Advances in Computer, Signals and Systems (2023)
Clausius Scientific Press, Canada

DOI: 10.23977/acss.2023.070310
ISSN 2371-8838 Vol. 7 Num. 3

A Robust Combinatorial Defensive Method Based on GCN

Xiaozhao Qian, Peng Wang*, Yuelan Meng, Zhihong Xu

Research Center for Complexity Sciences, Hangzhou Normal University, Hangzhou, 311121, China
wangpeng_621@163.com
**Corresponding author*

Keywords: Graph convolutional networks, graph adversarial attack, robustness, defense

Abstract: Graph Convolutional Neural Networks (GCNs) often demonstrate poor robustness when faced with adversarial attacks, which can be generated with malicious intent. Several heuristic defensive methods have been proposed to mitigate this issue, but they are often vulnerable to stronger adaptive attacks. Recently, researchers have shown that the non-robust aggregation functions used in GCNs are responsible for their vulnerability, and adversarial training in the popular space can enhance the model's accuracy and robustness. Building on this prior research, this paper analyzes the robustness of the winsorised mean function and the mean aggregation function from the perspective of model interpretability, based on the theory of breakdown points and influence function robustness. We propose an improved robust combinatorial defensive method, WLGCN, which replaces the mean aggregation function in the GCN operator with the more robust winsorised mean aggregation function, and incorporates a robust adversarial regularizer on the manifold space hidden layer $H^{(1)}$ of the GCN. Finally, we evaluate the robustness of the proposed model under different levels of adversarial perturbation cost, using accuracy and classification margin as evaluation metrics. The experimental results demonstrate that the proposed defensive approach can effectively enhance the model's robustness against adversarial attacks while maintaining model accuracy, when compared to other baselines.

1. Introduction

With the advent of deep learning, convolutional neural networks have made remarkable progress in various fields, including computer vision and natural language processing. However, traditional convolutional neural networks are only suitable for processing Euclidean space data, such as images and text, which have the characteristic of translation invariance [1]. On the other hand, graph data, a type of non-Euclidean data, has gained widespread attention due to its prevalence in modeling complex relationships in the real world, such as social network relationships, transportation relationships, and protein structure relationships. The local structure of each node in the graph may be vastly different, making translation invariance no longer applicable. To solve this issue, researchers have extended convolutional eural networks to graph data, resulting in graph convolutional neural networks. GCNs have shown great promise in extracting features from graph data, which enables us to perform various downstream tasks, including node classification, link prediction, and graph classification.

However, recent research has revealed that deep learning models are susceptible to adversarial perturbations, which can severely affect their output. For instance, minor variations in a few pixels in an image, although imperceptible to the human eye, can cause significant changes in the model's output. Similarly, graph neural network models are vulnerable to adversarial perturbations, such as adding or deleting edges, modifying node features, etc. Adversarial attacks on graph convolutional neural networks can have severe consequences, particularly in practical applications such as social networks, where malicious users can easily create fake followers to add false information, manipulate online comments, and product websites, or deceive target users to mislead analysis systems [2].

Therefore, the security issue of graph convolutional neural networks is currently one of the research hotspots. In-depth research on graph adversarial attacks and countermeasures can promote their successful application in a wider range of fields. Compared to other areas of deep learning, graph adversarial attacks are more challenging because graph attributes are not only affected by perturbations, but also discrete structures. Thus, developing robust countermeasures against graph adversarial attacks is crucial to ensure the reliability and trustworthiness of GCN models in practical applications.

We organized the remainder of the study as follows. To start with, Section 2 introduces the current researches that are related to our work. The preliminary definitions of GCNs, the attack and defense unified modeling are given in Section 3. Section 4 analyzed the robustness of the aggregation function. Section 5 illustrates the combinatorial defensive method we used that including winsorisedconv and the latent adversarial training. In Section 6, we provide detailed results and experimental analysis. We draw the conclusions in Sections 7.

2. Related Works

In recent years, research on adversarial attacks and defenses in graph convolutional neural networks (GCNs) has received increasing attention from researchers. Z ügner et al. [3] were among the first to propose the Nettack attack algorithm for graph adversarial learning, which modifies node data features and their connections to generate small adversarial perturbations guided by a scoring function. This sparked a wave of research on adversarial attacks on GCNs, with subsequent studies conducted by Dai [4], Wang [5], Zhou [6], Sun [7], and others.

Concurrently, research on defense methods against adversarial attacks on GCNs has also gained momentum. Feng [8] et al. introduced Graph Adversarial Training (GAT) as a robust defense method based on dynamic regularization using graph structure. Zhu et al. [9] proposed a sample-based Batch Virtual Adversarial Training to enhance the model's robustness. According to the study by G ünnemann et al. [10], GCN defense methods against adversarial attacks can be broadly classified into three categories:

1) Data pre-processing [11, 12]: For instance, graph purification, which purifies the perturbed graph to obtain a clean graph and trains the GCN model on it.

2) Model training [13, 14, 15]: For example, adversarial training, which trains the model by labeling adversarial samples with the correct label, giving the model defense capability against corresponding attack methods. However, this method is limited by the attack methods and cannot defend against unknown attacks.

3) Model architecture modification [16]: For example, introducing attention mechanisms to learn how to differentiate between adversarial perturbations and clean samples, and training a robust GCN model by penalizing the weights of adversarial nodes or edges.

In summary, a significant body of literature has emerged that focuses on the development of adversarial attacks and defenses in GCNs. The proposed defense methods offer a promising avenue for mitigating the impact of adversarial attacks, and further research is required to enhance their effectiveness and robustness.

3. Preliminary Definition

3.1. Definition of the GCNs

A graph convolutional neural network is a deep learning model designed for processing graph-structured data. It utilizes a neighbor aggregation strategy and a message passing mechanism to learn representations and perform classification tasks on nodes.

Formally, given an attribute graph $G=(A,X)$, where $A\in[0,1]^{N\times N}$ is the adjacency matrix and $X\in[0,1]^{D}$ represents the D-dimensional feature vector of each node. The set of nodes is denoted as $V=\{1,2,...,N\}$and the feature set as $F=\{1,2,...,D\}$. The labels of a subset of nodes $V_L\in V$ are drawn from a set of classes $F=\{1,2,...,C_k\}$. The goal of the GCN is to map the nodes in the graph to their corresponding class labels, by iteratively aggregating the features of neighboring nodes to update the representation of the target node.

The definition of the l layer of graph convolutional neural networks is as follows [17]:

$$h_v^{(l)}=\sigma^{(l)}[AGG^{(l)}\{(A_{uv},h_u^{(l-1)}W^{(l)}),\forall u\in N'(v)\}] \tag{1}$$

The representation vector of the target node is obtained by taking into account the information of its neighboring nodes The information is then combined with the representation vector of the target node from the previous layer, using an aggregation function $AGG^{(l)}$, to obtain the message vector. The message vector is then passed to the target node, using a normalized adjacency matrix A, a weight function $W^{(l)}$, and an activation function $\sigma^{(l)}$, to compute the representation vector $h_v^{(l)}$ of the target node.

The GCN employs a neighbor aggregation strategy and a message passing mechanism to iteratively update the representation vector of a target node, by aggregating and transferring the information from its neighboring nodes. As shown in Figure 1, to compute the representation vector of the target node at layer l, the information of its neighboring nodes is first obtained. Then, the information is combined with the target node's own representation vector from layer $l-1$, using an aggregation function, to obtain the representation vector of the target node at layer l.

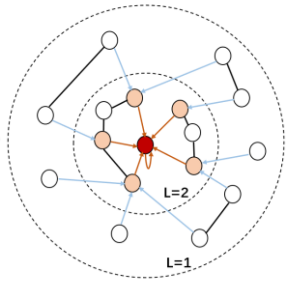

Figure 1: Node information aggregation

3.2. Definition of the Attack Unified Modeling

Consider an original attributed graph $G^{(0)}=(A^{(0)},X^{(0)})$, where $A^{(0)}$ is the original adjacency matrix and $X^{(0)}$ is the original node feature matrix. Let $\tilde{G}=(\tilde{A},\tilde{X})$ be the graph obtained by adding adversarial perturbations, where structural attacks are applied to the adjacency matrix A and feature attacks are applied to the node feature matrix X . Let Δ be the cost of adversarial perturbations, θ be the model parameters obtained by training on a set of instances, and $f_\theta(A,X)$ be the graph convolutional neural network model. The goal of the attacker is to maximize the loss function of the target node v_t on $f_\theta(A,X)$ in order to achieve the desired attack effect, which can be defined as follows [17]:

$$\max_{\tilde{A}s.t.\|\tilde{A}-A\|_0+\|\tilde{X}-X\|_0<\Delta} l(f_\theta(\tilde{A},X)) \tag{2}$$

subject to $\|\tilde{A}-A\|_0+\|\tilde{X}-X\|_0<\Delta$, where $\|\tilde{A}-A\|_0$ represents the number of non-zero elements in a vector. The constraint controls the size of the perturbations and limits the total number of modifications on the node feature matrix and the adjacency matrix to Δ.

3.3. Definition of the Defense Unified Modeling

As research on graph neural network attacks has progressed, the study of defensive methods against GNN-based smuggling has also made rapid progress, proposing corresponding defensive strategies for different attacker models. In this section, we provide a general definition of defensive models for graph data adversarial attacks and their related concepts:

Definition as: [18]

$$\min \sum_{\tau_i\in T} l_{att}(f_{\theta^*}(\hat{G},\tau_i),c_i) \tag{3}$$

$$s.t.\theta^* = \arg\min_{\theta} \sum_{\tau_j\in S_L} l_{train}(f_\theta(\tilde{G},\tau_j),c_j) \tag{4}$$

Let G be an original network or a perturbed network. The goal of defense is to minimize the loss function of the attacked model, making it as close as possible to the loss of the model that has not been attacked.

3.4. Definition of the Winsorised Mean

Winsorised mean: a method for handling outliers that is different from truncating the mean (removing outliers) or treating them equally as the sample mean. It limits the influence of outliers within a certain threshold. Specifically, in order statistics data, it replaces the values of the top 100α% (0 ≤ α ≤ 0.5) with the value of the upper segment median and the values of the bottom 100α% with the value of the lower segment median. Finally, the adjusted statistical data sample is averaged using the following mathematical expression [19]:

$$\bar{x} = \frac{1}{n}[\sum_{i=g+2}^{n-g-1} x_{(i)} + (1+g)(x_{(g+1)} + x_{(n-g)})] \tag{5}$$

4. Aggregation Function Robustness Analysis

The message passing mechanism is the core of graph convolutional neural networks (GCNs), and the commonly used aggregation function in existing GCN models based on message passing is the mean aggregation function. The sample mean is widely used but has no resistance to outliers. If one or more outlier samples exist in a sample, it may lead to a complete breakdown of the model's output, so it needs to be carefully considered when applied. When the sample data is more scattered or has a large range, the sample median is more robust, but the sample median is only one data point and is not fully utilized. An intuitive idea is to use the Winsorised mean, which is robust to outliers, for processing.

4.1. Breakdown point theory analysis

The theory of breakdown point is used to measure the robustness of a function f under data perturbations. The breakdown point m can be intuitively understood as the minimum number of data points that need to be added to a data sample set, in order to make the output of the function f diverge to infinity.

Definition: The breakdown point $m \in (f, N)$ is defined as the minimum perturbation value that causes the function f to breakdown, where N is the set of all possible perturbations. The breakdown point is calculated as follows [20]:

$$\min_{m \in N}\{\frac{m}{|N_v|+m}: \sup_{\tilde{N}:|\tilde{N}_V|=m} |f(N_v \mathrm{U} \tilde{N}_v)| = \infty\} \tag{6}$$

The concept of the "breakdown point" has been widely used in robust statistics. Chen et al. [19] found that the mean function is non-robust based on the crash point theory. The crash point of the mean function is $1/(|N_v|+1)$, which means that in the worst case, only a small perturbation is needed to make the output of the function go to infinity. In contrast, to make the upper bound of the winsorised mean tend to infinity, at least $\lfloor \alpha n \rfloor + 1$ perturbed data points with infinite values need to be injected into the function. Compared with the crash point of the mean function, which is $1/(|N_v|+1)$, the crash point of the winsorised mean is higher, indicating its higher robustness to outliers.

4.2. Influence Function Robust Estimation Analysis

Robust Estimation. It refers to selecting appropriate methods to minimize the influence of outliers and gross errors in data samples, in order to obtain the best estimate. A method is considered to be "robust" when the results obtained from it closely match the true values, despite the presence of outliers. If the estimated values are significantly different from the true values, it indicates poor performance of the method and suggests that the outliers are affecting the model. [21].

Influence Function. The influence function refers to the measure of the robustness of an estimator, and the corresponding robustness metric can be obtained through the influence function. This concept was initially proposed by Hampel [22] based on the concept of infinitesimals, and is defined as:

$$IF(x,T,F) = \lim_{\varepsilon \to 0} \frac{T[(1-\varepsilon)F + \varepsilon \Delta x] - T(F)}{\varepsilon} \tag{7}$$

where F is the distribution function, T is the estimator, and Δx is the dot product. If it is a finite sample, the corresponding empirical influence function can be obtained [19]:

$$S_c(X,T) = T([1 - \frac{1}{n} - n] f_{(n-1)}(x) + \frac{1}{n} \delta(x - x_0)) \tag{8}$$

where $f_{(n-1)}(x)$ is the empirical influence function, and δ is a coefficient related to ε.

Analysis of the Influence Function of the Winsorised Mean. Taking the difference between the trimmed mean of $n+1$ observations and that of n observations yields [19]:

$$\bar{x}_{n+1} - \bar{x} = \frac{1}{n+1} \begin{cases} (1+g)x_{(g)} - \bar{x}_n - gx_{(g-1)}, & x < x_g \\ x - \bar{x}_n, & x_{(g)} \le x \le x_{(n-g+1)} \\ (1+g)x_{(n-g+1)} - \bar{x}_n - gx_{(n-g)}, & x > x_{(n-g+1)} \end{cases} \tag{9}$$

By substituting the sample influence function formula (7) and considering symmetry, the sample influence function of the trimmed mean can be obtained as [19]:

$$S_{(x)} = \begin{cases} x_{(g)} + g(x_{(g)} - x_{(g+1)}), & x < x_{(g)} \\ x, & x_{(g)} \le x \le x_{(n-g+1)} \\ x_{(n-g+1)}, & x > x_{(n-g+1)} \end{cases} \tag{10}$$

The impact function of the winsorised mean is shown in Figure 2, and it can be seen that the impact function is a bounded jump function. The impact function provides a measure of robustness, and it indicates that the winsorised mean is more robust than the arithmetic mean, as it can resist the influence of outliers. In contrast, the impact function of the mean is unbounded, and the mean is very sensitive to outliers, lacking any robustness.

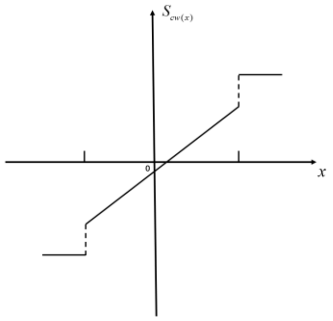

Figure 2: Winsorised mean influence function graph

5. Combinatorial Defensive Strategy Based on GCN

5.1. The Robust Aggregation Function

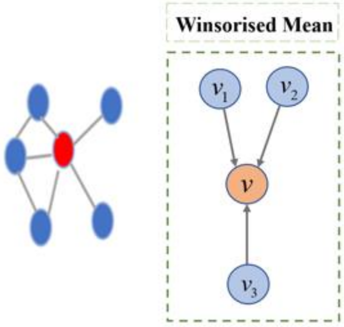

Figure 3: WinsorisedConv Aggregation

In Section 4, the robustness of the Winsorised mean function compared to the mean function is analyzed from two perspectives: the breakdown point and the influence function. By analyzing the robustness of the Winsorised mean and mean functions, this paper proposes an improved robust defense operator, WinsorisedConv, by modifying the aggregation function in its graph convolution operator based on the mainstream message-passing mechanism of graph convolutional neural network framework. The specific implementation is to replace the mean aggregation function with the more robust Winsorised mean aggregation function in the design stage of the graph convolution operator, as shown in Figure 3.

5.2. Graph hidden adversarial regularization

Miyato et al. [23] pointed out that perturbing word embeddings does not affect the mapping to any word and proposed this method as a robust classifier for normal text. Meanwhile, Stutz et al. [19] showed that adversarial instances can simultaneously improve robustness and accuracy if they are on a low-dimensional embedding of popular samples. To address similar problems in Graph Convolutional Networks, a direct analogue of perturbing word embeddings in GCNs is perturbing the output of the first hidden layer, denoted as $H^{(1)}$ which combines node features and graph information. In this paper, we use a proxy of the latent popular space and inject robust adversarial regularization terms to indirectly perturb graph and node information, implicitly enhancing the model's robustness against structural attacks. Experimental results in Section 6 show that this helps to reduce the success rate of GCN under adversarial attacks (robustness) while maintaining or improving the model's accuracy. The model framework is illustrated in the figure 3.

The forward propagation formula for the GCN model is shown as follows:

$$H^{(l+1)} = \sigma(\tilde{A} H^{(l)} W^{(l)}), l \geq 0 \tag{11}$$

where $H^0 = X$ represents the initial node representation. For the sake of symbol expression convenience, let's assume that all nodes are represented by d-dimensional vectors at all layers, denoted as $W^{(l)} \in R^{n \times d}$. Consider a standard two-layer GCN model that tries to find the optimal weight parameters $\theta := (W^{(1)}, W^{(2)})$, in order to minimize the model output loss function f_θ, that is $\min_\theta f_\theta(G, X)$.

The hidden layer combines the structure of the model graph and node information, and can directly perform adversarial training on it as follows: [24]

$$\min_\theta \max_{\zeta \in D} f_\theta(H^{(1)} + \zeta) \tag{12}$$

where f_θ is the loss function based on the perturbation amount ζ at $H^{(1)}$.The imperceptible vibration noise defined as $D := \{\zeta : \| \zeta_i \| \leq \varepsilon, \forall i \in \{1, \ldots, n\}\}$.

The perturbation amount in (11) is jointly chosen over all nodes in the graph setting, which is different from the common adversarial setting where each individual adversarial sample seeks its own perturbation. The result is a high computational cost, which further exacerbates the nested min-max optimization. To alleviate this problem, we further adopt adversarial training with a standard regularization variant, aimed at improving the smoothness of the model's perturbation predictions, as shown in Equation (13).

$$\min_\theta \mathrm{l}_\theta(\tilde{A}, X) := f_\theta(H^{(1)}) + \gamma R_\theta(H^{(1)}) \tag{13}$$

Here, γ is a balancing parameter, and the regularizer R_θ is defined as the Frobenius distance between the original model output (the second layer) and the output after perturbation.

After simplification, (13) becomes as follows: [24]

$$R_\theta(H^{(1)}) = \max_{\zeta} \| \tilde{A}\zeta W^{(2)} \|_F^2 \ s.t. |\zeta_{i:}| \le \varepsilon \qquad (14)$$

To find the perturbation parameter ζ, the perturbation parameter ζ is as follows:

$$\nabla_{\zeta} tr(\tilde{A}\zeta W \zeta^t \tilde{A}^T) = (W^T\zeta + W\zeta^T)\tilde{A}\tilde{A}^T \qquad (15)$$

Here, the $W = W^{(2)}W^{(2)^T}$. The overall procedure is summarized in Algorithm 1.

Algorithm 1 Hidden Adversarial Regularization for GCN

```
input: A, X
While not converged for (13) do
    While not converged for (14) do
        Apply ADAM to find $\zeta^*$ (gradient in $\zeta$ from Eq (15))
    Take one step of ADAM in $\theta$ with the gradient computed by $\nabla_\theta f_\theta(H^{(1)}) + \gamma\nabla_\theta \| \tilde{A}\zeta^* W^{(2)} \|_F^2$
```

5.3. The Robust Combined Defense Method

Through the analysis of the robustness of the winsorised mean and mean function in Section 4, this paper proposes an improved robust defensive method WLGCN based on the mainstream message-passing mechanism graph convolutional neural network framework. The specific implementation method is to incorporate potential adversarial perturbation training into the hidden layers, and to select a more robust trimmed mean aggregation function to replace the mean aggregation function when designing the graph convolutional operator. The overall architecture of the model is shown in Figure 4.

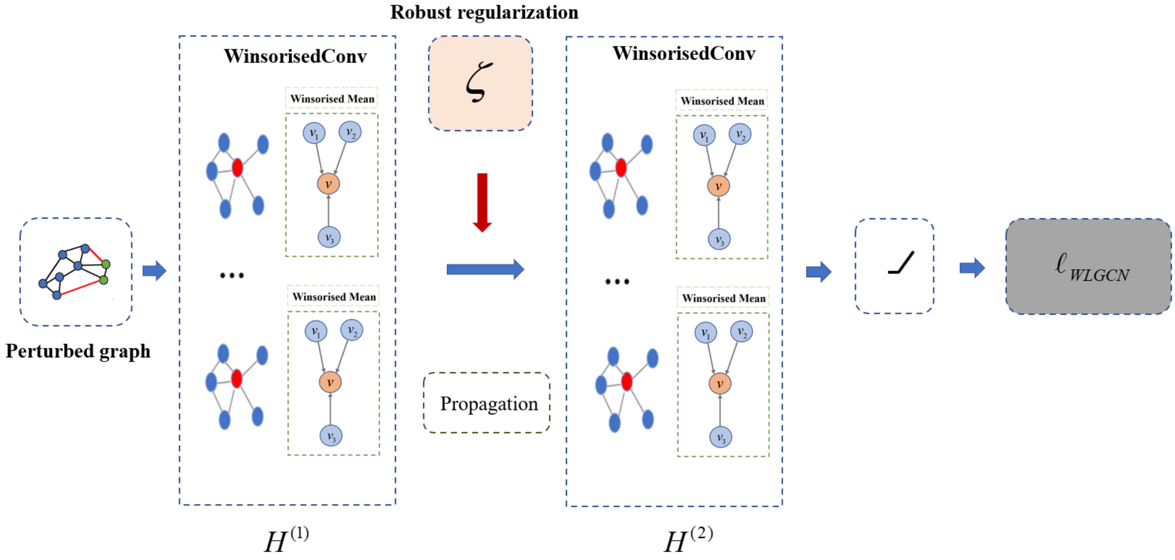

Figure 4: WLGCN framework

6. Experiments and Analysis

In order to evaluate the efficacy of the defensive method of the WLGCN in this paper, the study conducted experiments on three real datasets and two graph neural network attack models. The effectiveness of this method was compared with the latest to defensive method validate its performance.

6.1. Datasets and Evaluation Metrics

This study conducted research on three real datasets, including Cora [25], Cora-ML [26], and Citeseer dataset. Table 1 provides a statistical description of the datasets. The maximum connected component (LCC) of the datasets was calculated. NLCC and ELCC represent the maximum connected component of the node set and the maximum connected component of the edge set, respectively.

1) Cora dataset: The Cora dataset is a citation network dataset that contains a large number of academic papers, classified into 7 categories. It consists of 2485 articles and 5096 citation records, with each node containing 1433 features.

2) Cora-ML dataset: The Cora-ML dataset is a citation network dataset that contains a large number of academic papers related to machine learning, classified into 7 categories. It consists of 2810 articles and 7981 citation records, with each node containing 2879 features.

3) Citeseer dataset: The Citeseer dataset is also a citation network dataset that contains 2110 academic papers and 3668 citation relationships, classified into 6 categories, with 3703 features.

The degree distribution of Cora, Cora-ML, and Citeseer datasets are shown in Figure 5. We can find the majority of nodes are of low degree.

Table 1: The statistical description of the datasets

Datasets	Nodes	Edges	Features	Labels
Cora	2485	5069	1433	7
Cora-ML	2810	7981	2879	7
Citeseer	2100	3668	3703	6

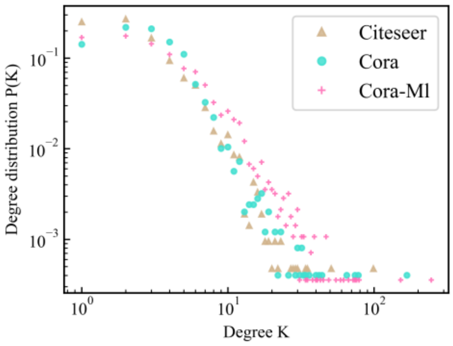

Figure 5: The three datasets degree distribution

In this study, we use Accuracy, Classification Margin (CM) and the variant of the CM to assess the performance of the models.

Classification Margin: To evaluate the effectiveness of the attacks, we use Classification Margin as a measure, which represents the maximum distance from the misclassified target node to the correct class boundary. The formula is as follows [27]:

$$CM = \max_{y'_{v_t} \neq y_{v_t}} H' - H_{v_t} \tag{16}$$

Here, v_t is the target node, y_{v_t} is the class label of the target node v_t. H_{v_t} is the model output before the target node v_t is attacked, and H' is the model output after the target node v_t is attacked.

6.2. Attack Algorithms

In this experiments, two classic adversarial attack algorithms with strong attack performance are used, namely NETTACK target attack algorithm and Metattack non-target attack algorithm. The following is a brief introduction to these two algorithms

1) NETTACK [28] is an algorithm that first selects candidate edges and features based on important data characteristics. It then designs two evaluation functions to assess the change in the target confidence after modifying the candidate edges and features. Finally, it updates the adversarial network iteratively by modifying the highest scoring edge or feature.

2) Metattack [29] is a global attack algorithm that treats the input network G as a hyperparameter and constructs a bi-level optimization problem. It utilizes the meta-gradient based on network edges to iteratively update the adversarial network.

6.3. Baselines

To verify the effectiveness of the proposed robust defense method WLGCN, this paper compares it with GCN and three other benchmark defense methods, namely GWNN, AGNN, DGAT, and GCN. The following briefly introduces these four defense methods.

GWNN: A novel graph convolutional neural network (GCN) that uses graph wavelet transform to solve the drawbacks of previous spectral GCN methods that relied on graph Fourier transform [30]. Unlike graph Fourier transform, graph wavelet transform can be obtained through fast algorithms without matrix decomposition, which reduces computation costs and provides good interpretability for GCN.

AGNN: A variant of GCN that performs semi-supervised classification on graph-structured data, where the model uses an efficient layer-wise propagation rule based on spectral graph convolution that approximates the first-order proximity [31].

DGAT: Adversarial training (AT) is a regularization technique that has been shown to improve the robustness of models against perturbations in image classification. Directed graph adversarial training (DGAT) incorporates graph structure into the adversarial process and automatically identifies the impact of perturbations from neighboring nodes, introducing additional adversarial regularization to defend against worst-case perturbations.

DGAT can resist the impact of adversarial perturbations in worst-case scenarios and reduce the impact of perturbations from neighboring nodes [32].

GCN: A scalable semi-supervised learning method for graph-structured data that uses an efficient layer-wise propagation rule, where the specific spectral-domain graph convolution adopts a weighted averaging method to aggregate messages from neighboring nodes [33].

GCN-W: A GCN variant model based on Winsorised Convolution, which we designed, is employed in the ablation study of the experimental section.

GCN-L: A variant of the GCN model based on Latent Adversarial Training. The model is utilized in the ablation study of the experimental section.

6.4. Adversarial Attack and Defense Experiments

In the defense process, two main issues need to be addressed: 1) maintaining the performance of graph neural network models on clean samples, and 2) minimizing the impact of adversarial attacks on the performance based on the first issue.

Accuracy of the Model before Attack. Due to the winsorised mean aggregation employed by WLGCN, some extreme value information may be discarded during the aggregation process, which may result in a decrease in the accuracy of this method. To verify the accuracy of this approach, this paper conducted 10 experiments on node classification tasks based on three types of original clean graph datasets before adversarial attacks, and took the average value. The results are shown in

Table 2. It can be observed that the proposed WLGCN method achieves the best performance on both the Cora and Citeseer datasets. The performance on the Cora-ML dataset is only slightly lower than that of the best-performing model, LATGCN. These results indicate that although the proposed model aggregation function is adjusted to discard some extreme values during winsorised mean aggregation, the accuracy of the model has not decreased, and the overall accuracy of the model has been improved by introducing potential adversarial perturbation training in the manifold space $H^{(1)}$

Table 2: The classification accuracy

Datasets	WLGCN	GWNN	AGNN	DGAT	GCN	GCN-W	GCN-L
Cora	**0.8592**	0.8589	0.8568	0.8458	0.8538	0.8590	0.8475
Cora-ML	0.8526	0.8537	0.8567	0.8555	0.8620	0.8524	**0.8628**
Citeseer	**0.7495**	0.7264	0.7191	0.7132	0.7245	0.7219	0.7469

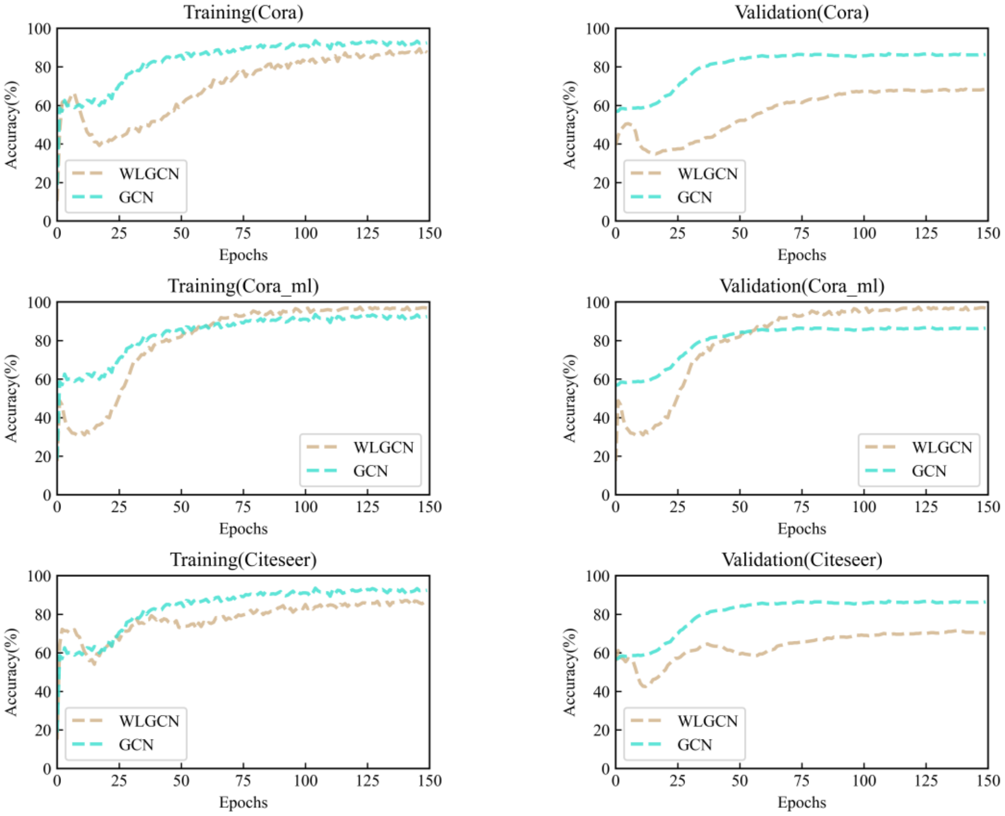

Figure 6: Training curves of the GCN and WLGCN on the training and validation of three datasets

Model Convergence before Attack. Next, we investigated the convergence behavior of the WLGCN and GCN models during the training process. Specifically, in this section, we observed the training and validation performance of the GCN and WLGCN models on the three different datasets after each training epoch, as shown in the Figure 6. It can be seen that the performance of both GCN and WLGCN becomes stable after 100 epochs on the different datasets, indicating that the designed improvements to WLGCN did not affect the convergence speed of the model.

Robustness of Models after Attack. In adversarial training, to further explore the robustness of different models, we utilized NETTACK, a potent and inconspicuous graph adversarial attack algorithm, to conduct our experiment. The degree distribution of nodes in all three datasets displayed a low-degree distribution with small degree values, as illustrated in Figure 5. Consequently, we devised an attack that imposed 0-9 perturbed edges to each target node, with the addition of 9 perturbed edges regarded as a substantial degree of perturbation. Our aim was to assess the efficacy of the defense algorithm against attacks of varying degrees of perturbation.

By recording the results of adversarial training averaged over 10 runs, the overall performance of the robustness of different models is obtained, using $\sum_{q=0}^{9} q \times cm_q$ as the robustness indicator, where q is the size of the perturbation, and cm_q is the classification margin under the attack perturbation size q. The smaller the value of this measure, the stronger the robustness of the corresponding model.

From Table 3, it can be observed that the proposed method is superior to the baseline methods under both direct and indirect attacks, indicating that the proposed improved model has high robustness. Compared with indirect attacks, it can be found that the model's robustness indicator data under direct attacks is much larger than that under indirect attacks, which indicates that all models are more susceptible to the influence of direct attacks. This is consistent with the previous research results of Zhu et al [32], showing that the effectiveness of attacks by directly manipulating and modifying the target node features is higher than that of manipulating other nodes to affect the target node.

In the ablation study, WLGCN demonstrates superior robustness performance in most cases, except for Cora (Direct) and Cora-ML where the model's robustness performance slightly lags behind GCN-L and GCN-W.

Table 3: The model robustness under direct and indirect attacks (Nettack)

Datasets	Attack	WLGCN	GWNN	AGNN	DGAT	GCN	GCN-W	GCN-L
Cora	Direct	31.51	33.37	37.44	37.96	37.64	36.70	**27.08**
	Indirect	**3.27**	9.33	13.23	8.51	15.15	7.40	11.17
Cora-ML	Direct	18.47	22.67	26.18	21.95	24.53	**16.34**	20.36
	Indirect	0.92	8.33	4.60	3.07	3.96	**0.39**	8.24
Citeseer	Direct	**14.84**	15.27	40.65	32.96	40.19	26.49	23.82
	Indirect	**5.61**	7.78	19.80	16.55	21.07	7.84	12.12

Analysis of Adversarial Training Perturbations. 1)Targeted Attack: In adversarial training, to investigate the robustness of the model under different levels of attack perturbation, this paper records the model classification robustness under different datasets and perturbation amounts under Nettack direct attack, as shown in Figure 7.

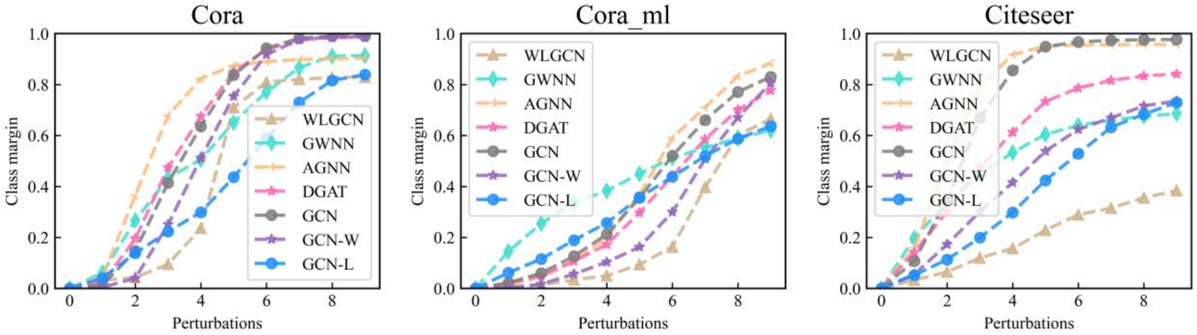

Figure 7: The class margin curves of the models under direct targeted attack

The CM metric represents the distance between a target node and the correct classification boundary. Therefore, when a node is correctly classified, the corresponding model output confidence should be higher. On the Citeseer dataset, the proposed WLGCN method outperforms other methods significantly. On the Cora dataset, when the perturbation amount is 1, all models perform well because the minimum degree of the Cora dataset is 2 (including self-loops), making it difficult for a single perturbation to change the model output. When the perturbation amount is greater than 2, the performance of GWNN and other baseline methods drops rapidly, while WLGCN can still maintain high performance. With the increase in perturbation intensity, the CM of different models increases, which can be attributed to the fact that most nodes in the Cora dataset have few neighboring nodes. After a direct attack on the target node, the information aggregated by the aggregation function is more heavily perturbed, leading to lower model robustness. In summary, the proposed method has better performance in model robustness in adversarial attacks.

2) Global Attack: In adversarial attack methods targeting graph neural networks, there is a class of attackers who focus on modifying a small number of edges to significantly degrade the performance of the graph neural network model, rather than attacking specific nodes. Metattack is an example of such a powerful attack algorithm. In the global attack defense experiments, we adopted Metattack to attack the graph neural network and conducted high-intensity attacks by modifying the proportion of perturbed edges in the network. The experiments were used to evaluate the performance of various defense methods based on the node classification accuracy of the model. The results of the experiments are shown in Figures 8. It is easy to observe that WLGCN and WGCN exhibit better overall robustness compared to other defense methods. It can be seen that Winsorised Conv, after design improvements, plays a major role in defending against global attacks.

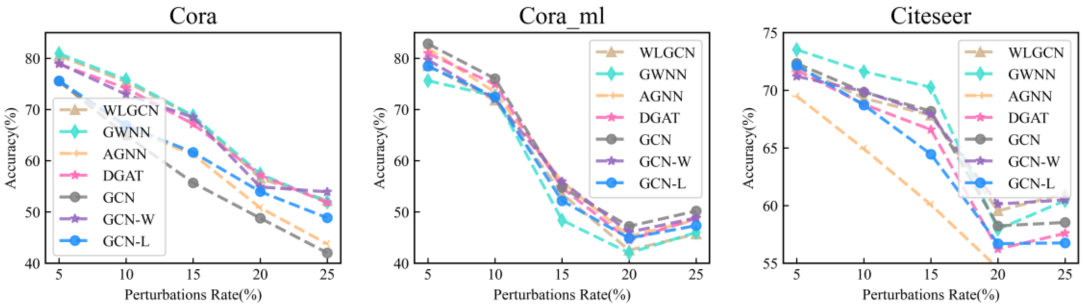

Figure 8: The accuracy curves of the models under global attack

6.5. Parameter Sensitivity Analysis

To better understand how different hyperparameters can effectively improve a model's resistance to adversarial attacks, this chapter conducts an ablation study on the WLGCN model's main parameters, which include the following three items: a) message aggregation weight parameter α (0 < α < 0.5); b) potential adversarial regularization weight parameter γ; c) potential adversarial perturbation parameter ε.

In this section, the sensitivity of the WLGCN model's hyperparameters α, γ, and ε is explored. In the experiment, one hyperparameter was fixed, and another hyperparameter was given a value while the other hyperparameters were fixed at their optimal values. The effect of different thresholds on the performance of WLGCN was studied by changing the other hyperparameters.

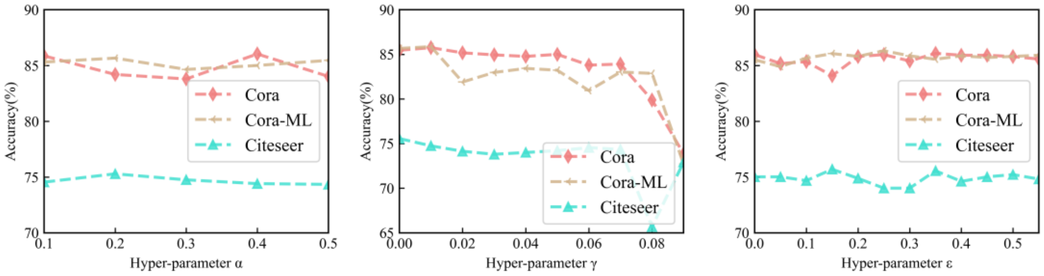

Figure 9: Parameter sensitivity analysis

Specifically, the value range of α was adjusted from 0.1 to 0.5, γ from 0 to 0.9, and ε from 0 to 0.6. This experiment takes the three datasets with a clean graph and uses the accuracy as the evaluation metric. The performance changes of the model are shown in the figure 9.

7. Conclusion

Despite achieving impressive performance, graph convolutional neural networks suffer from robustness issues. In this paper, we address the robustness issue of graph convolutional neural networks by investigating the non-robustness of aggregation functions. Inspired by the theory of breakdown point and influence function, we propose to use the more robust winsorised mean aggregation function and incorporate potential adversarial regularization into the $H^{(1)}$ layer of the message passing-based GCN. The robust combinatorial defensive method, named WLGCN, achieves improved robustness against graph attacks without sacrificing classification accuracy. We evaluate the performance of our proposed model under different perturbation costs using Nettack targeted attack and Metattack global attack methods. Extensive experiments on real datasets are conducted to evaluate the model's performance using accuracy and classification margin as evaluation metrics. We also perform parameter sensitivity analysis on the model. The experimental results demonstrate that our proposed method achieves high robustness while maintaining model accuracy.

Acknowledgement

This work was supported by Natural Science Foundation of Zhejiang Province, China (Grant Nos. LQ13F030015) and the National Natural Science Foundation of China (Grant Nos. 61304150).

References

[1] B.-B. Xu, K.-Y. Cen, J.-J Huang, et al. A review of graph convolutional neural networks. Journal of Computer Research and Development, 2020, 43(5): 755-780.

[2] Jiang Z, Gao Z, Duan Y, et al. Camouflaged Chinese spam content detection with semi-supervised generative active learning [C] //Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020: 3080-3085.

[3] Zϋgner D, Akbarnejad A, And Gϋnnemann S. Adversarial attacks on neural networks for graph data[C]// Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ser. KDD ’18. 2018: 2847C2856.

[4] Dai H, Li H, Tian T, et al. Adversarial attack on graph structured data[C]//Proceedings of the 35th International Conference on Machine Learning, ser. ICML ’18. 2018: 1123-1132.

[5] Wang X, Eaton J, Hsieh C J, et al. Attack graph convolutional networks by adding fake nodes[J]. arXiv preprint arXiv: 1810.10751, 2018.

[6] Zhou K, Michalak T P, Waniek M, et al. Attacking similarity-based link prediction in social networks[C] //Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, ser. AAMAS ’19. 2019: 305-313.

[7] Sun Y, Wang S, Tang X, et al. Node injection attacks on graphs via reinforcement learning[J]. arXiv preprint arXiv: 1909.06543, 2019.

[8] Feng F, He X, Tang J, et al. Graph adversarial training: Dynamically regularizing based on graph structure [J]. IEEE Transactions on Knowledge and Data Engineering, 2019, 33(6): 2493-2504.

[9] Zhu D, Zhang Z, Cui P, et al. Robust graph convolutional networks against adversarial attacks[C]//Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2019: 1399-1407.

[10] Zϋgner D, Gϋnnemann S. Certifiable robustness and robust training for graph convolutional networks[C]// Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2019: 246-256.

[11] Entezari N, Al-Sayouri S A, Darvishzadeh A, et al. All you need is low (rank) defending against adversarial attacks on graphs[C]//Proceedings of the 13th International Conference on Web Search and Data Mining. 2020: 169-

177.

[12] Jin M, Chang H, Zhu W, et al. Power up! Robust graph convolutional network via graph powering[C]//35th AAAI Conference on Artificial Intelligence. 2021.

[13] Chen J, Lin X, Xiong H, et al. Smoothing adversarial training for gnn [J]. IEEE Transactions on Computational Social Systems, 2020, 8(3): 618-629.

[14] Xu K, Chen H, Liu S, et al. Topology attack and defense for graph neural networks: An optimization perspective [J]. arXiv preprint arXiv:1906.04214, 2019.

[15] Zügner D, Günnemann S. Adversarial attacks on graph neural networks via meta-learning [J]. arXiv preprint arXiv: 1902.08412, 2019.

[16] Geisler S, Zügner D, Günnemann S. Reliable graph neural networks via robust aggregation [J]. Advances in Neural Information Processing Systems, 2020, 33: 13272-13284.

[17] D -Y. Zhu, Z -W. Zhang, P. Cui, and W -W. Zhu. Robust graph convolutional networks against adversarial attacks. In KDD, pages 1399–1407, 2019.

[18] J -Y. Chen, D -J. Zhang, et al. A review of graph neural network for adversarial attack and defense [J]. Journal of Network and Information Security, 2021, 7(3): 1-28.

[19] Stutz D, Hein M, Schiele B. Disentangling adversarial robustness and generalization[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 6976-6987.

[20] Zügner D, Günnemann S. Adversarial attacks on graph neural networks via meta-learning [J]. arXiv preprint arXiv:1902.08412, 2019.

[21] Zhou S J. Trimmed mean and winsorised mean [J]. Journal of Xi'an Institute of Geology, 1996, 18(4): 84-90.

[22] Hampel F R, Ronchetti E M, Rousseeuw P J, et al. Robust statistics: the approach based on influence functions [M]. John Wiley & Sons, 2011.

[23] Miyato T, Dai A M, Goodfellow I. Adversarial training methods for semi-supervised text classification [J]. arXiv preprint arXiv:1605.07725, 2016

[24] Jin H, Zhang X. Latent adversarial training of graph convolution networks[C]//ICML workshop on learning and reasoning with graph-structured representations. 2019, 2.

[25] Hampel F R, Ronchetti E M, Rousseeuw P J, et al. Robust statistics: the approach based on influence functions [M]. John Wiley & Sons, 2011.

[26] Dingyuan Zhu, Ziwei Zhang, Peng Cui, and Wenwu Zhu. Robust graph convolutional networks against adversarial attacks. In KDD, pages 1399–1407, 2019.

[27] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In ICLR, 2015.

[28] Zügner D, Akbarnejad A, And Günnemann S. Adversarial attacks on neural networks for graph data[C]// Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ser. KDD '18. 2018: 2847C2856.

[29] Jin W, Ma Y, Liu X, et al. Graph structure learning for robust graph neural networks[C]//Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining. 2020: 66-74.

[30] Xu B, Shen H, Cao Q, et al. Graph wavelet neural network [J]. arXiv preprint arXiv:1904.07785, 2019.

[31] Thekumparampil K K, Wang C, Oh S, et al. At-tention-based graph neural network for semi-supervised learning [J]. arXiv preprint arXiv:1803.03735, 2018.

[32] Hu W, Chen C, Chang Y, et al. Robust graph convolutional networks with directional graph adversarial training [J]. Applied Intelligence, 2021, 51(11): 7812-7826.

[33] Zhou J, Cui G, Hu S, et al. Graph neural networks: A review of methods and applications [J]. AI open, 2020, 1: 57-81.