 4 presents the Actual vs. Predicted NLCC curves under both degree-based and random attack scenarios. As observed in the plots, the prediction error—defined as the deviation between the estimated and actual values—is consistently smaller than the standard deviation of the true values for ER and SW graph types. For QS graphs, the error remains below the standard deviation in most cases.

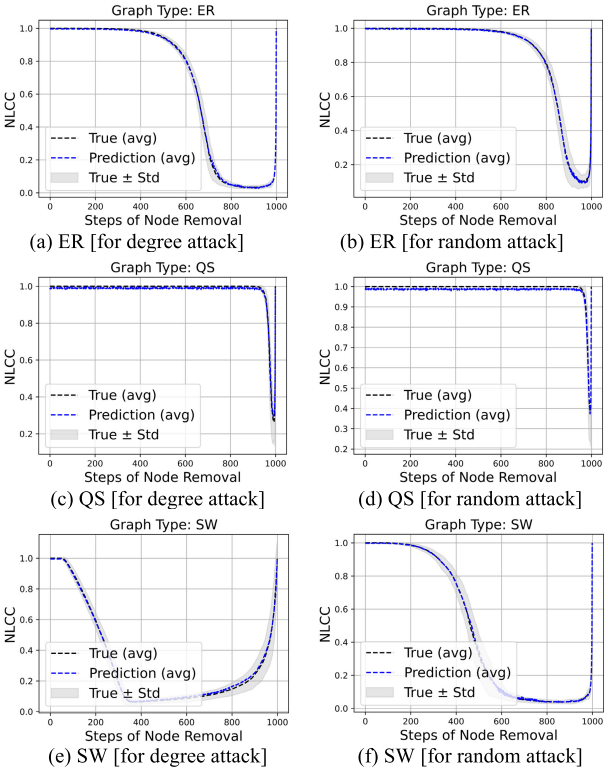


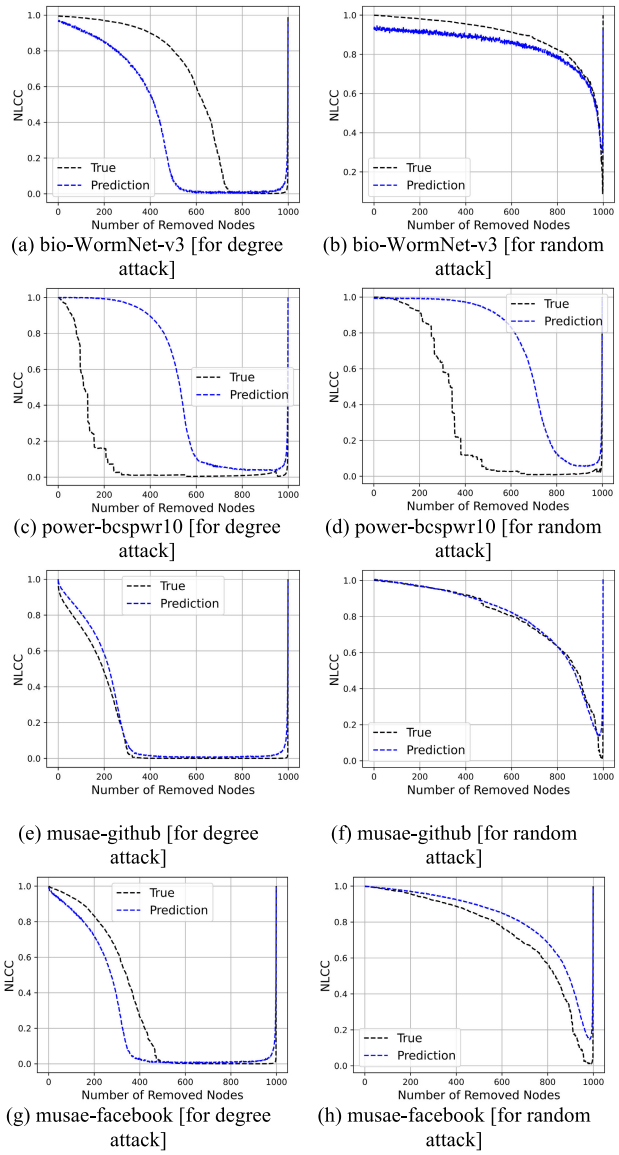
FIGURE 4. Comparison of predicted and actual NLCC curves for ER, QS, and SW graph types under both attack strategies. Subfigures (a–b) correspond to ER, (c–d) to WS, (e–f) to SW. Subfigures [a, c, e] illustrate results under degree-based attacks, while [b, d, f] correspond to random attacks.

B. RESULTS FOR INDIVIDUAL REAL-WORLD GRAPHS

Figure 5 displays the NLCC curves obtained for the Bio-WormNet-v3, power-bcspwr10, musae-github, and musae-facebook networks under both degree-based and random attacks, while Table 5 presents the corresponding prediction errors of the GCN model. Overall, the model demonstrates high predictive accuracy across all networks and attack types, with the exception of the power-bcspwr10 network, where performance is noticeably lower. This network exhibits significantly lower density—approximately one-tenth that of the others—and has a substantially larger diameter (49), which may contribute to the model's reduced effectiveness. Nonetheless, further investigation is required to understand the precise impact of these structural properties on model performance.

C. RESULTS FOR MULTIPLE REAL-WORLD GRAPHS

Finally, the experimental results are presented for two groups of graphs, each comprising three real-world networks.



(a) bio-WormNet-v3 [for degree attack] (b) bio-WormNet-v3 [for random attack]

(c) power-bcspwr10 [for degree attack] (d) power-bcspwr10 [for random attack]

(e) musae-github [for degree attack] (f) musae-github [for random attack]

(g) musae-facebook [for degree attack] (h) musae-facebook [for random attack]

FIGURE 5. Simulation results for individual real-world networks. Subfigures (a–b) correspond to bio-WormNet-v3, (c–d) to power-bcspwr10, (e–f) to musae-github, and (g–h) to musae-facebook. Subfigures [a, c, e, g] illustrate results under degree-based attacks, while [b, d, f, h] correspond to random attacks.

TABLE 5. NLCC curve prediction errors for individual real-world Graphs. The abbreviations BWN, p10, GH, and FB respectively refer to the bio-WormNet-v3, power-bcspwr10, musae-github, and musae-facebook networks. The last two columns represent the diameter and density, respectively.

	Degree		Random	
	True std	MAE	True std	MAE
BWN	0.4101	**0.2270**	0.1319	**0.0545**
P10	0.2754	0.4001	0.4026	**0.3689**
GH	0.2965	**0.0318**	0.2275	**0.0152**
FB	0.3904	**0.0818**	0.2690	**0.0704**

As previously noted, the first group consists of social networks, while the second includes interaction networks. A total of 12 tests were per-formed—covering both degree-based and random attack scenarios for each of the three graphs in both groups. One GCN model was trained per group of graphs-attack combination, resulting in four models in total. The prediction error values obtained from these models are summarized in Table 6. Due to their volume, the corresponding NLCC curve plots are provided in the Figure 6 in the Appendix.

TABLE 6. GCN's NLCC curve prediction errors for multiple real-world Graphs. RO, HU, HR, REA, WIK, and ENR denote the soc-gemsec-RO, soc-gemsec-HU, soc-gemsec-HR, ia-reality, ia-frwikinews-user-edits, and ia-enron-large networks, respectively.

	Degree		Random	
	True std	MAE	True std	MAE
RO	0.4005	**0.0386**	0.3311	**0.1312**
HU	0.4333	**0.1494**	0.2905	**0.0335**
HR	0.4303	**0.2341**	0.2265	**0.0553**
REA	0.0620	0.1200	0.3323	**0.2106**
WIK	0.0469	0.1008	0.2895	**0.0674**
ENR	0.1531	**0.0358**	0.2625	**0.0515**

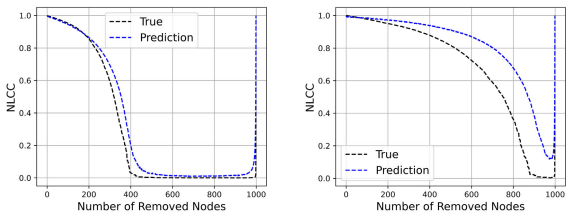
TABLE 7. MLP's NLCC curve prediction errors for multiple real-world Graphs. RO, HU, HR, REA, WIK, and ENR denote the soc-gemsec-RO, soc-gemsec-HU, soc-gemsec-HR, ia-reality, ia-frwikinews-user-edits, and ia-enron-large networks, respectively.

	Degree		Random	
	True std	MAE	True std	MAE
RO	0.4005	**0.0403**	0.3311	**0.1182**
HU	0.4333	**0.1611**	0.2905	**0.0444**
HR	0.4303	**0.2422**	0.2265	**0.0617**
REA	0.0620	0.1300	0.3323	**0.1934**
WIK	0.0469	0.1190	0.2895	**0.0664**
ENR	0.1531	**0.0260**	0.2625	**0.0927**

TABLE 8. Linear model's NLCC curve prediction errors for multiple real-world graphs. RO, HU, HR, REA, WIK, and ENR denote the soc-gemsec-RO, soc-gemsec-HU, soc-gemsec-HR, ia-reality, ia-frwikinews-user-edits, and ia-enron-large networks, respectively.

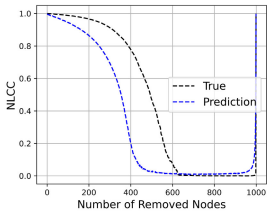
	Degree		Random	
	True std	MAE	True std	MAE
RO	0.4005	**0.0625**	0.3311	**0.1220**
HU	0.4333	**0.1771**	0.2905	**0.0739**
HR	0.4303	**0.2464**	0.2265	**0.0755**
REA	0.0620	0.1251	0.3323	**0.1687**
WIK	0.0469	0.1180	0.2895	**0.0745**
ENR	0.1531	**0.0358**	0.2625	**0.0986**

The results indicate that the models achieve very low prediction errors for NLCC under both degree-based and random attacks. However, model performance degrades in the case of degree attacks on the ia-reality and ia-frwikinews-user-edits networks. A closer examination reveals that these networks contain many nodes with degree 1, while a small subset of nodes exhibit extremely high degrees. For instance, the maximum degree in the ia-frwikinews-user-edits network is approximately 11,000, whereas the total number of edges is 68.7K. Similarly, the ia-reality network has a maximum degree of 261 with 7.7K total edges. As a result, removing just a few high-degree nodes under degree-based attacks rapidly fragments the network into isolated nodes, causing the NLCC values to approach zero. This abrupt structural degradation contributes to increased model error in the early stages of

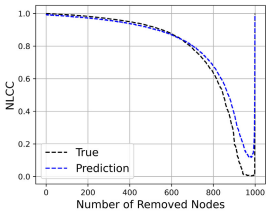


(a) soc-gemsec-RO [for degree attack]

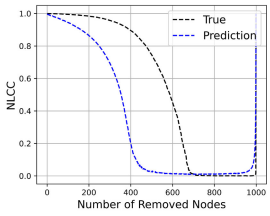
(b) soc-gemsec-RO [for random attack]



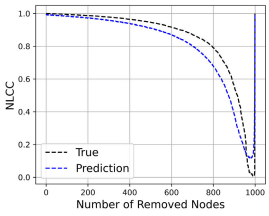
(c) soc-gemsec-HU [for degree attack]



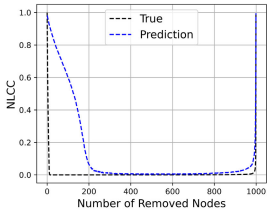
(d) soc-gemsec-HU [for random attack]



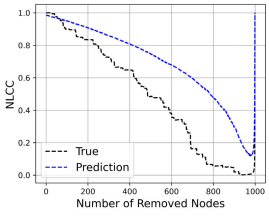
(e) soc-gemsec-HR [for degree attack]



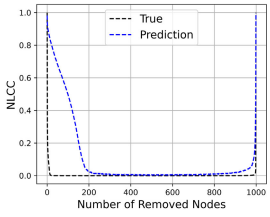
(f) soc-gemsec-HR [for random attack]



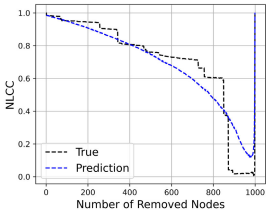
(g) ia-reality [for degree attack]



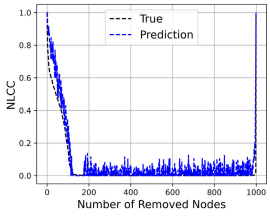
(h) ia-reality [for random attack]



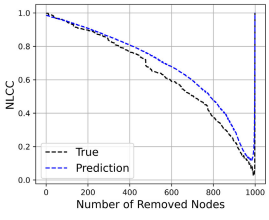
(i) ia-frwikinews-user-edits [for degree attack]



(j) ia-frwikinews-user-edits [for random attack]



(k) ia-enron-large [for degree attack]



(l) ia-enron-large [for random attack]

FIGURE 6. NLCC prediction results for multiple real-world networks. Subfigures (a–b) correspond to soc-gemsec-RO, (c–d) to soc-gemsec-HU, (e–f) to soc-gemsec-HR, (g–h) to ia-reality, (i–j) to ia-frwikinews-user-edits, and (k–l) toia-enron-large. Subfigures [a, c, e,g, i, k] illustrate results under degree-based attacks,while [b, d, f, h, j, l] correspond to random attacks.

the simulation. These effects are clearly visible in the NLCC plots provided in the Appendix.

The results of MLP and Linear model are provided in the Table 7, and Table 8. The performance of the GCN, MLP, and Linear models was compared based on their ability to predict NLCC curves under both degree and random attack scenarios across six real-world networks. The results show that GCN consistently outperforms both MLP and Linear models, especially in scenarios involving complex or heterogeneous network structures. The GCN's awareness of graph topology enables it to model structural vulnerabilities more accurately, leading to lower MAE values that often fall below the standard deviation of the true NLCC curves.

The MLP model, despite being topology-agnostic, delivered reasonably competitive performance in some networks, particularly under random attacks. However, its limited generalization capacity is evident in more complex networks. While early stopping was used to reduce overfitting, the relatively small dataset might still pose a risk of memorization.

The Linear model, as expected, showed the weakest performance, especially under degree attacks. Its simplistic structure limits its ability to capture the nonlinear and structural relationships inherent in the network resilience problem. These findings confirm that incorporating graph structure, as done in GCN, is crucial for reliable prediction of network robustness under node-removal attacks.

V. DISCUSSON AND CONCLUSION

This study demonstrates, through comprehensive experiments, that GCN models trained on smaller synthetic graphs—constructed based on the degree distributions of real networks—can effectively estimate the NLCC curves (i.e., resilience profiles) of real-world graphs under adversarial node-deletion attacks. The models exhibit particularly strong performance under random attacks. In contrast, their predictive accuracy decreases for degree-based attacks. We attribute this to the structural properties of certain networks. Specifically, networks dominated by a few high-degree hubs, with the vast majority of nodes having a degree of one or close to one (e.g., ia-frwikinews-user-edits), display highly predictable behavior under targeted attacks. In such cases, removing just a few hubs causes the network to fragment rapidly into disconnected components, making the use of complex models like GCN unnecessary. Incorporating these structurally extreme networks into the training process may introduce noise and diminish model performance.

The experimental results demonstrate that the GCN model consistently outperforms MLP and Linear models across six real-world networks in predicting NLCC curves under both degree-based and random attack scenarios. For instance, under degree attacks, GCN achieves a MAE as low as 0.0386 (RO) and 0.0358 (ENR), while MLP and Linear models yield higher errors of 0.0403 and 0.0625, respectively, on the same networks. Notably, GCN's MAE frequently falls well below the standard deviation of the ground truth NLCC values, indicating high prediction fidelity.

In contrast, the Linear model shows the weakest performance overall, with MAE values exceeding 0.17 in complex networks such as HU and HR under degree attacks. MLP performs competitively, particularly under random attacks, but its accuracy drops significantly in more heterogeneous networks.

These quantitative findings reinforce the advantage of incorporating graph topology awareness via GCNs. The results confirm that convolutional architectures are better equipped to generalize across diverse network structures and attack scenarios, capturing non-linear and structural dependencies more effectively than shallow models.

Our findings that GCN models can accurately estimate NLCC curves under node-removal attacks align with and extend prior research that leverages deep learning for robustness prediction. For instance, proposed a CNN-based methods [15], [18], [20] to estimate network robustness, demonstrating the feasibility of using learning-based approaches instead of simulation-heavy ones. However, unlike their method, which operates on fixed-size input vectors derived from graph features, our model directly leverages the graph topology via message passing, enabling better generalization across structurally diverse networks.

Moreover, recent studies highlight the strengths of convolutional approaches, such as CNN, in effectively capturing the complex structural dependencies present in heterogeneous or scale-free networks—capabilities that classical regression or simple learning models often lack. Our experiments validate this observation, particularly in degree-based attacks where MLP and Linear models failed to capture rapid structural degradation caused by the removal of hubs—behavior that GCNs captured effectively. The robustness of GCN under such conditions demonstrates its capability to learn fine-grained vulnerability patterns in multilayer systems.

Additionally, like the approach in [16], who employed degree-based synthetic sampling to train neural models, we use the configuration model to generate synthetic graphs. However, our method extends this by grouping structurally similar real-world graphs and using averaged degree distributions to improve generalization—a strategy that was not explored in their work. These comparisons not only reinforce the validity of our approach but also demonstrate how our model contributes a practical, scalable, and accurate alternative for resilience estimation in complex networks.

There are additional considerations that warrant discussion. Most notably, for the proposed approach to be practical, the combined time required to generate synthetic graphs, compute their NLCC curves, and train the GCN model should be significantly less than the time needed to compute the NLCC curve directly on the original real-world graph. While the runtime for obtaining NLCC curves varies across graphs and depends on the type of attack (e.g., degree-based or random), precise estimations are challenging. However, experimental results show that for a graph with approximately 25,000 nodes and a density of 0.003, our method—comprising the generation of 1,000 synthetic graphs, NLCC computation, and GCN training—requires roughly one-fifth of the time needed for direct NLCC calculation on the real graph. In experiments involving graph groups, the time savings were even more substantial. Thus, the proposed approach becomes increasingly advantageous as graph size grows.

Another important consideration concerns the method used to generate synthetic graphs that closely resemble real-world networks. In this study, the configuration model was employed, using sampled degree distributions from real graphs to construct synthetic counterparts. One limitation of this study is that synthetic graphs were generated using only the degree distributions of real networks. While degree is a fundamental structural property, it does not capture other important topological features such as clustering or community modularity, which are commonly observed in real-world networks. As such, the generated synthetic graphs may lack certain structures, potentially affecting the generalizability of the trained GCN models. Future work may consider incorporating additional structural features—such as clustering coefficients or community labels—into the synthetic graph generation process to improve fidelity.

Finally, an additional consideration is how to determine which real-world graphs should be grouped together when training models on multiple networks. In this study, graphs were categorized into social and interaction types. However, beyond this high-level classification, a more data-driven grouping approach could be adopted—such as computing pairwise similarity metrics between graphs. This would enable the formation of more homogeneous groups, from which synthetic graphs could be generated based on structural resemblance, potentially improving model generalizability and performance.

APPENDIX

See Figure 6.

REFERENCES

[1] Y. Zhang and S. T. Ng, "Identification and quantification of node criticality through EWM-TOPSIS: A study of Hong Kong's MTR system," *Urban Rail Transit.*, vol. 7, no. 3, pp. 226–239, Sep. 2021, doi: 10.1007/s40864-021-00155-6.

[2] Y. Zhang and S. T. Ng, "Robustness of urban railway networks against the cascading failures induced by the fluctuation of passenger flow," *Rel. Eng. Syst. Saf.*, vol. 219, Mar. 2022, Art. no. 108227, doi: 10.1016/j.ress.2021.108227.

[3] Z. Chen, J. Wu, Y. Xia, and X. Zhang, "Robustness of interdependent power grids and communication networks: A complex network perspective," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 65, no. 1, pp. 115–119, Jan. 2018, doi: 10.1109/TCSII.2017.2705758.

[4] I. M. Diop, C. Cherifi, C. Diallo, and H. Cherifi, "Robustness of the weighted world air transportation network components," in *Proc. IEEE Workshop Complex. Eng. (COMPENG)*, Jul. 2022, pp. 1–6, doi: 10.1109/COMPENG50184.2022.9905439.

[5] J. Wang, H. Zhou, X. Sun, and Y. Yuan, "A novel supply chain network evolving model under random and targeted disruptions," *Chaos, Solitons Fractals*, vol. 170, May 2023, Art. no. 113371, doi: 10.1016/j.chaos.2023.113371.

[6] A. F. Al Musawi, S. Roy, and P. Ghosh, "Examining indicators of complex network vulnerability across diverse attack scenarios," *Sci. Rep.*, vol. 13, no. 1, p. 18208, Oct. 2023, doi: 10.1038/s41598-023-45218-9.

[7] Q. Nguyen, T. V. Vu, H.-D. Dinh, D. Cassi, F. Scotognella, R. Alfieri, and M. Bellingeri, "Modularity affects the robustness of scale-free model and real-world social networks under betweenness and degree-based node attack," *Appl. Netw. Sci.*, vol. 6, no. 1, p. 82, Dec. 2021, doi: 10.1007/s41109-021-00426-y.

[8] J. Gao, B. Barzel, and A.-L. Barabási, "Universal resilience patterns in complex networks," *Nature*, vol. 530, no. 7590, pp. 307–312, Feb. 2016, doi: 10.1038/nature16948.

[9] J. M. John, M. Bellingeri, D. S. Lekha, D. Cassi, and R. Alfieri, "Robustness of real-world networks after weight thresholding with strong link removal," *Mathematics*, vol. 12, no. 10, p. 1568, May 2024, doi: 10.3390/math12101568.

[10] C. Zheng, Y. Hu, C. Zhang, W. Yu, H. Yao, Y. Li, C. Fan, and X. Cen, "Optimizing the robustness of higher-low order coupled networks," *PLoS ONE*, vol. 19, no. 3, Mar. 2024, Art. no. e0298439, doi: 10.1371/journal.pone.0298439.

[11] K. Yamashita and H. Ohsaki, "Effective node resistance and its implication on network robustness," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2021, pp. 149–153, doi: 10.1109/ICOIN50884.2021.9333923.

[12] T. Matoba, R. Matsuo, and R. Nakamura, "A study on the applicability of graph reduction to evaluating the robustness of complex networks," in *Proc. IEEE 48th Annu. Comput., Softw., Appl. Conf. (COMPSAC)*, Jul. 2024, pp. 1464–1467, doi: 10.1109/COMPSAC61105.2024.00196.

[13] M. Grassia, M. De Domenico, and G. Mangioni, "Machine learning dismantling and early-warning signals of disintegration in complex systems," *Nature Commun.*, vol. 12, no. 1, p. 5190, Aug. 2021, doi: 10.1038/s41467-021-25485-8.

[14] X. Pan and H. Wang, "Resilience of and recovery strategies for weighted networks," *PLoS ONE*, vol. 13, no. 9, Sep. 2018, Art. no. e0203894, doi: 10.1371/journal.pone.0203894.

[15] Y. Lou, Y. He, L. Wang, and G. Chen, "Predicting network controllability robustness: A convolutional neural network approach," *IEEE Trans. Cybern.*, vol. 52, no. 5, pp. 4052–4063, May 2022, doi: 10.1109/TCYB.2020.3013251.

[16] Y. Lou, R. Wu, J. Li, L. Wang, and G. Chen, "A convolutional neural network approach to predicting network connectedness robustness," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 4, pp. 3209–3219, Oct. 2021, doi: 10.1109/TNSE.2021.3107186.

[17] Y. Lou, Y. He, L. Wang, K. F. Tsang, and G. Chen, "Knowledge-based prediction of network controllability robustness," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 10, pp. 5739–5750, Oct. 2022, doi: 10.1109/TNNLS.2021.3071367.

[18] Y. Lou, R. Wu, J. Li, L. Wang, C.-B. Tang, and G. Chen, "Classification-based prediction of network connectivity robustness," *Neural Netw.*, vol. 157, pp. 136–146, Jan. 2023, doi: 10.1016/j.neunet.2022.10.013.

[19] X. Qi and G. Mei, "Network resilience: Definitions, approaches, and applications," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 36, no. 1, Jan. 2024, Art. no. 101882, doi: 10.1016/j.jksuci.2023.101882.

[20] C. Wu, Y. Lou, R. Wu, W. Liu, and J. Li, "CNN-based prediction of network robustness with missing edges," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2022, pp. 1–8, doi: 10.1109/IJCNN55064.2022.9892188.

[21] C. Wu, Y. Lou, J. Li, L. Wang, S. Xie, and G. Chen, "A multitask network robustness analysis system based on the graph isomorphism network," *IEEE Trans. Cybern.*, vol. 54, no. 11, pp. 6630–6642, Nov. 2024, doi: 10.1109/TCYB.2024.3422430.

[22] Z. Xu, S. Liu, and Y. Zhang, "Resilience optimization of transportation network via deep reinforcement learning: An emergency resource allocation framework," in *Proc. 6th Int. Conf. Comput., Netw. Internet Things (CNIOT)*, May 2025, pp. 1–8, doi: 10.1109/cniot65435.2025.11071195.

[23] L. Wang, Y. Fu, M. Z. Q. Chen, and X. Yang, "Controllability robustness for scale-free networks based on nonlinear load-capacity," *Neurocomputing*, vol. 251, pp. 99–105, Aug. 2017, doi: 10.1016/j.neucom.2017.04.011.

[24] Y. Zhang, J. Li, J. Ding, and X. Li, "A graph transformer-driven approach for network robustness learning," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 71, no. 5, pp. 1992–2005, May 2024, doi: 10.1109/TCSI.2024.3355091.

[25] G. Chen, "Controllability robustness of complex networks," *J. Autom. Intell.*, vol. 1, no. 1, Dec. 2022, Art. no. 100004, doi: 10.1016/j.jai.2022.100004.

[26] L. Zhou, Y. Xiao, Z. Ye, H. Zhao, and Z. Liu, "Adaptive symbiotic graph convolutional network," *Neurocomputing*, vol. 637, Jul. 2025, Art. no. 130049, doi: 10.1016/j.neucom.2025.130049.

[27] Y. Chang, C. Jiang, A. Chandra, S. Rao, and M. Tawarmalani, "Lancet: Better network resilience by designing for pruned failure sets," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 3, no. 3, pp. 1–26, Dec. 2019, doi: 10.1145/3366697.

[28] C. M. Schneider, N. A. M. Araújo, and H. J. Herrmann, "Algorithm to determine the percolation largest component in interconnected networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 87, no. 4, Apr. 2013, Art. no. 043302, doi: 10.1103/physreve.87.043302.

[29] R. G. de Jong, M. P. J. van der Loo, and F. W. Takes, "The anonymization problem in social networks," 2024, *arXiv:2409.16163*.

[30] M. Bellingeri, D. Bevacqua, F. Scotognella, and D. Cassi, "The heterogeneity in link weights may decrease the robustness of real-world complex weighted networks," *Sci. Rep.*, vol. 9, no. 1, p. 10692, Jul. 2019, doi: 10.1038/s41598-019-47119-2.

[31] M. Newman, *Networks*. London, U.K.: Oxford Univ. Press, 2018.

[32] P. Erdős and A. Rényi, "On the strength of connectedness of a random graph," *Acta Mathematica Academiae Scientiarum Hungaricae*, vol. 12, nos. 1–2, pp. 261–267, Mar. 1964, doi: 10.1007/bf02066689.

[33] M. E. J. Newman and D. J. Watts, "Renormalization group analysis of the small-world network model," *Phys. Lett. A*, vol. 263, nos. 4–6, pp. 341–346, Dec. 1999, doi: 10.1016/s0375-9601(99)00757-4.

[34] Y. Lou, L. Wang, and G. Chen, "Toward stronger robustness of network controllability: A snapback network model," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 9, pp. 2983–2991, Sep. 2018, doi: 10.1109/TCSI.2018.2821124.

[35] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using networkX," in *Proc. Python Sci. Conf.*, Jun. 2008, pp. 11–15.

[36] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," 2019, *arXiv:1912.01703*.

[37] *Pandas-Dev/Pandas: Pandas*, Zenodo, Geneva, Switzerland, Feb. 2020.

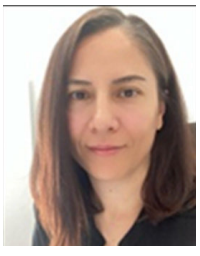
[38] W. McKinney, "Data structures for statistical computing in Python," in *Proc. Python Sci. Conf.*, 2010, pp. 56–61, doi: 10.25080/majora-92bf1922-00a.

[39] C. R. Harris et al., "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020, doi: 10.1038/s41586-020-2649-2.

[40] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 90–95, 2007, doi: 10.1109/mcse.2007.55.

[41] R. A. Rossi and N. K. Ahmed, "The network data repository with interactive graph analytics and visualization," in *Proc. 29th AAAI Conf. Artif. Intell.*, vol. 29, 2015, pp. 4292–4293. [Online]. Available: http://networkrepository.com

[42] J. Leskovec and A. Krevl, "SNAP datasets: Stanford large network dataset collection," SNAP Group, Stanford, CA, USA, Jun. 2014. [Online]. Available: http://snap.stanford.edu/data



AYBIKE ŞIMŞEK received the B.Sc. degree in computer engineering from Selçuk University, in 2003, the M.Sc. degree in computer engineering from Gazi University, in 2010, and the Ph.D. degree in electrical, electronics, and computer engineering from Düzce University, in 2018. She was a Postdoctoral Researcher with the Department of Computer Science, Humboldt University of Berlin, from August 2018 to July 2019. She was a Lecturer with the Department of Computer Programming, Düzce University, from January 2015 to December 2020, and as an Assistant Professor with the Department of Computer Engineering, Düzce University, from December 2020 to June 2022. From June 2022 to October 2025, she was an Assistant Professor with the Department of Computer Engineering, Turkish Military Academy, National Defense University. She is currently an Assistant Professor with the Department of Computer Engineering, Sinop University, Türkiye. Her research interests include social network analysis, complex networks, and epidemic modeling.