Iranian Journal of Fuzzy Systems

Volume 22, Number 2, (2025), pp. 97-127

Original paper

University of Sistan and Baluchestan

Knowledge-based fuzzy defensive classification model for robust MPGNNs

E. Bastami [1], H. Soltanizadeh [2], M. Rahmanimanesh [3], P. Keshavarzi [4] and M. Asadpour [5]

[1,2,3,4] *Electrical and Computer Engineering Department, Semnan University, Semnan, Iran*
[5] *Computer Engineering Department, Tehran University, Tehran, Iran*

es.bastami@semnan.ac.ir, h_soltanizadeh@semnan.ac.ir, rahmanimanesh@semnan.ac.ir, pkeshavarzi@semnan.ac.ir, asadpour@ut.ac.ir

Abstract

Message-passing graph neural network (MPNN) is a highly effective learning tool for graph-structured data but is particularly vulnerable to adversarial attacks on information flow, where poisoned data from attacked nodes can propagate to neighboring nodes and be aggregated, amplifying the negative impact across the graph. The malicious perturbations to node features or graph structures can degrade model performance in defensive decision-making, which requires a clean flow. We propose a creative defense mechanism to robustly enhance the resilience of MPGNNs against adversarial perturbations by leveraging a creative *knowledge-based fuzzy approach* for node classification that is implemented by a rule-based fuzzy model. Our study assigns each neighboring node a membership function, which quantifies its relevance to the target node based on knowledge extracted from structural information, node attributes, and community relationships to the target node. By dynamically adjusting the influence of neighboring nodes, the proposed model mitigates the effects of adversarial noise and reduces the impact of irrelevant or poisoned data, ensuring more robust message aggregation. Furthermore, the fuzzy approach helps prevent over-squashing, a common issue when increasing the number of GNN layers, by maintaining important information through controlled message passing and is followed by picking an opinion leader agent from each egonet to recycle the graph. However, our model does not require prior knowledge of the graph or attack types, making it adaptable to various datasets and attack scenarios. Extensive experiments on benchmark datasets, including Cora, PubMed, and Citeseer, demonstrate greater robustness and better accuracy when compared to state-of-the-art defense strategies with reasonable complexity.

Keywords: Adversarial defending strategy (ADS), message-passing graph neural network (MPGNN), knowledge-based fuzzy graph neural network (KFGNN), opinion leader node (OLN), social networks, fuzzy logic method (FLM), node classification.

1 Introduction

MPGNNs are a category of neural network frameworks that can learn low-dimensional graph data embeddings, consisting of both node attributes and graph topology. MPGNNs have demonstrated effective performance across a diverse range of practical applications, such as link prediction [7], drug discovery [5], recommender systems [52], self-driving cars [43], chip placements [38], forecasting [24], and outlier detection [19]. However, MPGNNs can be susceptible to malicious perturbations that manipulate the graph structure by adding, deleting, or rewiring edges in a subtle way. Adversarial attacks are crafted to mislead machine learning algorithms by introducing subtle alterations to the input data. In deep learning, such minute alterations are typically beyond human detection, yet they can skew the model's predictive accuracy. By estimating the uncertainty of the model's predictions, it is possible to identify data points that are close to the decision boundary and, therefore, more susceptible to adversarial attacks [2]. These attacks can significantly affect the performance and reliability of MPGNNs in downstream tasks, including the classification of

Corresponding Author: H. Soltanizadeh
Received: December 2024; Revised: February 2025; Accepted: April 2025.
https://doi.org/10.22111/ijfs.2025.50384.8892

Table 1: All used Acronyms, symbols, notations, and their descriptions or standard definitions [authors].

Acronym	Description	Acronym	Description
ADS	Adversarial Defending Strategy	NFMD	Node Fuzzy Membership Degree
AGG	Aggregation	NLP	Natural Language Processing
AI	Artificial Intelligence	NCS	Node Color Spectrum
GAT	Graph Attention Network	NSD	Node Slyness Degree
GCN	Graph Convolutional Network	OLA	Opinion Leader Agent
GIN	Graph Isomorphism Network	OLN	Opinion Leader Node
GNN	Graph Neural Network	KFGNN	Knowledge-based Fuzzy GNN
GNSD	Graph Neural Stochastic Diffusion	RGCN	Robust Graph Convolutional Network
KRG	Knowledge Relational Graph	RNN	Recursive Neural Network
MGA	Meta Gradient Attack	SNA	Social Network Analysis
ML	Machine learning	SVD	Singular Value Decomposition
MPGNN	Message-passing Graph Neural Network	UGNN	Uncertainty-aware GNN
Symbol	**Definition**	**Symbol**	**Definition**
Computation Graph	An egonet	METTACK	A MGA attack type on GNNs
Egonet	Egocentric social network data	NETTACK	An attack type on GNNs
Notation	**Definition**	**Notation**	**Definition**
$A \in \mathbb{R}^{R \times R}$	Adjacency matrix	$H^k \in \mathbb{R}^{N \times F}, k \in \{1, 2, ..., k\}$	Node representation in k^{th} layer
$G(V, E)$	Graph with Vertex set & Edge set	$h_i^{l+1}, i \in \{1, 2, ..., L\}$	Feature of Node i at $(l+1)^{th}$ layer
$\hat{G}(V, E)$	Attacked graph with Vertex set & Edge set	$N(h_v)$	Neighboring set of node v
$\alpha(u, v)$	Structural distance u and v nodes	$\gamma(u, v)$	Community distance u and v nodes
$\beta(u, v)$	Feature distance u and v nodes	$X \in \mathbb{R}^{N \times C}$	Node features

Note: In our work, the abbreviations MPGNN and GNN mean the same thing.

nodes [23]. Robustness in the MPGNN community has various interpretations, such as maintaining node classification performance regardless of graph topology or features, generalizing within and across domains, and resisting adversarial attacks [12]. Therefore, understanding and enhancing the robustness of MPGNNs against adversarial modifications affecting the graph's topology is a critical and challenging problem. Furthermore, these investigations also contribute to understanding the limitations and robustness of MPGNNs, designing more reliable and interpretable MPGNNs, advancing graph learning theory, etc. For better readability of the rest of this research paper, a comprehensive list of all standard abbreviations and visible symbols used in this study, along with their descriptions and definitions, is presented in Table 1.

1.1 Message passing graph neural network and challenges

Graph neural networks are neural networks that operate on graph-structured data. GNNs have emerged as a powerful alternative to traditional models for graph analysis [26]. The development of advanced deep learning frameworks has fueled the widespread adoption of GNNs in social network analysis [6]. These models boast potential applications in diverse domains, including recommender systems, molecular graphs, and social network analysis. Their effectiveness in solving various downstream tasks has been well-established [25]. MPGNN models generate node embedding through the application of message-passing layers. These layers employ non-linear activation functions and aggregation operators to capture the structural information of each node within its receptive field [51]. The primary objective of MPGNNs is to iteratively *update* the hidden representation of a node. This task is achieved by *aggregating* the information from the neighbors of each node and *combining* their own representation from the previous iteration to update the node representations. The core operation of MPGNNs is information diffusion in the form of *message passing*, which propagates node features by exchanging information between adjacent nodes, also known as MPGNN [17]. Here, each node receives a hidden representation that is continuously updated by aggregating the embedding of its neighboring nodes [6]. Recently, a line of research has investigated enhancing the robustness of MPGNN models for event detection, which we will address in this paper as our research [56].

Information diffusion prediction studies how information items spread in the MPGNN, which either concentrates on leveraging a single diffusion sequence at a time or uses the graph relationships among neighbors and the diffusion sequence as a simple chain structure. This approach cannot fully capture the complex interactions between nodes. Additionally, the diffusion paths of all messages can construct a dynamic diffusion graph (See Figure 1) [9, 45]. In practice, MPGNNs were observed not to benefit from more than a few layers. The accepted explanation for this phenomenon is *over-smoothing*: node representations become indistinguishable when the number of layers increases. Nonetheless, over-smoothing was mostly demonstrated in short-range tasks—tasks that have small problem radii, where a node's correct prediction mostly depends on its local information. Such tasks include paper subject classification and product category classification. Since these learning problems depend mostly on short-range information, it makes sense why more layers than the problem radius might be extraneous. In contrast, in tasks that also depend on long-range

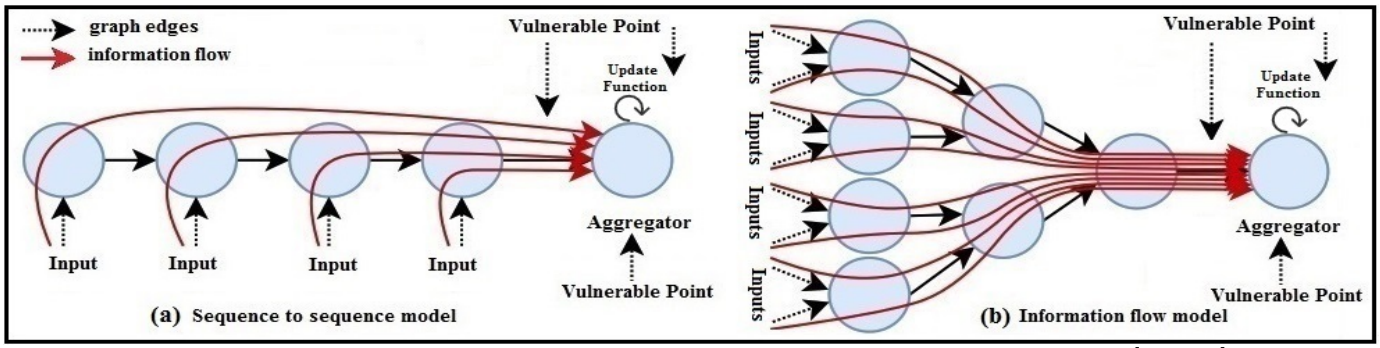


Figure 1: The overall bottlenecks of information flow in MPGNNs from a node perspective [authors].

information and thus have larger problem radii, we hypothesize that the explanation for limited performance is over-squashing. To allow a node to receive information from other nodes at a radius of K, the GNN needs to have at least K layers, or otherwise, it will suffer from under-reaching—these distant nodes will simply not be aware of each other.

Clearly, to avoid under-reaching, problems that depend on long-range interaction require as many GNN layers as the range of the interaction. However, as the number of layers increases, the number of nodes in each node's receptive field grows exponentially. This causes over-squashing, where the information from the receptive field must be compressed into fixed-length node vectors. In fact, the GNN vulnerability is analogous to the vulnerable points of sequential models (See Figure 1). Traditional sequence-to-sequence models suffered from a vulnerability at every decoder state; the model had to encapsulate the entire input sequence into a fixed-size vector, which is related to translating text from one language to another in the field of natural language processing (NLP) (See Figure 1(a)). The receptive field of a node grows linearly with the number of recursive applications, which is related to recursive neural networks (RNN) (See Figure 1(b)). However, in GNNs, the vulnerability is asymptotically more harmful because the receptive field of a node grows exponentially [4]. Despite their widespread application across various domains, MPGNNs still face five inherent challenges as follows:

- **Challenge 1: Performance Degradation-** MPGNN models often fail to accurately capture the similarity between local structures, significantly impacting their performance. This issue arises from the failure of Laplacian positional encoding to represent the structural similarity between adjacent nodes. The discrepancy in initial positional encodings for structurally similar nodes introduces false signals that hinder the accurate learning of graph structure representations. This flawed information flow from the start can propagate through the network, leading to suboptimal performance and increased vulnerability to adversarial attacks. If adversarial nodes inject misleading positional information, the entire network's ability to accurately learn and represent the graph can be compromised [16].

- **Challenge 2: K-hop Distance Destruction-** Integrating local and global structural features into a unified vector representation is a significant challenge for existing GNN models. Global features derived from node roles often diverge from local node features, attributes, or classes. This disparity complicates merging these features into a cohesive representation, which is crucial for tasks that rely on local and global information. An adversary could exploit this by targeting nodes that play key roles in global structures, injecting false information that disrupts the integration process and diminishes the model's ability to learn accurate representations [58].

- **Challenge 3: Information Over-smoothing-** Over-smoothing refers to the convergence of node representations into indistinguishable embeddings due to excessive message aggregation, leading to a loss of structural information. This challenge is particularly problematic in heterophilic graphs, where neighboring nodes often belong to different classes and exhibit diverse features [10, 18]. Several structural and computational factors contribute to over-smoothing, including large spectral gaps, which accelerate the transition to uniform embeddings, inter-class noise, which dilutes feature distinctions, imbalanced class distributions, which suppress signals from minority classes, and the absence of adaptive aggregation mechanisms in models such as GCN and GAT [18, 40]. While existing techniques such as DropEdge, which removes edges randomly to retain feature diversity, and skip connections, which preserve residual information, provide partial mitigation, they introduce trade-offs that can disrupt long-range dependency preservation [40]. The knowledge-based fuzzy graph neural network (KFGNN) model effectively mitigates over-smoothing through a fuzzy rule-based system that dynamically adjusts aggregation weights based on contextual relevance [28]. By integrating Feature Distance, Structural Distance, and Community Distance, KFGNN prevents excessive feature mixing while maintaining essential long-range information, making it a robust solution for graph classification, anomaly detection, and adversarial learning applications [10, 18].

- **Challenge 4: Information Over-squashing-** Over-squashing is a significant limitation in GNNs, occurring when message passing compresses information from distant nodes into fixed-size representations, thereby hindering the propagation of meaningful signals. This issue becomes more pronounced in deep GNNs, where the exponential growth of receptive

fields results in substantial information loss, affecting tasks such as node classification, link prediction, and anomaly detection [1, 10]. Several factors contribute to over-squashing, including graph bottlenecks, which restrict information flow in sparsely connected regions, large graph diameters, which require excessive message passing, uneven degree distributions, where hub nodes dominate feature aggregation, and static aggregation mechanisms in models like GCN and GAT, which fail to distinguish relevant from redundant information [18, 40]. The phenomenon has been formally analyzed using the Jacobian sensitivity matrix, demonstrating that a node's representation exponentially loses its ability to capture distant node features as the number of layers increases [47]. Additionally, effective resistance between node pairs and commute time analysis further quantify over-squashing effects, confirming that weak connectivity exacerbates information loss [46]. Traditional solutions, such as graph rewiring and edge augmentation, attempt to mitigate over-squashing but risk introducing over-smoothing due to excessive feature mixing [18, 40]. The proposed KFGNN addresses this issue by dynamically regulating message passing through Feature Distance, Structural Distance, and Community Distance, ensuring that long-range dependencies are propagated effectively while preserving structural integrity [18, 40].

- **Challenge 5: Aggregation Defects-** MPGNNs, including graph convolutional networks (GCNs) and graph attention networks (GATs), primarily rely on neighboring features to create node embedding. This limitation restricts their ability to capture long-range dependencies and global structural information. As a result, the network may fail to recognize critical patterns and roles that span broader graph structures. Adversarial attacks exploiting this limitation can manipulate distant nodes, disrupting the global information flow and leading to incorrect predictions. Effective management of global information flow is essential to ensure the model can resist such attacks and maintain robust performance [49].

Understanding and managing the flow of information in GNNs is crucial for addressing the challenges of inaccurate local structural similarity capture, limited global structural information, difficulty integrating local and global features, over-smoothing, and over-squashing. These challenges not only impact model performance but also introduce significant vulnerabilities, as adversarial attacks can exploit these weaknesses by manipulating key nodes or distorting information propagation within the network. If several nodes along the information flow path are compromised, the entire flow can be contaminated, leading to widespread misinformation and degraded performance [40]. The trade-off between over-squashing and over-smoothing poses a fundamental challenge in GNNs, as solutions that alleviate one often exacerbate the other [18, 28]. Over-squashing limits the propagation of long-range dependencies, preventing effective message passing between distant nodes, while over-smoothing leads to uniform embeddings, reducing the network's ability to capture meaningful structural distinctions [18]. These issues are particularly detrimental in tasks requiring fine-grained node representations, such as node classification, anomaly detection, and link prediction [46]. Addressing these competing phenomena is essential for developing more resilient and effective graph-based learning models [18, 28].

To mitigate these challenges, the KFGNN framework introduces a novel adaptive mechanism that balances information flow dynamically. By leveraging fuzzy membership functions and knowledge extraction techniques, KFGNN optimizes the trade-off between over-squashing and over-smoothing, ensuring that message passing is neither excessively compressed nor overly homogenized [18, 40]. The model refines spectral properties during graph diffusion, adjusting connectivity to prevent bottlenecks while maintaining diversity in node embeddings [46]. Additionally, graph repair mechanisms selectively refine the network structure to enhance robustness against adversarial perturbations, preventing excessive information loss or oversaturation [46]. Empirical results demonstrate that KFGNN achieves enhanced node classification accuracy and improved resilience against adversarial attacks, effectively striking an optimal trade-off between over-squashing and over-smoothing.

The challenges of managing information flow in MPGNNs not only impact performance but also introduce significant vulnerabilities. Adversarial attacks can exploit these vulnerabilities by targeting key nodes or manipulating the information flow within the network. For example, if several nodes in the information flow path are compromised, the entire flow can be contaminated, leading to widespread misinformation and degraded performance. Understanding and managing the flow of information in GNNs is crucial for addressing the challenges of inaccurate local structural similarity capture, limited global structural information, difficulty integrating local and global features, over-smoothing, and over-squashing. By focusing on these aspects, we can enhance the robustness and accuracy of MPGNNs, making them more resilient to adversarial attacks.

Effective information flow management involves improving encoding techniques, revising aggregation mechanisms, balancing network depth, and ensuring comprehensive propagation of long-range information. This adversarial approach will lead to the development of more secure and efficient MPGNN models capable of performing reliably across various tasks and resisting adversarial manipulations. We focus on MPGNN architectures that can be interpreted as over-smoothing of information flow on graphs and over-squashing of the information by using aggregators effects for properly designing a robust defense algorithm, and accurately presenting a specific knowledge-based fuzzy scoring for a clean information flow of modern MPGNN classification to overcome information over-smoothing and over-squashing, as well as the trade-off between them. This paper aims to delve deeply into these information flow issues by constraining *K-hop distance* to study *information over-smoothing*, *information over-squashing*, and *aggregation defects* and provide insights

and solutions to mitigate these vulnerable points and enhance the overall *performance* and efficacy of GNNs in complex real-world applications.

1.2 Adversarial defending strategy and challenges

The direct challenges experienced by standard GNN models in the specific context of adversarial defense realistically represent local sensitivity to excellent choices of hyperparameters, sparsity of social graphs, and a considerable number of perturbed modifications. The Pro-GNN is typically designed to sufficiently restore the original graph structure using low-rank and sparsity constraints [22]. This not only allows for the active learning of clean graph structures but also ensures the robustness of MPGNNs. However, one principal drawback of Pro-GNN correctly represents the sensitivity to hyperparameter settings. Local sensitivity can negatively affect the model's performance and overall robustness, making it difficult to identify optimal hyperparameters. Graphs are typically sparse and discrete, and the representation of any node depends on its neighborhood. For this reason, we find defenses are sometimes sensitive to a common attack vector, and transferring attacks can also be successful. RGCN uses a Gaussian distribution to represent hidden layer outputs in GCNs [61]. The developed model is capable of defending against both target and non-target attacks. When there are too many perturbed edges, its defensive performance is poor.

The inflexibility of the model shows that the existing methods are often designed for a specific type of attack. Using fuzzy modeling and membership values gives the MPGNN model more necessary flexibility to be resistant to various types of adversarial attacks [31]. Complex dependencies and direct correlations: Current work may not account for the complex dependencies and correlations among the graph nodes and edges, which can affect the quality of the posterior distribution and the predictions' robustness. The relationship between nodes and edges in real networks is often non-linear and uninterpretable. Fuzzy modeling with membership value provides practical tools to explore and interpret these complex relationships. In the realm of defending MPGNNs against adversarial attacks, various methods have been developed. However, these methods face several significant challenges and limitations. Here are multiple primary challenges associated with current defense mechanisms, including the inability to properly model the presence and impact of nodes within the acceptance field and manage information flow under adversarial attacks:

- **Challenge 1: Limited Generalizability-** Most existing defense methods are designed to counter specific types of adversarial attacks and may not perform well against a wide variety of attacks. For instance, a method effective against feature-based attacks might not be as effective against structural perturbations. This limitation in generalizability means that a single defensive approach may not comprehensively protect MPGNNs from all potential adversarial threats [59].

- **Challenge 2: High Computational Complexity-** Many defensive techniques impose substantial computational overhead. Techniques such as adversarial training and data augmentation often require significant additional computational resources and extended training times [59]. This increased complexity can limit the practical applicability of these defenses in real-world scenarios, particularly in large-scale or resource-constrained environments [15].

- **Challenge 3: Impact on Model Performance-** Implementing defense mechanisms can adversely affect the performance of GNNs on non-adversarial tasks. For example, adversarial training might reduce the overall accuracy of the model on clean data. Balancing robustness against adversarial attacks with maintaining high performance on standard tasks is a persistent challenge [50].

- **Challenge 4: Ineffective Modeling and Management-** Current methods often struggle to accurately model the presence and influence of nodes within the acceptance field, particularly under adversarial conditions. When an adversarial attack occurs, the information flow within the graph can be severely disrupted, leading to significant degradation in performance. Existing defenses frequently fail to manage this disrupted information flow effectively, resulting in poor resilience to node poisoning and other perturbation-based attacks. This limitation is critical as it affects the MPGNN's ability to maintain robust performance when faced with adversarial manipulations that target the core structure and communication pathways within the network [54].

- **Challenge 5: Weakness on Defensive Decision Making-** The success of the correct decision-making task is mainly related to the elimination of the key challenges of information flow by using a clean flow. MPGNN's fundamental weakness in defensive decision-making lies in key decisions allegedly based on valuable information that is required to efficiently utilize the global history of system structure and organizational behavior on the information flow path, which requires accounting for attack effects in the decision equations. We need *under-attack* system behavior gathering and analysis as transient states information, along with the system history information on the attacked graph for real-time decisions. To progressively improve the desired effects of the standard usage of structural and behavioral information, we should profitably employ knowledge-based reasoning in the possible use of system-recorded history and analyze online local behavior regarding attack results by new information. The success of this important task belongs to the elimination of the key challenges of information flow that the knowledge graph is applied to solve, ensuring a clean flow. Online management needs to use new

information that changes node history for behavior and features and requires a powerful knowledge of transient changes as a main challenge to reduce uncertainty states for risk factor acceptance of this *noisy behavior* [29].

1.3 Knowledge-based fuzzy decision-making

Machine learning (ML) techniques play a major role in the successful and accurate classification task of nodes. Graph neural network tools, despite their remarkable success, traditionally produce stochastic estimates of their accurate predictions due to decision-making in uncertain spaces. This remains underexplored, naturally limiting their practical applications, especially in risky areas. Current works suffer severely from either intractable posteriors or inflexible prior specifications, leading to suboptimal empirical results. However, studies have shown these models are vulnerable to noise and adversarial attacks. Many studies have focused on removing noise and adversarial effects in image and text processing domains as safe modes [55]. However, learning robust models in graph domains is more challenging because adding *noise* or *perturbation* to *graph data* makes enhancing robustness even more difficult. The noise and perturbation of edge and node attributes easily *propagate* to neighboring nodes through relational information in the MPGNN's message-passing operation, requiring certain decision-making [33, 57]. Destructive analysis of these deleterious effects in uncertain conditions of a direct attack can accurately establish a hypocritical role for the same node among various egonets.

To accurately eliminate the effects of an attack and ensure model safety, it is crucial for models to assess risk and make *uncertainty-aware* decisions [11]. This naturally generates calculation errors, resulting in incorrect predictions and time-consuming overhead. This subject is an important research domain for accurately defining considerable uncertainty in various scientific models of modern applications [30, 34]. We explained that the core operation of message-passing in MPGNNs by constraining K-hop distance is a main challenge of *information flow* for reliable decision-making. This *data flow* in all MPGNN models can radically change the data over the main vulnerable points in the used form of information over-smoothing, information over-squashing, and information aggregation defects. Also, an attack can merely change the static conditions to an uncertain environment for *deterministic* decision-making [21]. To overcome these challenges and solve the stochastic decision problem, proposing a safe adversarial defending strategy can encounter many critical problems, such as limitations in generalizability, considerable computational complexity, and low performance, which can fail to manage the disrupted information effectively, resulting in poor robustness to node poisoning and other perturbation-based attacks.

The attacked graph can lead inexorably to all certain predictions of the decision-making phase becoming uncertainty estimations in a risky area. By carefully exploring key nodes with hypocritical behavior, we must willingly find opinion leader nodes (OLNs) and allow the active role of the modifier as an opinion leader agent (OLA) for the proper self-repair of an egonet to instantly modify the uncertain conditions to certain favorable states [3, 8, 39]. We genuinely need to find all OLNs based on a precise knowledge and instantly swap the active role of each OLN to an OLA in an egonet for a given attack in online networks; reliably detect the essential topics of discussion in a given domain; and reasonably calculate a competency and popularity score for each local node in the given domain [41]. It then calculates a *knowledge-based* probability function for defining the role of an opinion leader node in that domain by using competency and popularity scores and finally ranks the egonet nodes of the graph based on their probability of being in an opinion leader role.

Based on these discussions, our problem definition efficiently is "*the reliable information flow of complex MPGNN issues by constraining K-hop distance to correctly solve information over-smoothing, information over-squashing, and aggregation defects in a possibly under-attack egonet provides insights and creative solutions for mitigating all vulnerable points and uncertainty conditions to positively enhance the global performance of MPGNNs in deterministic knowledge-based decision-making of complex real-world networks by constructing a knowledge relational graph and finding all hypocrite nodes with hypocritical roles and accurately defining a leader opinion agent to repair the egonet from effective attacks to thoughtfully delete anomalous node behavior by removing their hypocritical roles for node classification.*"

1.4 Knowledge extraction

As a methodological limitation, while we put into the hands of local practitioners a global attack method aware of regional or local constraints, there is no logical strategy for accurately determining the optimal neighboring budget as an open problem [20]. Indeed, these naturally have to be set application-dependent by domain experts for correct data gathering, better information processing, and useful knowledge extraction. We leverage local knowledge about the attack effect on all node features of an egonet to eliminate data-flow harmful events in information flow to determine budgets that correspond to semantic-preserving perturbations. However, in most real-world applications, there is a notable lack of superior knowledge about the data-generating distribution. As a practical result, we additionally use the different interpretability aspects of robust diffusion to collect other direct evidence and key insights into the learned message

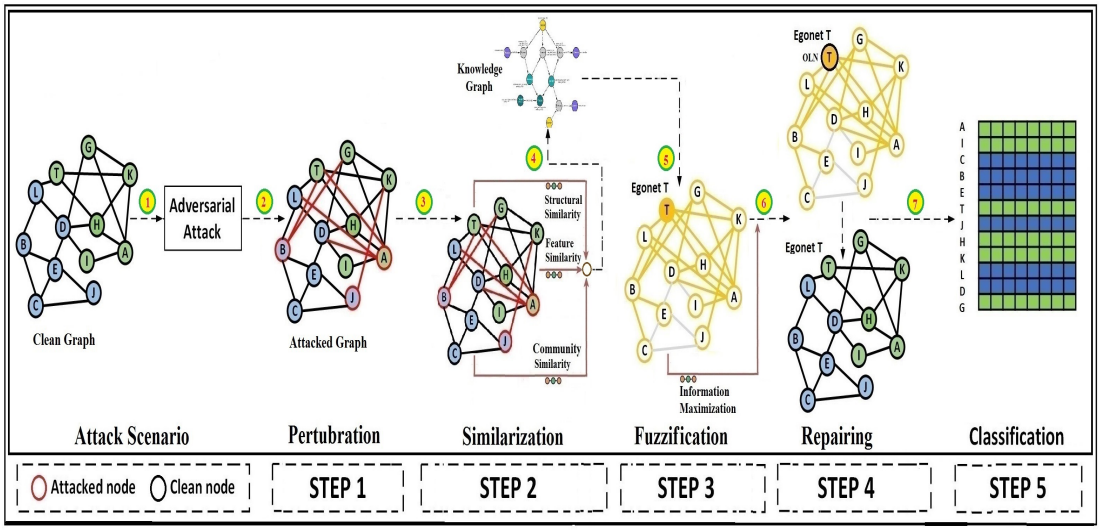


Figure 2: Conceptual modeling of our knowledge-based fuzzy graph neural network defensive model (KFGNN) [authors].

passing. In this work, for attacked time management, our focus is on evasion attacks on the features of nodes in an egonet structure. We carefully extract our needed intelligence from the attacked graph in three forms of present information: *structure*, *feature*, and *community* (See Figure 2). This historical awareness is extracted efficiently by utilizing the key concepts of *distance* established on structure, features, and community information from old graph history that profitably employs a pre-constructed knowledge relation graph based on the current state for result improvement.

We aim to carefully provide accurate predictions for globally attacking key nodes with realistic local constraints by using extracted knowledge from the graph in the specific form of a knowledge relation graph and adding useful, applicable knowledge from node features and the creative community of the attacked graph to properly enforce unnoticeability. We describe our knowledge-based problem definition and design an adversarial defense strategy to robustly detect and eliminate attack effects on data flow. This is done by using the knowledge relational graph in a simple example to understand the steps of the knowledge-aware approach by finding the used metrics toward a practical application. The knowledge relational graph structure is created by defining a *semantic graph* or *ontology graph* for ontological concepts and constructing an example knowledge graph or *instance graph* for instance concepts related to one attack scenario of an egonet, which is a union of a node and all its neighbors. The detailed method of constructing all used knowledge in KFGNN will be described in Section 3. The data vulnerability in the message-passing process of MPGNN in attack scenarios remains a critical challenge, and the illegal flow must be professionally cleaned of noisy information for deterministic decision-making. Practical results improve instantly by typically using the system's history to carefully extract practical knowledge from previous states and the currently extracted attack state for structural and feature similarities by utilizing community organizations.

1.5 Proposed KFGNN approach and evaluation overview

We propose an knowledge-based fuzzy graph neural network defensive model for knowledge-based node classification, designed specifically to enhance the robustness of MPGNNs against poisoning attacks (See Figure 2). The KFGNN model introduces a novel approach by fuzzy model into a latent space, enabling the detection of malicious actors through similarity comparisons. This defense method is broadly applicable and effectively addresses various types of poisoning attacks that can occur in graph-based systems. At the core of KFGNN is the application of fuzzy modeling to graph-based parameters, which allows the model to assess the degree of normality or abnormality for each node in relation to a target node. Instead of relying on binary classification for present or absent, the model assigns a membership value between 0 and 1 to each node within the computation graphs. This membership value indicates the likelihood that a node is anomalous, with higher values suggesting greater abnormality.

The conceptual pipeline model of KFGNN, as illustrated in Figure 2, comprises six main steps, each designed to ensure the robust detection and mitigation of adversarial threats. These steps represent the flow of information through the model, beginning with the perturbation of a clean graph to simulate attacks, finding structural or feature similarities, identifying anomalies, and culminating in the accurate classification of nodes based on their fuzzified embeddings. Our proposed KFGNN represents a significant advancement in the field of GNNs, offering a powerful tool for safeguarding against adversarial attacks while maintaining high classification accuracy. The overall core steps of the KFGNN approach are presented below:

- **STEP 1: Perturbation -** The first step initiates the methodology by systematically subjecting a clean graph to adversarial perturbations to simulate attack scenarios. This process involves deliberately altering the graph's structure, such as adding or removing edges, to affect node features and introduce potential vulnerabilities. The primary goal is to create a modified graph that accurately represents various adversarial attacks, including those modeled by Nettack and Mettack [62, 63]. By doing so, we lay a solid foundation for developing robust detection methods and implementing effective defense mechanisms.

- **STEP 2: Similiarization -** Following the perturbation, the next step involves calculating the similarity between the target node and other nodes within the computation graph to derive a fuzzy score by using an online extracted *knowledge graph* (KG). This fuzzy score is crucial for assessing the normality or anomaly of each node relative to the target node. Similarity using the KG is accurately computed across three specific dimensions: *feature distance* as *knowledge relational graph* (KRG), and *structural distance* and *community distance* parameters. Structural distance evaluates the roles and shared connections between nodes, feature distance captures the co-occurrence of attributes among nodes, and community distance measures the likelihood of a node being anomalous based on its community affiliations. By integrating these measures, we generate a comprehensive fuzzy score that informs the subsequent steps in refining the classification and enhancing the robustness of the MPGNN.

- **STEP 3: Fuzzification -** With the similarity measures in place, the third step focuses on refining the graph representation through fuzzification by using the constructed *knowledge graph*. Unlike traditional methods where node membership is binary, after attack detection, this essential task, by changing the role of the key node to OLN, introduces a more nuanced approach by assigning each node a fuzzy *membership value*. This observed value properly reflects the degree to which a local node belongs to a specific computation graph relative to the target node. Fuzzification objectively evaluates multiple factors — *structural roles*, *feature similarities*, and *community attributes* — to correctly determine the *membership degree*, allowing the specific model to carefully handle uncertainties and vagueness inherent in the attacked graph by OLN of each *egonet*. This key step transitions the MPGNN from a rigid, deterministic framework to one that is more adaptable and robust by *information maximization*, particularly in the face of adversarial attacks or other graph perturbations.

- **STEP 4: Repairing -** After precisely calculating the fuzzy scores, the next critical repair task of the OLN is efficiently cleaning the message-passing stream of the GNN to retrieve the exact data from various local levels of the information flow. This step involves adjusting the *computation graph* or *egonet* to better reflect the uncertainties and variabilities identified through the fuzzy membership functions. In a traditional hard GNN, nodes are strictly assigned to computation graphs based on their proximity to the target node. However, in the fuzzy MPGNN approach, the rigidity is relaxed, and nodes are assigned real-valued membership degrees that provide a more nuanced representation of their association with the target node. By incorporating these fuzzy membership values into the message-passing mechanism, the MPGNN can better accommodate and correct potential distortions caused by adversarial attacks, resulting in a more robust and accurate information flow. This *repair process* is essential to enhance the network's reliability, ensuring stable performance even in complex and uncertain environments.

- **STEP 5: Classification -** In the ultimate step, the model leverages the refined node embeddings and repaired data from information flow to classify each node accurately. The *classification* process is guided by a task-specific *loss function*, such as cross-entropy loss, commonly used in semi-supervised node classification tasks. The developed model is trained end-to-end, minimizing the notable difference between predicted and actual labels through *iterative optimization*. By properly applying a *softmax function* to the hidden representations in the final GNN layer, the model generates *class probabilities* for local nodes. The classification step is crucial for completing the processing pipeline, ensuring that the MPGNN can reliably identify the correct labels for local nodes, even in the active presence of adversarial attacks or other uncertainties within the social graph.

In summary, our proposed KFGNN method has been thoroughly evaluated using four distinct defense strategies against worst-case scenarios involving three types of adversarial attacks: NETTACK-Di, NETTACK-In, and METTACK. These strategies were tested on various realistic datasets, resulting in two primary scenarios: a best-case scenario, where only graph properties are altered, and a worst-case scenario, where the logical topology undergoes significant changes, such as node additions or deletions and link insertions or disruptions. However, in our KFGNN approach, these topological changes are clearly detectable, and their potential impacts can be effectively mitigated with precision. Our defense strategy emphasizes the prevention of feature-related attacks and the rectification of structural vulnerabilities within the graph, such as the deletion of local nodes or the disruption of invalid links in the worst-case scenario. A key aspect of our approach is focusing on the attacked node within an egonet, treating it as a critical or urgent node, and progressively neutralizing the adverse effects of attacks. We carefully address the uncertain conditions affecting critical nodes within the egonet, thereby avoiding stochastic decision-making pitfalls. By identifying and addressing urgent nodes, we define the self-healing role of these key nodes as "leader opinion nodes," responsible for autonomously repairing and recovering local nodes from the effects of attacks. This process ensures deterministic decision-making by removing risk conditions during the fuzzification phase.

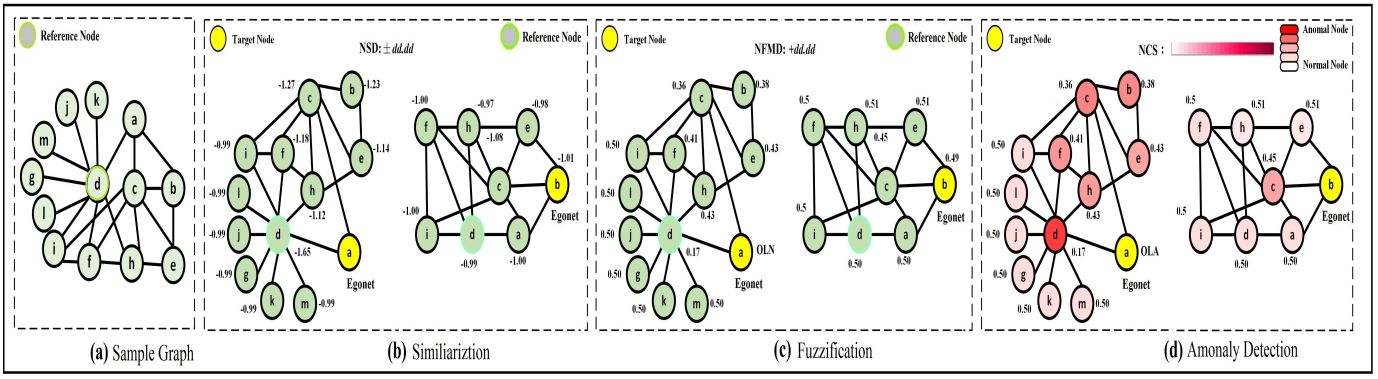


Figure 3: Our observation of different behaviors of a node in various egonets for anomaly detection [authors].

Our evaluation results demonstrate that the proposed KFGNN method significantly enhances detection accuracy, achieving, for instance on different datasets, Cora with 82.76%, PubMed with 85.23%, and Citeseer with 73.74% improvement by removing poisoned data, even in the absence of prior knowledge regarding the specific types of attacks or GNN algorithms used, with complexity $O(KV^2)$ where K represents the number of network layers and V represents all nodes of the graph. Furthermore, technical experiments on real-world graphs show that our proposed method outperforms state-of-the-art defense mechanisms, even when the graph is subjected to perturbations, proving its robustness against various attacks. These findings highlight KFGNN's capability to effectively identify and filter out anomalous nodes, leading to more reliable and trustworthy graph representation learning through automated self-healing and repair processes. Additionally, we have objectively analyzed and compared the robustness and complexity of KFGNN with emerging defense methods, confirming its practical applicability across various scenarios, with superior accuracy, performance, and scalability. Ultimately, our modern, scalable defense technique significantly improves detection accuracy and offers greater robustness against the natural noise inherent in environmental graphs, the artificially generated noise from complex computations, and a wide range of adversarial attacks.

1.6 Motivation and innovation

We carefully explain that the destructive analysis of uncertain conditions of a direct attack can accurately establish a hypocritical position for the same node among various egonets. In addition, our defense method needs to explore these key agents with hypocritical behavior to willingly allow the active role of the modifier as an opinion leader agent (OLA) for the self-repair of an egonet to instantly modify the uncertain conditions to certain. Ultimately, the role of each OLA must change to an opinion leader in an egonet for the self-repairing of each egonet. During our observations of studying MPGNNs for designing adversarial defense mechanisms, key nodes in a computational graphs exhibit very different behaviors (See Figure 3). In this effective method, all egonets or computational graphs with considerable complexity are found recursively, which sufficiently demonstrates the organizational behavior of local nodes for a sample graph and only two egonets in the notable figure. We typically show the sample graph in Figure 3(a), the inconsistent behavior of node d in two separate a and b egonets of Figure 3(b) in the similarization task, the membership degree valuation for anomalous nodes identification of Figure 3(c) in the fuzzification task, and the anomalous node degree valuation for OLA identification of Figure 3(d) in the repair task. It was an anomalous node and vulnerable to adversarial attacks, detected by correctly assigning the *node slyness degree* (NSD), which is shown by $\pm dd.dd$ value to the current quality of the critical node to define the membership degree in the fuzzyfication task.

This considerable disparity highlights a significant limitation of traditional GNNs based on binary classification as *our motivation*. The observation prompted us to develop a more refined approach to node representation, leading to the design of our uncertainty-aware fuzzy-based GNN. Next, we properly design a KRG and assign relational value to each local node of an egonet to find the degree of similarity of all local nodes to the opinion leader key node of the egonet (See Figure 3(b)). By assigning these values, we find all vulnerable local nodes of an egonet and its vulnerable degree as a fuzzy member (See Figure 3(c)). By incorporating fuzzy logic in Figure 3(c), we assign each node a membership degree as *node fuzzy membership degree* (NFMD) is shown by $0 \leq dd.dd \leq 1$ value that reflects its varying presence across different egonets to show the *node color spectrum* (NCS) from sneaky to smart in the repair task of Figure 3(d). The used method accurately captures the dual nature of local nodes, allowing node d in egonet b to be normal in one specific context and anomalous in another, as presented in Figure 3(d). Node b in egonet a is accurately detected as an anomalous node, and node a of egonet a is carefully chosen for OLA role-playing. However, in egonet b, it is accurately identified as a normal node and is unselected for OLA role-playing. We naturally need to mitigate the negative impact of adversarial nodes by sufficiently reducing uncertainty in the propagation of misleading information. This effective

strategy must meaningfully address key issues in the data flow and typically provide a scalable and generalizable solution for robust GNNs. This innovative approach must instantly improve the active management of necessary information under adversarial conditions and positively enhance the considerable accuracy and robustness of GNNs in complex real-world applications.

Our investigation into MPGNNs and adversarial defense strategies revealed several critical challenges and motivated the design of a novel approach to positively enhance the remarkable robustness and outstanding performance of GNNs. The results showed that nodes can possess varying degrees of influence, underscoring the need for a more nuanced approach to node representation. We focused on delving deeply into the reliable data flow of MPGNN issues by constraining K-hop distance to correctly solve information over-smoothing, information over-squashing, and aggregation defects in an under-attack egonet provides insights and creative solutions for mitigating all vulnerable points and uncertainty conditions to positively enhance the global performance of MPGNNs in deterministic decision-making in complex social networks. The defensive mode is developed by finding a hypocritical node in an egonet and typically changing its active role to an opinion leader agent broker for carefully negotiating with other agents to repair its egonet. Inspired by fuzzy sets, which adeptly handle the uncertainty and vagueness inherent in graph data, we propose an uncertainty-aware fuzzy-based graph neural network approach as KFGNN method. Instead of categorizing nodes as merely present or absent in the receptive field for a target node, we advocate for a fuzzy perspective, where nodes have varying degrees of presence. The fuzzy approach aims to mitigate the impact of adversarial nodes by reducing the uncertainty with which misleading information propagates through the network.

Our proposed KFGNN approach not only leverages feature information to guide model robustness but also harnesses fuzzy graph perspectives for improved generalization. By treating node presence in the acceptance field as a fuzzy membership degree, we aim to enhance the network's ability to manage information flow under adversarial conditions, ensuring more accurate and distinct node representations. This innovative strategy promises to deliver more secure and efficient MPGNN models capable of performing reliably across various tasks and resisting adversarial attacks. By meaningfully addressing the core issues of information flow management and properly integrating a fuzzy logic framework, our approach aims to positively enhance the robustness, acceptable accuracy, reasonable scalability, and generalizability of MPGNNs in complex real-world applications by carefully selecting self-repairing roles for all OLAs concurrently. In this creative knowledge-based work, we tentatively propose a fuzzy method for reliably detecting anomalous nodes based on constructing a *knowledge graph* and a key algorithm to carefully define OLNs, as well as a repair-aware model for progressively eliminating the negative effects of adversarial attacks in an egonet by self-repairing tasks. These continuous innovations are significant to the chosen field of graph neural networks and will naturally inspire further research in this key area. *Our innovations in this creative work briefly include carefully constructing a static knowledge graph, properly designing a robust defense algorithm, and accurately presenting a specific knowledge-based fuzzy scoring for a clean information flow of modern MPGNN classification to overcome information over-smoothing and over-squashing, as well as the trade-off between them.*

1.7 Case-study and paper organization

Social networks, with their intricate tapestry of interconnected individuals, provide a rich domain for analysis using MPGNNs. These networks are excellent at capturing the complex dynamics of social interactions, enabling researchers to gain valuable insights into various aspects of social behavior. MPGNNs are particularly adept at modeling phenomena such as *information diffusion*, *community detection*, and *link prediction* [27, 35, 42]. By effectively propagating information across the network and learning from interactions between individuals, MPGNNs can identify influential individuals, understand opinion dynamics, and predict collective behavior. These capabilities make GNNs an invaluable tool for analyzing social networks in various domains, including social media, public health, and marketing. For example, GNNs can be used to track the spread of misinformation on social media platforms, identify high-risk individuals in *public health networks*, or optimize *marketing campaigns* by targeting specific user groups [48].
After the notable descriptions and valuable expressions of Section 1, the remaining organizational structure of this paper is as follows. First, the main and state-of-the-art works that are related to our approach are described in Section 2. Second, Section 3 details the proposed KFGNN method. Third, Section 4 shows the experimental results of our method used in the proposed KFGNN method. Fourth, the discussion of experimental results is presented in Section 5. Finally, Section 6 states the conclusion and future work.

2 Related works

We have already stated the important works and their history in Section 1, and in this section we compare our KFGNN defense model to the state-of-the-art defense UGNN methods under adversarial attacks. Adversarial attacks

on GNNs have led to the development of several defensive mechanisms and alternative GNN architectures to enhance robustness. These methods include a variety of approaches, such as GCN [25], GAT [49], GCN-Jaccard [53], GCN-SVD [14], RGCN [61], Pro-GNN [22], GNNGuard, and Soft Medoid, each addressing different vulnerabilities within GNNs. Graph convolutional network (GCN) is a foundational architecture in GNNs that generalizes convolutional operations to graph-structured data [25]. It aggregates features from a node's neighbors to learn a node's representation, allowing the model to capture the structural and feature-based relationships within the graph. While GCN effectively captures local structure, it assumes that connected nodes are similar due to homophily, which can lead to suboptimal performance on heterophilic graphs. Furthermore, GCN needs to adequately manage the flow of information from neighbors, particularly when considering whether this information is anomalous or normal relative to the local neighborhood. The message-passing function applies uniform weighting to information from neighbors and the target node without considering the context of the neighborhood, which can result in the propagation of noise or adversarial perturbations through the graph [53].

Graph attention network (GAT) introduces attention mechanisms to GNNs, allowing the model to learn the importance of each neighbor when aggregating features [49]. This adaptability enables GAT to focus more on relevant nodes, improving the representation learning process. Despite its flexibility, GAT's attention mechanism may not always accurately distinguish between normal and anomalous neighbors, especially in complex neighborhoods. The method focuses on individual node-to-node relationships but does not account for the overall structure or community associations within the neighborhood. This can lead to incorrect assessments of node importance, particularly in adversarial settings where the structural context is crucial [14]. GCN-Jaccard is a defense mechanism that enhances GCN robustness by removing edges with low Jaccard similarity before model training [53]. This preprocessing step eliminates potentially adversarial connections that could distort the model's learning process. Although GCN-Jaccard improves robustness, it relies heavily on the assumption that adversarial edges have low feature similarity, which might not always be true. Moreover, this method needs to dynamically assess the relevance of neighbor information during the message-passing process, potentially leading to the exclusion of critical connections that could aid in defending against attacks GCN [25].

GCN-SVD strengthens GCN by applying singular value decomposition (SVD) to the adjacency matrix, focusing on its low-rank approximation [14]. This method reduces the impact of high-rank adversarial perturbations, which are often introduced to mislead the GNN. GCN-SVD, while effective in noise reduction, does not consider the inherent associations between node features and their community structures. This oversight can lead to information loss, especially in scenarios where the community context within a neighborhood is critical for accurate node classification. Robust graph convolutional network (RGCN) or (Robust GCN) is designed to defend GCNs against adversarial attacks by modeling each node's representation as a Gaussian distribution [61]. Nodes with high variance in this distribution, which indicates higher uncertainty, are down-weighted or removed from the aggregation process to prevent their influence from degrading the model's performance. While RGCN effectively mitigates the impact of noisy or adversarial nodes, it primarily focuses on individual node uncertainty without fully addressing how these uncertainties might propagate through the graph. Additionally, RGCN does not adequately account for the community structure or the relationships between node features within the local neighborhood, which could result in suboptimal information flow management.

Pro-GNN enhances GNN robustness by simultaneously learning the graph structure and model parameters, maintaining a low-rank and sparse adjacency matrix that aligns with real-world graph characteristics [22]. This approach focuses on preserving essential graph properties while defending against adversarial attacks. Despite its effectiveness, Pro-GNN can be limited by its reliance on graph sparsity and rank assumptions, which may only hold in some real-world scenarios. Furthermore, it needs to fully consider the dynamic interactions between nodes within a neighborhood, particularly in managing the information flow based on the context of the local neighborhood or community. GNNGuard offers a defense mechanism by estimating the importance of each edge based on node similarity and then controlling the message-passing process to mitigate the impact of adversarial edges [53]. While effective in addressing pairwise relationships, GNNGuard needs to fully consider the broader issue of managing information flow across the entire graph. It lacks mechanisms to assess whether the information from neighbors is contextually relevant or anomalous relative to the local neighborhood, which is crucial for maintaining robust performance in adversarial environments. Soft Medoid provides robustness against adversarial attacks by using a medoid-based aggregation function that is less sensitive to outliers, often introduced by adversarial perturbations [15]. This method, while improving stability, does not dynamically adjust to changes in information flow within the graph. It needs to assess the community structure and the relationships between node features within the neighborhood, potentially limiting its effectiveness in complex networks where such context is critical for accurate classification.

Although these methods have significantly improved the robustness of GNNs against adversarial attacks, they generally focus on localized improvements or specific types of adversarial scenarios. They often neglect the broader issue of managing *information flow* across the graph, particularly within dynamic and complex neighborhoods. Additionally,

they do not adequately consider the relationships between *node features* and the *community structure* within each neighborhood. This is crucial for distinguishing between *normal* and *anomalous nodes* during message-passing. We propose a novel method that directly addresses the challenge of information flow management within GNNs. By focusing on *observational local neighborhoods* and using a *knowledge graph* to assess node normalcy or anomaly, our approach effectively mitigates the shortcomings of previous methods, offering enhanced *robustness* against adversarial influences. These unclean flows can have negative effects on messaging, such as over-smoothing and over-squashing of used information along possible paths that typically cannot avoid the attack effects and are unable to reduce the considerable degree of uncertainty of valuable information in the egonet for deterministic pseudo-label selection.

In our definition, information flow correctly describes the intellectual movement of an *active message* substance from specific regions of higher to lower concentration. The active message in computing is effectively a messaging object capable of performing processing on its own messages available for positive receiving and correct processing. One line of our work aims to develop online knowledge-based K-hop message-passing GNNs, where node representation is updated by aggregating information from not only observational local neighborhoods but also all neighbors within K hops of the node. Additionally, our work leverages community information to enhance expressive power, which has been proven to be strictly more powerful than older works. Our essential contribution is a broad modern class of GNNs based on the discretized information flow stream, controlling the data flow on graphs and studying different information schemes for their solution. Second, we thoughtfully provide stability conditions for these flow schemes on different neighboring K-hop and additional parameters to correctly solve over-smoothing and over-squashing of valuable information, to forcefully withdraw aggregation defects by finding all vulnerable points in an egonet, guessing the value of subsequent data destruction, and properly eliminating these adverse effects on message flow. Ultimately, based on our model, a linear data flow architecture develops that performs competitively on many popular benchmark datasets to provide better performance on different data paths in an egonet by knowledge-based finding an OLA and self-repairing the damaged subgraph, whether externally expected to arise from adversarial attacks or internally due to unexpected self-destruction.

Although these methods have significantly improved the robustness of GNNs against adversarial attacks, they generally focus on localized improvements or specific types of adversarial scenarios. They often neglect the broader issue of managing information flow across the graph, particularly within dynamic and complex neighborhoods [60]. Additionally, they do not adequately consider the relationships between node features and the community structure within each neighborhood [36]. This is crucial for distinguishing between normal and anomalous nodes during message passing [60]. We propose a novel method that directly addresses the challenge of information flow management within GNNs. By focusing on observational local neighborhoods and using a knowledge graph to assess node normalcy or anomaly, our approach effectively mitigates the shortcomings of previous methods, offering enhanced robustness against adversarial influences. These unclean flows can have negative effects on messaging, such as over-smoothing and over-squashing of used information along possible paths that typically cannot avoid the attack effects and are unable to reduce the considerable degree of uncertainty of valuable information in the egonet for deterministic pseudo-label selection [10, 32].

In our definition, information flow correctly describes the intellectual movement of an active message substance from specific regions of higher to lower concentration. The active message in computing is effectively a messaging object capable of performing processing on its own messages available for positive receiving and correct processing. One line of our work aims to develop online knowledge-based K-hop message-passing GNNs, where node representation is updated by aggregating information from not only observational local neighborhoods but also all neighbors within K hops of the node. Additionally, our work leverages community information to enhance expressive power, which has been proven to be strictly more powerful than older works. Our essential contribution is a broad modern class of GNNs based on the discretized information flow stream, controlling the data flow on graphs and studying different information schemes for their solution. Second, we thoughtfully provide stability conditions for these flow schemes on different neighboring K-hop and additional parameters to correctly solve over-smoothing and over squashing of valuable information, to forcefully withdraw aggregation defects by finding all vulnerable points in an egonet, guessing the value of subsequent data destruction, and properly eliminating these adverse effects on message flow. Ultimately, based on our model, a linear data flow architecture develops that performs competitively on many popular benchmark datasets to provide better performance on different data paths in an egonet by knowledge-based finding an OLA and self-repairing the damaged subgraph, whether externally expected to arise from adversarial attacks or internally due to unexpected self-destruction.

Algorithm 1: The main algorithm of the KFGNN method for classifying attacked nodes [authors].

Input: $\hat{G}$, A, X, K;{$\hat{G}$:*Attacked Graph; A:Graph Node Matrix; X:Feature Matrix; K :Number of Network Layers*}
Output: $Z \in R^{n\times d}$; {*Z:Classified Nodes Matrix*}

for *Layer*$(l) = 1 \rightarrow K$ {*Message-passing Process:* $l \in \{1, 2, ..., K\}$ *K: Number of Layer* l} **do**
  for $(i = 1; i \leq V; i = i + 1)$ {*OLN assigning & repairing:* $T_i \rightarrow OLN$} **do**
    Similarization(*egonet*(i));{*Similarity finding* v_i *via egonet* T_i *& extract knowledge parameters*}
    Fuzzification(*egonet*(i));{*Fuzzification egonet* T_i *using the knowledge parameters*}
    Repairing(*egonet*(i));{*Repairing egonet* T_i *using OLN*}
  end
end
CleaningGraph$(\hat{G}) \rightarrow G$;{*Cleaned graph G*}
Classification(G);{*Node classification* v_i *of graph G*}
return *Z;* {Return Classified Z Matrix}

3 Proposed KFGNN approach

Up to now, we have properly presented the problem definition of the adversarial defending strategy in Section 1.2. In this specific section, the proposed KFGNN method will be accurately described. Initially, the problem modeling will be completely defined in Section 3.1. Next, the total problem definition and its key functions and algorithms will be described in Section 3.2. Also, the knowledge graph construction will be proposed in Section 3.3 and the similarization method will be illustrated in Section 3.4. In addition, the weighting key of the fuzzy method will be presented in Section 3.5. Finally, we will sufficiently illustrate the repair idea in Section 3.6.

3.1 Problem modeling

Our problem can be defined as follows: in GNNs, one of the critical processes is the information diffusion from neighboring nodes to a target node. This process is conducted through a message-passing function, which aggregates the information from neighbors and propagates it to the target node. Generally, the information propagation across different layers of the network is modeled as Eq.(1) where $h_i^{(l+1)}$ represents the updated features of node i at layer $l+1$, $N(x_i)$ is the set of neighbors of node i, D_{ii} and D_{jj} are degree matrices ensuring the normalization of the diffusion process, $W^{(l)}$ denotes the learned weights at layer l, and σ is the activation function applied to introduce non-linearity into the model. This diffusion model is commonly employed to transfer information from neighboring nodes to the target node,continuously updating the feature representations during the learning process.

$$h_i^{(l+1)} = \sigma\Big(\Sigma_{j\in N(x_i)} \frac{1}{\sqrt{D_{i,i}D_{j,j}}} U_{ij}^l h_j^l W^{(l)}\Big). \tag{1}$$

3.2 Proposed KFGNN method

This paper addresses the problem of semi-supervised node classification over graph data. We review our defense mechanism for analyzing the defense against adversarial attacks against GNNs, namely KFGNN. We first demonstrate the general defense mechanism and then explain more details in the following. Our defense mechanism was illustrated in Figure 2 and more details in Algorithm 1. A GNN model aggregates each node's attributes in each iteration to capture the topology and features of its rooted subtree computation graph (See Figure 2). We hypothesize that the unlabeled and test nodes may not be drawn from the same distribution as labeled nodes, contributing to the non-robustness of the graph neural network model defender. The GNN model may also be susceptible to failure in situations with a limited number of labels. To back this hypothesis, we analyze it as the case that such an assumption often happens in downstream tasks. During node classification, adversarial nodes easily occur in the graph, causing high-confidence error predictions.

To improve the accuracy and robustness of GNNs against adversarial attacks, fuzzy modeling has been integrated into the diffusion process. The purpose of this fuzzy approach is to assign each node within the receptive field of a target node a fuzzy membership degree, representing the strength of its association with the target node. This membership degree, U_{ij}, for each node u_i, is computed based on various distances such as structural, feature, and communal distances, and is defined to reflect the normality or abnormality of the node concerning the target node. In the past, the essential procedure of KFGNN was sufficiently illustrated in Figure 2. The main key algorithm of the modern method is sufficiently shown in Algorithm 1, which is demonstrated in pseudo-code language. KFGNN trains

Algorithm 2: The knowledge extraction algorithm as Similarization function [authors].

Input: $\hat{G}$, A, X;{$\hat{G}$:*Attacked Graph; A:Graph Node matrix; X:Feature matrix*}
Output: α, β, γ; {*Structural, Feature, and Community Knowledge Parameters*}
1 **Initial:** Structural-distance(u, v): **Calculate** $\alpha(u, v)$;{*Using Eq.(1)*}
2 **Initial:** Feature-distance(u, v): **Calculate** $\beta(R, S)$ &**Construct** $feature–feature\ graph$; {*Using Eq.(2) & Eq.(3)*}
3 **Initial:** Community-distance(u, v): **Calculate** $\gamma(R, S)$; {*Using Eq.(4)*}
4 **Compute** $\alpha_i(u, v) = Structural\_distance(u, v)$ for all $v \in N_u$;{$\alpha_{distance}$ }
5 **Compute** $\beta_i(u, v) = Feature\_distance(u, v)$ for all $v \in N_u$;{$\beta_{distance}$ }
6 **Compute** $\gamma_i(u, v) = Community\_distance(u, v)$ for all $v \in N_u$;{$\gamma_{distance}$ }
7 **return** α, β, γ {Return Knowledge Parameters }

the robust node representation Z on the perturbed graph $\hat{G}$. First, it examines all found egonets for attack effects, identifying and repairing them through recovery tasks (Lines 1-7). According to Algorithm 1, KFGNN applies the similarity function (Line 3), the fuzzy method (Line 4), and the repairing technique (Line 5) for each egonet inside the loop **for** iteration (Lines 2-6). After repairing all the egonets, a clean graph is constructed (Line 8) for node classification (Line 9).

3.3 Knowledge extraction method

The key knowledge extraction algorithm utilizes the fuzzy scoring method, taking as input an attacked graph G, node matrix A, feature matrix X, and the number of network layers K, as shown in Algorithm 2. The knowledge in this context refers to the essential information extracted from the graph, which includes structural, feature-based, and community-related relationships between nodes. Initially, structural, feature, and community distances were extracted for each node using their respective functions (Lines 1-3). The structural knowledge is computed via the structural-distance function using Eq.1 (Line 1), capturing how nodes are connected in the graph. The feature knowledge is extracted by calculating feature distance using Eq.2 (Line 2) and constructing a feature-feature graph by applying Eq.3 (Line 2), which reveals similarities or differences in node attributes. Similarly, community knowledge is determined through Eq.4 (Line 3), indicating how nodes are grouped within the network. Ultimately, the extracted knowledge concludes in three parameters: α (Line 4), β (Line 5), and γ (Line 6). The calculated parameters are returned (Line 7).

3.4 Similarization method

To improve the accuracy and robustness of GNNs against adversarial attacks, fuzzy modeling has been integrated into the diffusion process. The purpose of this fuzzy approach is to assign each node within the receptive field of a target node a fuzzy membership degree, representing the strength of its association with the target node. This membership degree, U_{ij}, for each node u_i, is computed based on various distances such as structural, feature, and communal distances, and is defined to reflect the normality or abnormality of the node concerning the target node. In this work, we propose a novel fuzzy setting for robust GNNs, designed to handle uncertainty and vague graph data. This fuzzy model takes three inputs that capture different aspects of the graph's structure and features, and assigns a membership value between 0 and 1 to each of the computation graphs. This value indicates whether a node is considered normal or adversarial in relation to a target node. The identification of adversarial nodes is primarily determined by the graph's topology and the features of its nodes. Three types of adversarial nodes can be distinguished based on the graph information: (a) *Global anomalies*, which are identified based solely on node features, as they have distinct features compared to other nodes in the graph; (b) *Graph structural anomalies*, which are detected by analyzing the graph's structural information, revealing abnormal nodes with different patterns of connections; and (c) *Community-based anomalies*, which consider both node features and graph structural information, identifying nodes whose feature values are distinct from those of others within the same community (See Figure 4).

3.4.1 Structural distance

The similarity between two nodes can be understood in terms of their structural roles within the graph or their shared connections. Two nodes are considered similar if they are connected to many of the same neighboring nodes, which suggests they might have similar properties or roles within the network. For a target node v, The similarity score $S(u, v)$ for a node u with respect to the target node v can be calculated as Eq.(2) (See Figure 4).

$$\alpha = \frac{|N(u) \bigcap N(v)|}{|N(u) \bigcup N(v)|}. \tag{2}$$

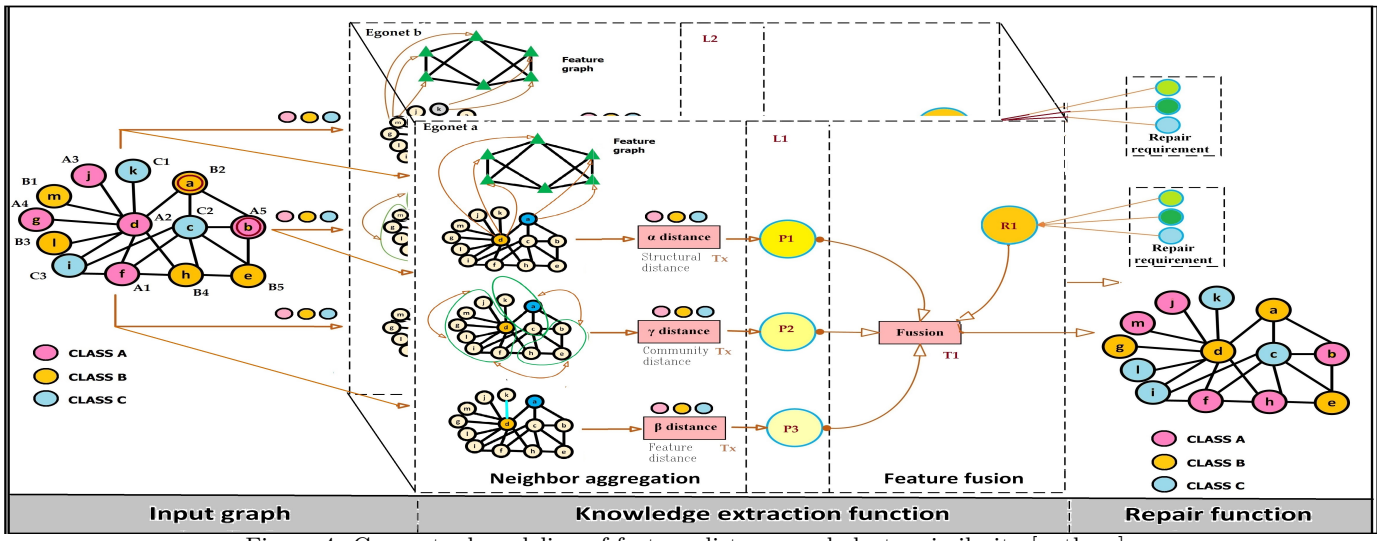


Figure 4: Conceptual modeling of feature distance and cluster similarity [authors].

3.4.2 Feature distance

In this section, we focus on the second input, which measures the feature similarity between the target node and its neighbors in the graph. This input is used by the fuzzy system to assess the degree of consistency or anomaly of the node features. To compute this input, we construct a feature–feature graph that represents the co-occurrence of features among the nodes, and then we calculate the feature similarity based on the feature–feature graph. The motivation for using the feature–feature graph is based on the observation that node features can significantly improve the performance of GNNs, as nodes with similar features tend to have similar roles or functions in the graph. Therefore, by capturing the feature similarity between the target node and its neighbors, we can exploit the rich information embedded in the node features and enhance the robustness of the fuzzy system. The feature–feature graph is a high-level graph that is built on top of the original node–node graph, and it reflects the feature correlations among the nodes.

The feature–feature graph has a two-layer network structure, where the first layer is the node–node graph that models the node interactions, and the second layer is the feature–feature graph that models the feature correlations. The two layers are not independent, but interact with each other through the co-occurring features. The feature–feature graph can capture higher order feature correlations that are not directly observable in the node–node graph, and thus provide a more comprehensive and expressive representation of the graph data. The construction of the feature–feature graph is based on the feature lists of all nodes, where each node is annotated with a set of features. Two features are connected in the feature–feature graph if they co-occur in the same node, and the weight of the connection reflects the strength of the co-occurrence. Specifically, for each feature f_i, we compute a normalized correlation weight with each neighboring feature f_j to measure the directed affinity degree from f_i to f_j, based on their co-occurrence frequency.

The correlation weight is calculated Eq.(3) where $occu(f_i, f_j)$ is the frequency of feature f_i and f_j co-occurring in the same node, and $|F|$ is the number of features. The correlation weight w_{ij} indicates how likely feature f_j is to appear when feature f_i is present, and it ranges from 0 to 1. The feature–feature graph is a undirected graph, as the correlation weight may be symmetric, i.e., $w_{ij} = w_{ji}$. After constructing the feature–feature graph, we use it to calculate the feature similarity between the target node and its neighbors in the node–node graph. The feature similarity is defined as the average correlation weight of the features that are shared by the two nodes, and it ranges from 0 to 1. A higher feature similarity means that the two nodes have more common and strongly correlated features, and thus are more likely to be normal with respect to each other. A lower feature similarity means that the two nodes have fewer or weakly correlated features, and thus are more likely to be abnormal with respect to each other.

The feature similarity is used as the first input to the fuzzy system, which will combine it with the other two inputs to produce a final fuzzy score for each node. In this study, we put the features belonging to a node that have a value of 1 in an cluster. To calculate the similarity between two nodes, we need to calculate the distance between two clusters U and V. For two clusters U and V, first for the distance between any node u in U and any node v in V and then the arithmetic mean of these distances are calculated. Eq.(4) defines how the similarity between two clusters U and V is calculated. It is described as a function of the distances $d(p, q)$ between pairs of elements $(u, v), u \in U, v \in V$ of the clusters. This distance is shown in Eq.(4) where $d(i, j)$ is the length of a weighted shortest path connecting feature u

Algorithm 3: The rull-based Fuzzification algorithm [authors].

Input: $d_{feature}$, $d_{structural}$, $d_{community}$;{*Feature Distance, Structural Distance, and Community Distance for each neighboring node x_j of the target node x_i* }
Output: U_{ij}; {*Influence weight for each neighboring node x_j*}

1 **Fuzzification:** for each distance parameter d calculate membership degrees;{}
Calculate $\mu_{Low}(d)$;{*Membership degree using Eq.(6) where* $d \in \{d_{feature}, d_{structural}, d_{community}\}$}
Calculate $\mu_{Medium}(d)$;{*Membership degree using Eq.(6) where* $d \in \{d_{feature}, d_{structural}, d_{community}\}$}
Calculate $\mu_{High}(d)$;{*Membership degree using Eq.(6) where* $d \in \{d_{feature}, d_{structural}, d_{community}\}$}

2 **Rule Evaluation:** for each of 27 fuzzy rules Rk;{}

Determine Firing Strength W_{Rk}
$W_{Rk} = \min(\mu S_1(d_{feature}), \mu S_1(d_{structural}), \mu S_1(d_{community}))$;{$S_i \in \{Low, Medium, High\}$ *using Eq.(7) & Eq.(8)*}

3 **Aggregation:** calculate total firing strengths;{}

In Normal Category: $W_{normal} = \sum_{Normalrules} W_{Rk}$;{*Using Eq.(9)*}
In Anomalous Category: $W_{anomalous} = \sum_{Anomalousrules} W_{Rk}$;{*Using Eq.(10)*}

4 **Defuzzification:** calculate the crisp influence weight U_{ij} by applying the centroid method;{}

Aggregate $A(z)$
$A(z) = op\_activation\_normal \cup op\_activation\_anomalous$;{*Using Eq.(13)*}
Calculate Center of Area U_{ij}
$U_{ij} = \frac{\int z.\mu_A(z)dz}{\int \mu_A(z)dz}$;{*Using Eq.(14)*}

5 **return** U_{ij} {Return influence weight for each neighboring node x_j}

and j, and the sum is taken over all pairs of nodes in two clusters (See Figure 4).

$$w_{ij} = \frac{occu(f_i, f_j)}{\Sigma_{t=1}^{|F|} occu(f_i, f_t)}, \qquad f(v_i, T) = \begin{cases} w_{ij} : \text{if } w_{ij} > \sigma \\ 0 : \text{Otherwise} \end{cases} \tag{3}$$

$$\gamma = \frac{1}{|N_U|.|N_V|} \Sigma_{i=1}^{N_U} \Sigma_{j=1}^{N_V} d(u_i, v_j) i \in R, j \in S. \tag{4}$$

3.4.3 Community distance

We address the task of computing scores for individual nodes within the context of GNN's robustness. Specifically, we consider scenarios in which nodes within the graph may exhibit anomalous behavior in relation to a target node. In order to address this issue, we propose leveraging community information. We aim to detect anomalies and assign appropriate scores by considering the target node's community and its interactions with other nodes. Our approach rests on the assumption that nodes belonging to the same community as the target are more likely to be normal. In contrast, those from different communities are more likely to be anomalous. We introduce an outlier score based on a node-centric metric, which can be formally defined as Eq.(5) where v represents a node and u is tarfet node, each e_i, $1 \leq i \leq k$, represents the neighbors count of v belonging to the i^{th} external community and we assume $e = 1$, if there is no external neighbor of node v, d_v represents the degree of node v. In general, $0 \leq d_S \leq 1$, and a lower d_S value means the node v is loosely connected within its community, i.e., belongs to more than one community. In contrast, $d_S = 1$ indicates that a node v has a strong association with its community (See Figure 4).

$$\gamma = \frac{\frac{1}{e_1} + \frac{1}{e_2} + \ldots + \frac{1}{e_k}}{d_v}. \tag{5}$$

3.5 Knowledge-based fuzzification

In Algorithm 3, we calculate the influence weight W_{ij} for each neighboring node x_j of the target node x_i using a fuzzy inference system (See Figure 4). In Line 1, we begin by fuzzifying the input parameters —Feature Distance

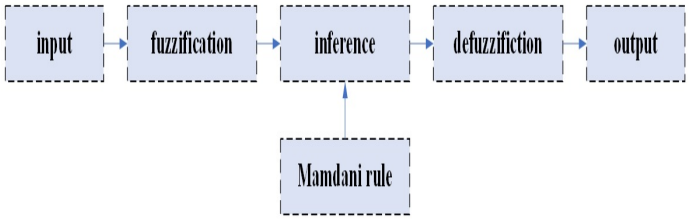


Figure 5: The framework of using a fuzzy logic-based model [authors].

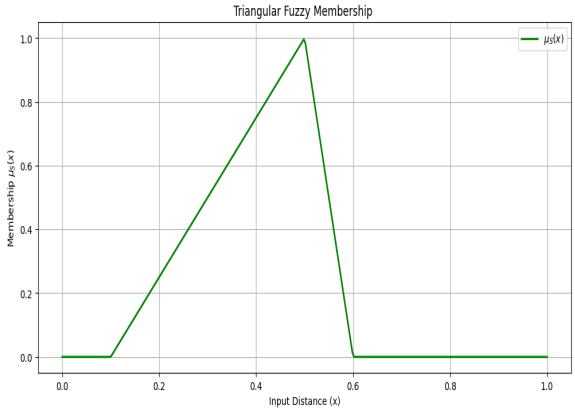


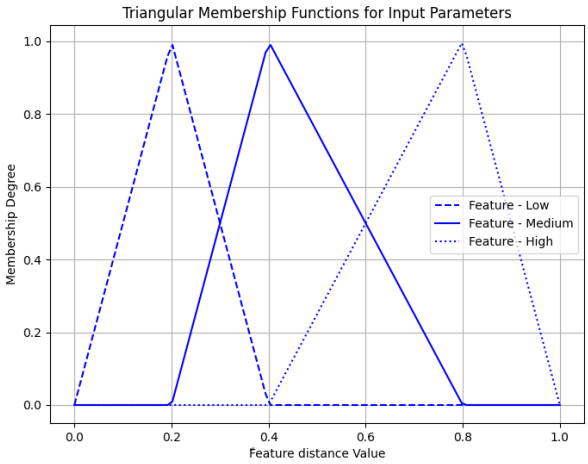
Figure 6: Triangular fuzzy membership [authors].

$d_{feature}$, Structural Distance $d_{structural}$, and Community Distance $d_{community}$ — by computing their membership degrees using triangular membership functions defined in Eq.(6). Next, in Line 2, we evaluate the fuzzy rules by applying all 27 rules from Table 2, calculating the firing strength W_{Rk} for each rule Rk using the fuzzy AND operator (minimum of membership degrees), as shown in Eq.(7) and Eq.(8). In Line 3, we aggregate the firing strengths for the Normal and Anomalous categories using Eq.(9) and Eq.(10). Then, in Line 4, we compute the activated output membership functions by multiplying the aggregated firing strengths with the output membership functions for Normal and Anomalous classifications, as indicated in Eq.(11) and Eq.(12) where C_{normal} and $C_{anomalous}$ are the centroid of output membership for the Normal and Anomalous categories, respectively. In Line 5, we aggregate the outputs to form the overall membership function $A(z)$ using Eq.(13). Finally, in Line 6, we perform defuzzification to calculate the crisp influence weight W_{ij} using the centroid method as described in Eq.(14). This crisp value W_{ij} quantifies the adjusted influence of neighboring node x_j on the target node x_i and is used in the GNN's message-passing process to enhance robustness against adversarial attacks and mitigate over-squashing.

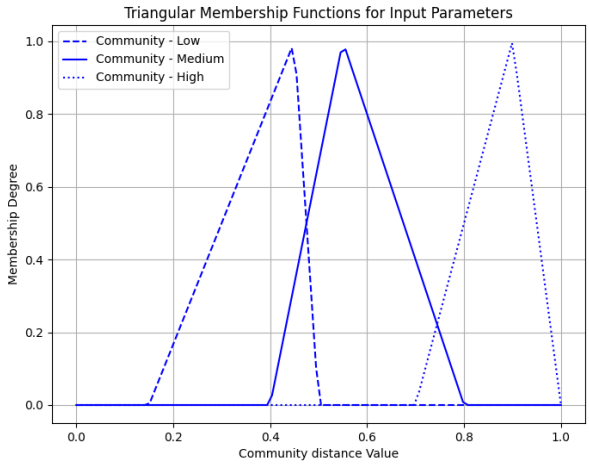
3.5.1 Membership functions for fuzzification

The proposed fuzzy rule-based model employs triangular membership functions to transform three key parameters—Feature Distance, Structural Distance, and Community Distance—into fuzzy sets categorized as Low, Medium, or High. This fuzzification process, as depicted in Figure 5, ensures a smooth transition between crisp input values and their linguistic representations, enhancing the model's ability to manage uncertainty, particularly in adversarial scenarios. The triangular membership function, illustrated in Figure 6, defines each fuzzy set using three key parameters: lower bound (d), midpoint (e), and upper bound (f), allowing the system to capture fine-grained distinctions in node similarity and dissimilarity. To ensure adaptability across datasets, the membership function parameters (d, e, f) are dynamically derived from data distributions rather than fixed thresholds. Specifically, they are computed based on the observed range of Feature, Structural, and Community Distances, ensuring robustness without requiring manual tuning. This domain-agnostic approach enables the model to generalize effectively across different graph structures and adversarial conditions. During rule evaluation, each node's fuzzy scores across the three parameters are aggregated to classify it as Normal or Anomalous, capturing uncertainty in node behavior. Figure 7 illustrates how multiple triangular sets are utilized to classify output variables, ensuring a flexible yet precise assessment of anomalies. This adaptive mechanism enhances the stability of GNNs, mitigating adversarial perturbations while maintaining structural integrity in various real-world applications.

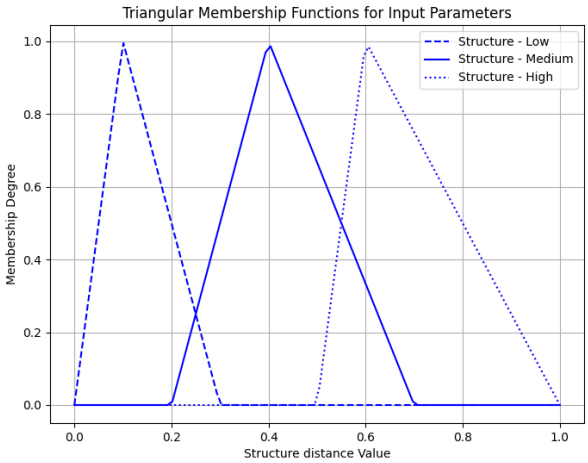
To fuzzify each parameter, we define triangular membership functions for the levels Low, Medium, and High. These membership functions assign a fuzzy score between 0 and 1 to each input value based on its proximity to



(a) Triangular fuzzy membership for Feature Distance Value.



(b) Triangular fuzzy membership for Community Distance Value.



(c) Triangular fuzzy membership for Structure Distance Value.

Figure 7: Triangular fuzzy membership sets for output variables [authors].

the characteristic range of each linguistic term (Low, Medium, High). The membership function for each fuzzy set $S \in \{Low, Medium, High\}$ is represented as Eq.(6) where d is the lower bound of the fuzzy set, be Low which the membership is zero. E is the midpoint or peak of the fuzzy set, where the membership is maximal (equal to 1). F is the upper bound of the fuzzy set, above which the membership returns to zero.

$$\mu S(x) = \begin{cases} 0 & : x \leq d \\ \frac{x-d}{e-d} & : d < x \leq e \\ \frac{f-x}{f-e} & : e < x \leq f \\ 0 & : x \geq f \end{cases} \tag{6}$$

This triangular function ensures a smooth transition between membership values, capturing the gradual change in similarity or dissimilarity between nodes. Below is an example of how each parameter can be mapped to fuzzy levels based on defined ranges. Through fuzzification, each parameter is mapped onto a linguistic scale (Low, Medium, High), allowing the system to assess node similarity across various dimensions of distance. The fuzzified values will then be used in the rule evaluation step, where the fuzzy rules determine the node's classification (normal or anomalous) based on these fuzzy degrees of membership. This approach provides a robust method for handling uncertainty and ensuring resilience in the GNN against adversarial nodes, as each node's influence is calibrated according to its computed similarity or anomaly.

Table 2: The complete set of 27 fuzzy rules based on the five parameters [authors].

Rule	Feature Distance	Structural Distance	Community Distance	Classification
1	Low	Low	Low	Normal
2	Low	Low	Medium	Normal
3	Low	Low	High	Normal
4	Low	Medium	Low	Normal
5	Low	Medium	Medium	Normal
6	Low	Medium	High	Normal
7	Low	High	Low	Normal
8	Low	High	Medium	Normal
9	Medium	Low	Low	Normal
10	Medium	Low	Medium	Normal
11	Medium	Medium	Low	Normal
12	Medium	Medium	Medium	Normal
13	Medium	Low	High	Normal
14	Low	High	High	Normal
15	Medium	Medium	High	Normal
16	High	High	Low	Anomalous
17	High	High	Medium	Anomalous
18	High	High	High	Anomalous
19	Medium	High	High	Anomalous
20	Low	High	High	Anomalous
21	High	Low	High	Anomalous
22	High	Medium	High	Anomalous
23	Medium	High	Medium	Anomalous
24	High	Low	Medium	Anomalous
25	High	Medium	Medium	Anomalous
26	Medium	High	Low	Anomalous
27	Medium	Low	High	Anomalous

3.5.2 Formulation of fuzzy rules

The proposed fuzzy rule-based system is designed for node classification using 27 distinct rules, as presented in Table 2. It evaluates three key parameters—Feature Distance, Structural Distance, and Community Distance—each categorized into Low, Medium, or High linguistic subsets using triangular membership functions. These parameters serve as fundamental indicators for detecting anomalies within a graph: Feature Distance captures semantic similarity, Structural Distance assesses topological proximity, and Community Distance measures a node's alignment within its local neighborhood. This structured rule-based approach systematically covers all possible parameter relationships, ensuring a refined assessment of node similarity and anomaly detection while preventing the model from overlooking critical deviations. The classification process dynamically determines membership boundaries (d, e, f) from data distributions rather than fixed thresholds, ensuring dataset independence and adaptability to varying graph sizes and structures. The fuzzy inference mechanism evaluates node states based on the combination of fuzzy subsets, where nodes exhibiting predominantly Low or Medium values across parameters are classified as Normal (e.g., w_{R1} in Eq.(7)), signifying high similarity to the target node. Conversely, nodes displaying High values in two or more parameters are flagged as Anomalous (e.g., w_{R16} in Eq.(8)), indicating substantial dissimilarity and potential adversarial characteristics. Rules R1 through R15 classify nodes as Normal, reflecting their close association with the target's local neighborhood.

$$w_{R1} = Feature\_Low \wedge Structural\_Low \wedge Community\_Low. \tag{7}$$

$$w_{R16} = Feature\_High \wedge Structural\_High \wedge Community\_Low. \tag{8}$$

The robustness of KFGNN is validated through experiments on benchmark datasets (Cora, Citeseer, and PubMed) (See Table 4), demonstrating stable classification performance under adversarial conditions. The system's resilience is attributed to triangular membership overlaps, which capture gradual variations in node similarity, and comprehensive rule combinations, preventing adversarial perturbations from evading detection. This adaptability is particularly critical for scenarios where adversaries manipulate specific node attributes while maintaining structural integrity. By integrating fuzzy logic-based inference with dynamic membership assignment, the proposed approach provides an interpretable and scalable anomaly detection mechanism, ensuring improved performance in complex graph-based learning environments. Consequently, these nodes exert a greater influence on the target node during the GNN embedding update. The rule evaluation is done on the basis of the Table 2. Rules R16 through R27 are used to classify nodes as "Anomalous". These rules activate when two or more parameters exhibit High distances. Such a configuration implies dissimilarity across multiple dimensions, which may indicate an adversarial or abnormal node. In the embedding update process, these nodes are assigned a lower influence to minimize their impact on the target node. Each rule's firing strength is

determined based on the combination of fuzzy values for the three parameters, and the fuzzy AND operator $\wedge$ is used to connect these conditions. This structured rule-base, as presented in Table 2, provides a systematic approach for assessing node influence, allowing for robust classification and mitigating adversarial effects in the GNN.

3.5.3 Aggregation and defuzzification of rule outputs

After evaluating each individual rule, the outputs are aggregated to form an overall fuzzy influence score for each neighboring node. This score, which represents the combined effect of the rules, is then defuzzified to yield a crisp influence weight W_{ij}, quantifying the impact of each node on the target node. The aggregation process involves calculating the firing strengths of all relevant rules for each output category (e.g., Normal or Anomalous). In Eq.(9)-Eq.(10) rules depicts the overall firing strength or degree of fulfillment of the fuzzy rules pertaining to anomaly or normally nodes where w_{normal} and $w_{anomalous}$ are the overall firing strengths representing the degree to which the conditions for each classification are met. To compute the final influence score, the fuzzy sets associated with each rule are clipped at their firing strengths. For instance, the induced or resultant membership functions for each category (e.g., op_activation_normal and op_activation_anomalous) are computed as Eq.(11) and Eq.(12). The final aggregated output membership function A(z) is obtained by combining the induced membership functions using the union operator as Eq.(13).

$$w_{normal} = w_{R1} \vee w_{R2} \vee \ldots \vee w_{R15}, \tag{9}$$

$$w_{anomalous} = w_{R16} \vee w_{R2} \vee \ldots \vee w_{R27}, \tag{10}$$

$$op\_activation\_normal = w_{normal} \wedge op\_normal, \tag{11}$$

$$op\_activation\_anomalous = w_{anomalous} \wedge op\_anomalous, \tag{12}$$

$$A(z) = op\_activation\_normal \cup op\_activation\_anomalous. \tag{13}$$

3.5.4 Defuzzification

The final step in our fuzzy rule-based system is defuzzification. We employ the centroid defuzzification method, as it provides a precise and reliable output. This method calculates the center of area (COA) under the membership function curve, which represents the fuzzy influence score. The COA method is based on finding the center of gravity of the combined fuzzy set, allowing for an accurate crisp output value. To achieve this, the aggregated membership function, $A(z)$, which represents the influence score for each neighboring node, is divided into multiple sub-areas. For each sub-area, the area and center of gravity are calculated. The defuzzified value, U_{ij}, is obtained by taking the weighted average of these sub-areas, as shown in Eq.(14) where z denotes the influence weight variable, and $A(z)$ is the aggregated membership function. This approach provides a crisp influence weight for each neighboring node, reflecting its overall impact on the target node. To further refine the classification, we evaluate the defuzzified output against predefined ranges. For instance, nodes are classified based on their influence weight W_{ij} as $Normal : 0 < U_{ij} < 0.33$, $ModerateInfluence : 0.33 < U_{ij} < 0.67$, and $Anomalous : 0.67 < U_{i,j} < 1$. This method ensures that the final influence score accurately reflects each node's role, whether it enhances or diminishes the GNN's robustness. Through the defuzzification process, the model effectively handles uncertainties, yielding a clear and actionable influence weight for each node.

$$U_{ij} = COA = \frac{\int z.\mu_A(z)dz}{\int \mu_A(z)dz}. \tag{14}$$

3.6 Repair function

Our repair function consists of incorporating the membership degree into the calculations of the neural network, as shown in Eq.(15) and Eq.(16). By integrating the membership degrees, we can adjust the influence of each node in the computation graph, thereby repairing the GNN and enhancing its robustness against adversarial attacks. In fuzzy GNNs, we describe the k^{th} layer of a fuzzy computation graph as a vector of fractions of nodes assigned to it by the membership function. The membership degree is expressed as real-valued numbers rather than binary values in the range $[0, 1]$. In hard GNNs, the $(l+1)^{th}$ convolutional layer is defined as Eq.(16). We expect that the mapped samples can preserve the same local topological structure as the original space while increasing the distance between adversarial instances and normal nodes. This approach ensures that the neural network can effectively distinguish between normal

and anomalous patterns, reducing risk factors in the decision-making processes.

$$\begin{aligned} h_i^{(l+1)} &= \sigma\Big(\Sigma_{j\in N_f(x_i)}\frac{1}{\sqrt{D_{i,i}D_{j,j}}}U_{ij}^{l}h_j^{(l)}W^{(l)}\Big), \\ \text{Sbject to } \Sigma_{j\in N(xi,2)}U_{ij} &= 1 \;\&\; 0 < \Sigma_{j\in N(xi,2)}U_{ij} < n \\ &\textit{for all } i \in [n] \textit{ and } j \in N_f(xi,2), \end{aligned} \tag{15}$$

$$h_i^{(l+1)} = \sigma\Big(\Sigma_{j\in N(x_i)}\frac{1}{\sqrt{D_{i,i}D_{j,j}}}U_{ij}^{(l)}h_j^{(l)}W^{(l)}\Big). \tag{16}$$

Table 3: Statistical information of our experimental Cora, Citeseer, and Pubmed datasets.

Dataset	# Nodes	# Edges	# Features	# Classes	Avg. Degree	Intra-Class Edges	Inter-Class Edges
Cora	2,485	5,069	1,433	7	4.07	8,550	2,006
Citeseer	2,110	3,668	3,703	6	3.48	6,696	2,408
Pubmed	19,717	44,338	500	3	4.50	71,130	17,518

4 Experimental results

Our successful experiments aim to investigate the proposed KFGNN method. Initially, the typical setups in the experimental results, settings of the adversarial attacks, the necessary hardware with their running environments, and comparison methods are described, along with the specifications of the benchmark dataset used in Section 4.1. Second, the empirical experimental evaluation results on accuracy, path length, and complexity are explained in Section 4.2. Ultimately, the parameter analysis is presented in Section 4.3.

4.1 Experimental setup and comparison methods

Our experimental hardware machine and environment consist of Intel Core(TM) i7-10750H CPU@ 2.60GHz 2.59 GHz, ,6 Core(s), 12 Logical Processor(s), 16.0 GB DDR4 memory, and Windows 11 Home 64-bit operating system, and x64-based data-chunk. The relevant algorithms used in the experiment and the comparative experiments are all implemented in the Python 3.11.5 deep learning framework under Windows 11. Python is used to implement all algorithms. Also, our experiments on citation network datasets are designed to evaluate the model's performance and robustness on the semisupervised node classification task. Documents are nodes, while citation links are directed edges in a citation network dataset. A human-annotated topic and a feature vector are assigned to each node. In this paper, we conduct comprehensive experiments on four widely used datasets: Cora, and Pubmed, Citeseer [37, 44]. The PubMed dataset contains entries containing real-values indicating the Term Frequency-Inverse Document Frequency (TF-IDF) of corresponding words. There are binary entries in the feature vector for CiteSeer and Cora datasets that indicate whether a given word is present or absent in a dictionary. We use undirected versions of the graphs for all experiments, even though the networks are directed.

$$Accuracy = \frac{NumberofCorrectlyClassifiedNodes}{TotalNumberofNodes} = \frac{TP+TN}{TP+TN+FP+FN}. \tag{17}$$

In our study, the accuracy metric was employed as a standard measure for classification performance, defined as Eq.(17) where True Positives (TP) and True Negatives (TN) denote correctly classified nodes, while False Positives (FP) and False Negatives (FN) represent misclassified nodes. Given the adversarial nature of the problem, we report two accuracy measures: *Clean Accuracy* as the classification accuracy on the original, unperturbed graph and *Robust Accuracy* as the classification accuracy on the adversarially attacked graph after applying the proposed defense mechanism. To ensure a rigorous and unbiased evaluation, we adopted a 10-fold cross-validation approach. Table 3 summarizes the statistics of the datasets. The dataset was randomly partitioned into 10 equal subsets, where in each iteration, one subset was designated as the test set, while the remaining nine subsets were utilized for training and validation. This process was iterated 10 times, and the final accuracy was reported as the mean accuracy across all folds, providing a robust assessment independent of specific data partition.

The following seven categories of methods are considered to provide a comprehensive comparison of our proposed KFGNN with several state-of-the-art approaches: GCN, GAT, RGCN, GCN-SVD, GCN-Jaccard, and Pro-GNN. GCN is a classic low-pass filtering method that aggregates neighboring information to update node representations. The

Table 4: Accuracy comparison of our KFGNN and multiple defense methods against various attacks [authors].

Model	Attack Type	Ptb Rate	Dataset		
			Cora	Citeseer	Pubmed
GCN	Mettack	0%	83.63	71.83	86.45
		10%	70.71	67.41	80.38
		20%	59.84	61.98	76.24
	Nettack	0%	82.44	81.07	79.55
		10%	71.18	79.71	71.68
		20%	60.07	62.38	66.63
GAT	Mettack	0%	0.827	73.09	82.97
		10%	0.713	70.53	73.57
		20%	60.08	62.93	67.54
	Nettack	0%	76.04	81.27	79.23
		10%	70.24	60.85	71.54
		20%	65.24	61.59	66.59
GCN-Jaccard	Mettack	0%	82.14	72.01	84.28
		10%	75.22	69.41	66.83
		20%	65.85	59.21	62.87
	Nettack	0%	81.72	80.17	82.06
		10%	70.03	76.48	78.76
		20%	61.42	76.19	69.88
GCN-SVD	Mettack	0%	80.85	70.49	82.84
		10%	71.52	68.68	82.82
		20%	58.99	58.37	81.96
	Nettack	0%	77.02	80.24	83.44
		10%	71.53	72.87	78.27
		20%	58.45	62.01	67.01
RGCN	Mettack	0%	83.39	71.03	82.44
		10%	72.38	67.56	77.73
		20%	59.55	62.33	70.94
	Nettack	0%	76.75	78.25	80.94
		10%	70.69	75.40	72.35
		20%	59.28	60.31	61.52
Pro-GNN	Mettack	0%	83.95	73.09	86.88
		10%	79.45	72.35	85.39
		20%	73.44	70.08	81.65
	Nettack	0%	82.75	80.97	83.26
		10%	78.87	79.85	77.20
		20%	69.95	74.87	76.15
DualRGNN	Mettack	0%	82.38	74.32	86.84
		10%	78.35	74.71	85.14
		20%	77.41	70.55	82.10
	Nettack	0%	84.29	82.65	85.19
		10%	73.25	81.47	76.33
		20%	74.11	77.37	77.44
KFGNN	Mettack	0%	81.23	72.25	84.39
		10%	77.88	73.80	83.84
		20%	75.20	71.11	82.16
	Nettack	0%	80.23	81.22	82.55
		10%	78.21	80.23	77.88
		20%	71.05	77.84	77.55

GAT approach is a graph neural network model that generates node representations by using a local self-attention mechanism. The RGCN method represents the features of nodes as Gaussian distributions, which can enhance model robustness by penalizing nodes with high variances. The GCN-SVD algorithm determines that the nettack will affect lower singular values of the adjacency matrix and uses low-rank approximation to preserve large singular values above a certain threshold to restore graphs. The GCN-Jaccard technique removes adversarial edges where the Jaccard similarity of the node is below a certain threshold, resulting in a clean graph. This clean graph is taken as the input of GCNs. In order to denoise graphs, Pro-GNN constrains the graph data to be low-rank, sparse, and feature-smoothed. This is a state-of-the-art GCN model for defense applications. In our experiments, we follow the setup described in [25]. In the original paper, the default parameters for GCN and GAT were used. In GCN-SVD, the reduced rank of the attacked graph is chosen from 5,15,100,200. The number of hidden units for RGCN is set to 16,32,64,128. We set the similarity threshold between 0.01 and 0.2 in GCN-Jaccard. To adjust the hyperparameters for Pro-GNN, we follow the original settings.

Table 5: Statistical performance information for node classification of used GCN, GAT, and GIN models [authors].

Model	Cora	Citeseer	Pubmed
GCN	0.821	0.718	0.781
GAT	0.822	0.709	0.776
GIN	0.828	0.721	0.789

4.2 Empirical result evaluation

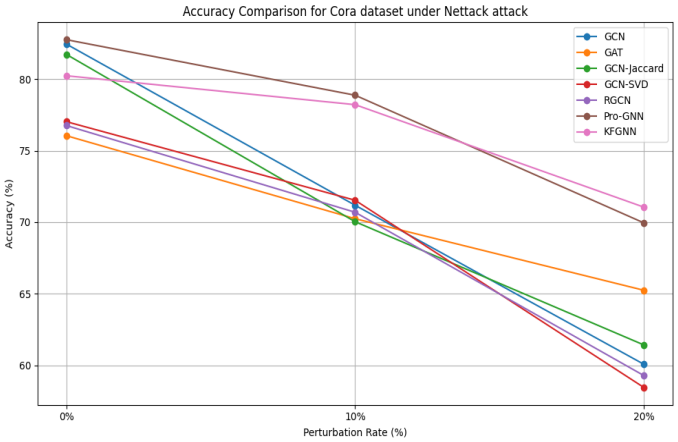
Nettack and Mettack are two strong adversarial attacks for evaluating and designing defenses, and we have utilized these two attacks to evaluate our work. Table 4 presents the average node classification accuracies for the GCN, GCN-SVD, GCN-Jaccard, RGCN, GAT, Pro-GNN, and the proposed approach, KFGNN. The empirical findings reveal that, in the absence of attacks, the proposed approach demonstrates comparable accuracy to the classical GCN, and in some cases, it even improves the model's generalization and performance. Importantly, these results affirm that our approach does not compromise the performance of the underlying network. This is particularly significant as real-world scenarios often involve uncertain knowledge regarding potential malicious perturbations on the input graph. Hence, an effective defensive strategy must not diminish the model's predictive capabilities while enhancing its robustness. From another perspective, the results indicate that our proposed approach, KFGNN, performs on par with or even surpasses state-of-the-art defensive baselines in several instances.

Our results indicate that the proposed detection technique is effective. Statistically, it outperforms comparison methods for local and global attacks on all datasets except Polblogs, with a relatively high accuracy. Due to the nature of our proposed method, the number of intraclass edges and interclass edges is significant. Based on Table 5 for all three datasets, intraclass edges are greater than interclass edges, implying homophily. In the context of GNN, homophily refers to the tendency of nodes with similar attributes or features to be more connected than nodes with dissimilar attributes. The proposed method demonstrates superior performance in cora and pubmed datasets where intraclass edges outnumber interclass edges due to its ability to exploit the inherent clustering properties of the nodes. In such datasets, nodes typically form groups or clusters, with dense connections within the same class and fewer connections between different classes. The proposed method leverages this clustering effect to identify anomalies effectively; it assumes that anomalous nodes are likely to exhibit connection patterns that are inconsistent with the prevalent intraclass connectivity. Therefore, by focusing on deviations from the dense intraclass connections, the method can more accurately pinpoint nodes that do not conform to the expected patterns, thereby enhancing its anomaly detection capabilities. This approach is particularly adept at recognizing subtle variations in connectivity that may indicate anomalous behavior, which may only be as apparent in datasets with a pronounced clustering tendency.

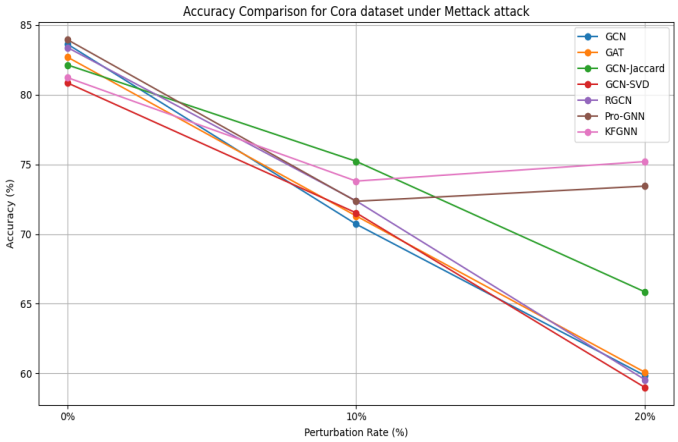
4.3 Parameters analysis

In this study, we propose a novel approach that not only improves the robustness of GNNs against adversarial attacks but also effectively addresses the issue of over-squashing—a problem that occurs when messages from distant nodes become overly compressed as they propagate through multiple layers. Over-squashing limits the expressiveness of GNNs, especially as the number of layers increases, leading to information loss and reduced prediction accuracy. To mitigate this, our method leverages a knowledge-based fuzzy logic system, assigning fuzzy membership degrees to neighboring nodes based on their structural, feature-based, and community relationships to the target node. We employ a set of 27 fuzzy rules that consider all possible combinations of these three input parameters, each categorized into three fuzzy subsets: Low, Medium, and High. This comprehensive rule base allows for a nuanced evaluation of node influence, ensuring that only the most relevant information is propagated while reducing the influence of noisy or irrelevant nodes.

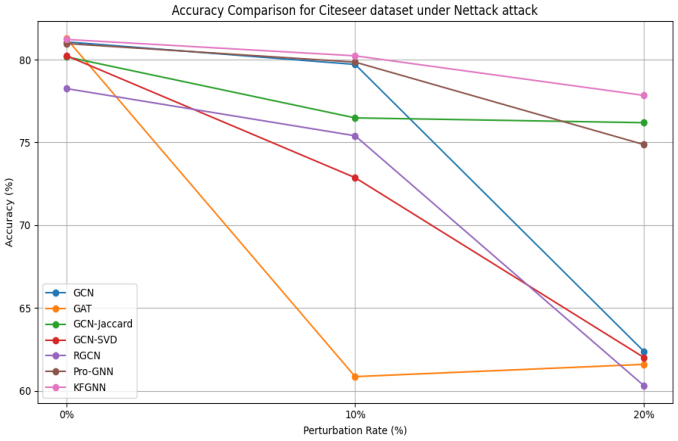
Nodes with greater combined distances from the target node receive lower fuzzy membership degrees as determined by the fuzzy inference system and defuzzification process. This prevents distant, irrelevant nodes from having a disproportionate influence, particularly in the presence of adversarial modifications. By assigning appropriate fuzzy membership degrees through the rule-based system, our method effectively controls the message-passing process and prevents over-squashing of node embeddings. By regulating the contribution of distant nodes using the fuzzy rule-based system, our method allows the GNN to maintain its expressiveness even as the number of layers increases. This is particularly evident in the experimental results on benchmark datasets such as Cora, PubMed, and Citeseer, where increasing the number of layers from 2 to 3 or more did not result in the typical performance drop observed in standard GNNs due to over-squashing. For instance, on the Cora dataset, our method achieved a stable accuracy of 82.76%, while on PubMed and Citeseer, accuracies of 85.23% and 73.74% were maintained, respectively (See Figure 8 and Table 4). These results highlight the effectiveness of the fuzzy logic-based approach in preserving information flow, preventing the over-squashing of messages, and enhancing the overall robustness of GNNs against adversarial attacks, even as the network scales to deeper layers. This dual advantage ensures that our method maintains high performance in both



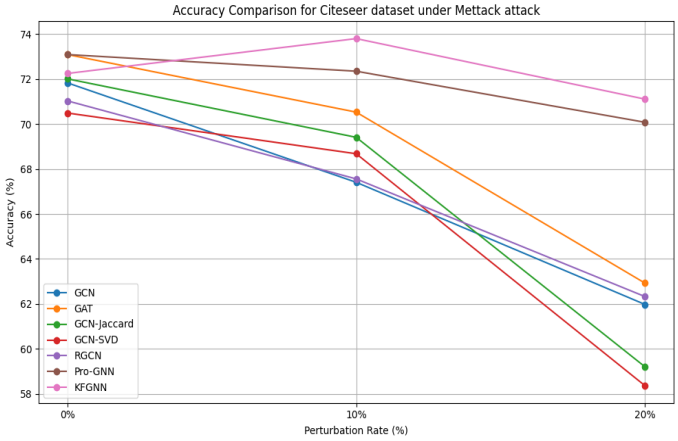
(a) Effect on Accuracy for Cora under Nettack attack.



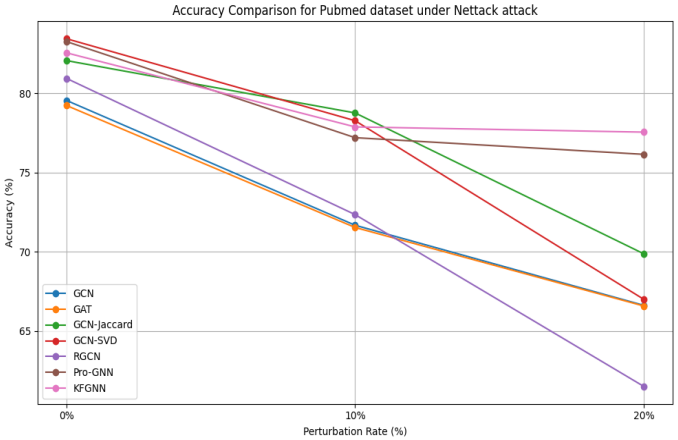
(b) Effect on Accuracy for Cora under Mettack attack.



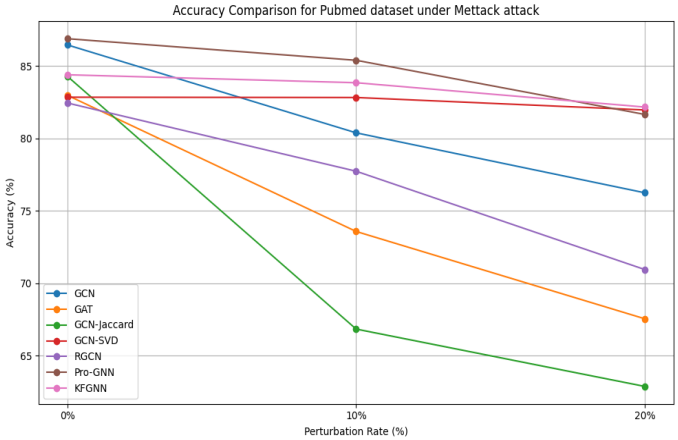
(c) Effect on Accuracy for Citeseer under Nettack attack.



(d) Effect on Accuracy for Citeseer under Mettack attack.



(e) Effect on Accuracy for Pubmed under Nettack attack.



(f) Effect on Accuracy for Pubmed under Mettack attack.

Figure 8: Effect of perturbation rate on Accuracy for Cora, Citeseer, and Pubmed datasets [authors].

adversarial and non-adversarial settings, providing a scalable solution for complex, large-scale graphs.

The time complexity of the proposed KFGNN is approximately $O(K \times V^2)$, which can be computed as follows. The time complexity of Algorithm 1, which is the main algorithm of the KFGNN method, can be analyzed by examining its loops and the complexity of the functions it calls. The outer loop runs K times, where K represents the number of network layers. Inside this loop, there is another loop that iterates over all nodes V in the graph. For each node, the Similarization, Fuzzification, and Repairing functions are invoked. The Similarization function calculates the structural,

Table 6: The Values of P-value Parameters for KFGNN and the Comparison Models [authors].

P-value	Accuracy Difference	Comparison Accuracy	KFGNN Accuracy	Perturbation	Attack	Dataset	Model
0.02	1.19	82.44	83.63	0%	Mettack	Cora	GCN
0.03	1.45	72.35	73.80	10%	Mettack	Citeser	GCN
0.01	-2.12	84.28	82.16	20%	Mettack	Pubmed	GAT
0.04	-0.99	76.19	75.20	20%	Mettack	Cora	GCN-Jaccard
0.04	0.33	77.88	78.21	10%	Nettack	Citeser	GCN-SVD

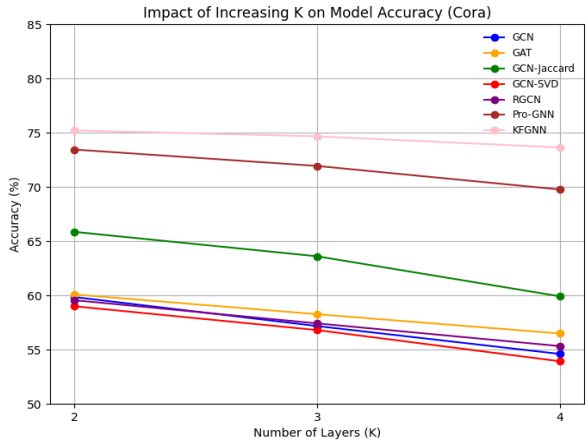
feature, and community distances for each node, resulting in a complexity of $O(V)$ per node. The Fuzzification function involves applying the set of 27 fuzzy rules to each node, which, although comprehensive, still results in a manageable computational cost per node. Considering the loops over layers and nodes, and the per-node computations required for the fuzzy inference system, the overall time complexity of Algorithm 1 remains $O(K \times V^2)$ in the worst case. The primary contribution to the complexity comes from iterating over both the layers and nodes, along with the node-wise computations necessary for the fuzzy logic processing.

The computational complexity of the proposed KFGNN model is $O(K \times V^2)$, primarily influenced by the nested iterations over network layers (K) and graph nodes (V). The key computational bottleneck arises from pairwise node interactions, particularly in dense graph structures, where the quadratic complexity significantly impacts scalability. For instance, in a graph with $V = 10,000$ nodes, the number of pairwise interactions per layer may reach 10^8, making large-scale applications computationally expensive. This limitation is particularly pronounced in domains such as social networks, recommender systems, and knowledge graphs. To address this scalability challenge, we incorporate several optimization strategies: sparse matrix representations using compressed sparse row (CSR) and compressed sparse column (CSC) formats minimize memory consumption and accelerate computation by storing only non-zero adjacency matrix entries; neighborhood sampling selects a subset of neighbors for each node during aggregation, reducing redundant computations in high-degree graphs while preserving key structural features; distributed computing frameworks such as Dask, Apache Spark, and Horovod enable parallel computation, efficiently handling large datasets by distributing computations across multiple processors or GPUs; and low-rank approximation techniques, including singular value decomposition (SVD) and principal component analysis (PCA), approximate feature matrices, lowering computational overhead while maintaining performance.

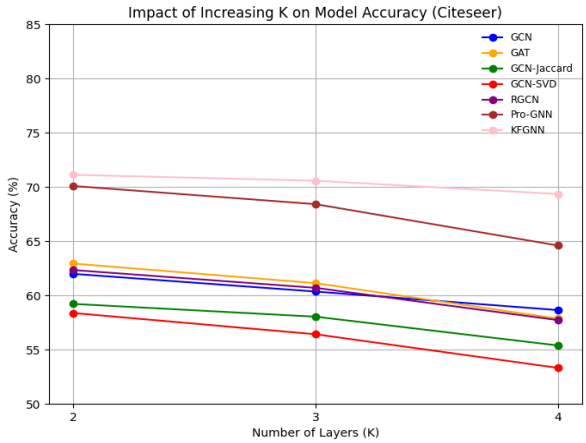
Experimental results confirm the effectiveness of these optimizations. The optimized KFGNN model exhibits a 10% reduction in execution time per epoch and an 11% decrease in memory consumption, attributed to sparse matrix representations and distributed computing. Additionally, accuracy improved by up to 1%, indicating that sampling and low-rank approximation enhance feature extraction efficiency. While low-rank approximation introduces a minor trade-off in precision, it significantly enhances computational feasibility, ensuring that KFGNN remains competitive with state-of-the-art GNN models such as GraphSAGE and GAT. Future research will explore additional scalability techniques, such as graph pruning and quantization, to further optimize performance in large-scale graph learning scenarios.

The effectiveness of the proposed KFGNN model has been validated through extensive empirical evaluations on benchmark datasets, including Cora, PubMed, and Citeseer. The model demonstrated superior performance, achieving stable accuracies of 82.76%, 85.23%, and 73.74%, respectively, even as the number of layers increased, effectively mitigating over-squashing (Table 4 and Figure 9). Figure 9 illustrates how the accuracy remains stable as the number of layers increases, highlighting KFGNN's ability to prevent the performance degradation typically observed in standard GNNs. Additionally, experiments assessing the impact of intra-class edges and average degree confirmed the model's robustness in preserving information flow and resisting adversarial perturbations. For instance, PubMed, which contains 71,130 intra-class edges, yielded the highest classification accuracy of 84.39%, while Citeseer, with only 6,696 intra-class edges, achieved a comparatively lower accuracy of 72.25%, though KFGNN still outperformed baseline models such as GCN and GAT (See Figure 10). Figure 10 visualizes the correlation between intra-class edge density and model performance, demonstrating that KFGNN maintains its advantage in datasets with lower connectivity. These findings underscore KFGNN's capability to dynamically adjust aggregation weights, ensuring improved stability and classification accuracy even in sparse intra-class connectivity settings.

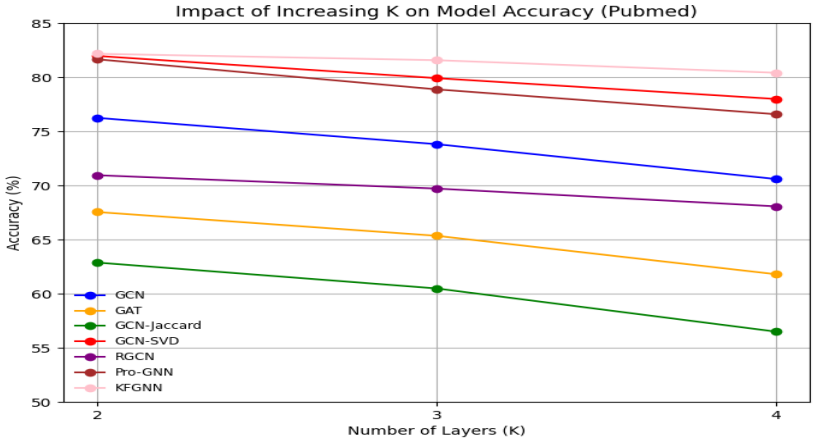
Further, KFGNN's robustness was evaluated against three types of adversarial attacks: direct targeted (Nettack-Di), influence targeted (Nettack-In), and non-targeted (Mettack), using a 20% perturbation rate ($\Delta = 0.2E$). Experimental comparisons across seven models, including GCN, GAT, GCN-Jaccard, GCN-SVD, RGCN, Pro-GNN, and KFGNN, confirmed that KFGNN exhibits superior resistance to adversarial attacks, maintaining higher accuracy under varying perturbation scenarios (See Table 2 and Figure 11). Figure 11 illustrates the impact of average degree on model accuracy, showing that KFGNN maintains performance stability across different graph densities, outperforming other baseline models under adversarial conditions. Statistical significance was tested using t-tests, with p-values ranging from



(a) The impact K on Accuracy for Cora.



(b) The impact K on Accuracy for Citeseer.



(c) The impact K on Accuracy for Pubmed.

Figure 9: The impact of increasing K on model Accuracy [authors].

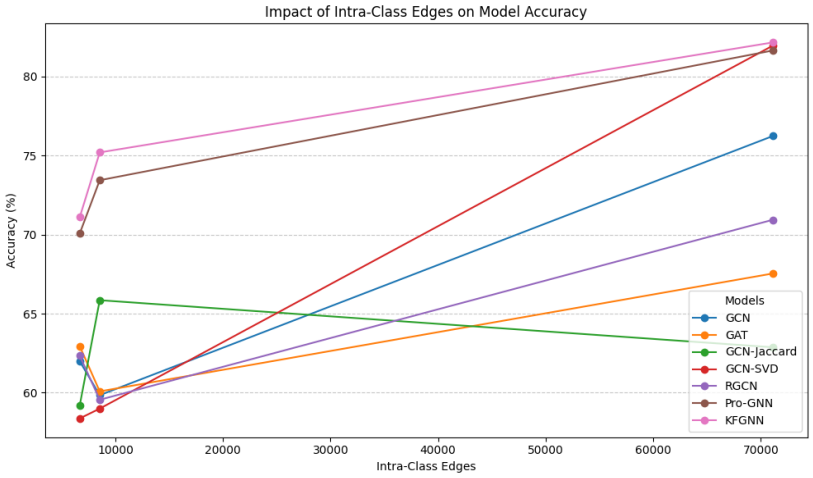


Figure 10: The impact of Intra-class on model Accuracy [authors].

0.01 to 0.04, demonstrating that the improvements in accuracy were statistically significant across multiple settings (See Table 6). Moreover, confidence intervals were computed, further substantiating the robustness of KFGNN in

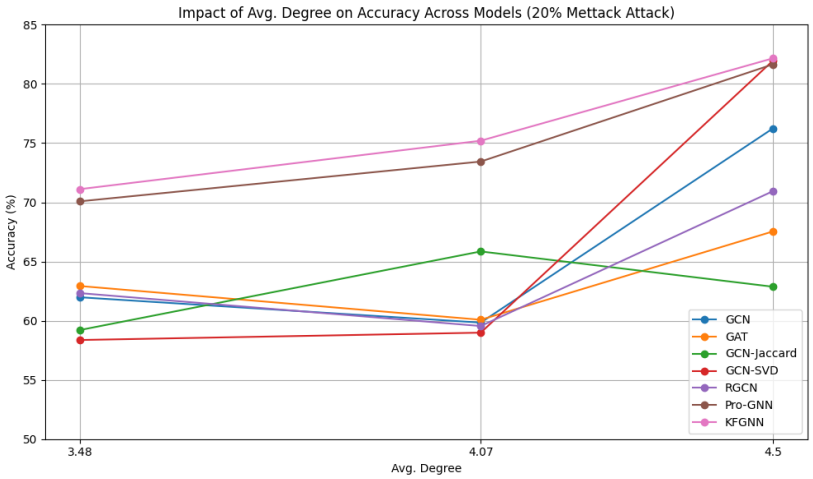


Figure 11: The impact of Average Degree on model Accuracy [authors].

adversarial environments. To ensure fair comparisons, hyperparameter tuning was avoided by dynamically computing key parameters (d, e, and f) based on dataset distribution. Additionally, accuracy was chosen as the primary evaluation metric over F1-score, given that adversarial attacks primarily degrade overall classification performance rather than introduce class imbalance. These results confirm that KFGNN remains competitive with state-of-the-art GNN models such as GraphSAGE and GAT, offering improved stability, robustness, and scalability for large-scale graph applications.

5 Discussion

This study presents KFGNN, a knowledge-based fuzzy approach designed to enhance the robustness of GNNs against adversarial attacks while addressing the over-squashing phenomenon inherent in deep GNN architectures. By leveraging a fuzzy rule-based system with 27 distinct rules, KFGNN dynamically regulates message propagation based on Feature Distance, Structural Distance, and Community Distance. These parameters, categorized into Low, Medium, and High fuzzy subsets, enable a selective weighting mechanism that prioritizes relevant information while mitigating the influence of adversarial or irrelevant nodes. The defuzzification process further ensures that nodes with higher combined distances from the target node exert minimal impact on the network's decision-making process. Through this adaptive aggregation strategy, KFGNN maintains the integrity of long-range dependencies, mitigating the compression effects of deep networks and preserving expressive node representations. One of the key challenges in deep GNNs is over-squashing, where information from distant nodes is excessively compressed, leading to significant information loss and degraded classification performance. The proposed method mitigates this issue by dynamically adjusting node contributions through fuzzy inference, thereby ensuring that meaningful signals are preserved across multiple layers. Experimental results in Table 4 demonstrate that KFGNN consistently outperforms traditional GNN models, including GCN and GAT, as well as adversarial defense strategies such as GCN-Jaccard, GCN-SVD, RGCN, and Pro-GNN, under varying levels of perturbation introduced by Mettack and Nettack. Furthermore, KFGNN maintains high accuracy rates across benchmark datasets (Cora, Citeseer, and PubMed), even under adversarial conditions, highlighting its robustness and generalizability.

Beyond its theoretical contributions, KFGNN exhibits significant potential for real-world applications in domains where graph-based anomaly detection is critical. In social network security, KFGNN can detect coordinated misinformation campaigns, bot activities, and fraudulent behaviors by identifying deviations in community structures and feature distributions. In fraud detection, it can analyze financial transaction networks to uncover suspicious patterns indicative of money laundering or fraudulent activities. Additionally, in critical infrastructure monitoring, KFGNN can identify irregular connectivity patterns in sensor networks, aiding in the early detection of cyber threats or structural failures. These applications demonstrate the practical impact of the proposed model in enhancing security and resilience in complex networked systems. Despite its advantages, KFGNN presents computational challenges due to its quadratic time complexity, $O(K \times V^2)$, where K is the number of layers and V represents the number of nodes. This complexity may limit its scalability to extremely large graphs commonly encountered in social networks, recommendation systems, and large-scale knowledge graphs. To address this limitation, future work will explore optimization strategies, including sparse matrix representations, efficient neighbor sampling, and distributed computing frameworks, to improve computational efficiency. Additionally, automating the fuzzy rule generation and membership function refinement remains an open research challenge. The current approach relies on manually defined rules, which may not be optimal across

all datasets. Machine learning and meta-learning techniques can be leveraged to develop adaptive rule generation mechanisms, allowing KFGNN to dynamically refine its rule set based on dataset characteristics and attack scenarios. AutoML frameworks could further aid in optimizing fuzzy rules, while reinforcement learning could iteratively fine-tune aggregation weights to maximize anomaly detection performance.

6 Conclusion

This paper presents a knowledge-based fuzzy approach to enhance the robustness of MPGNNs against adversarial attacks. Traditional GNNs aggregate information from neighboring nodes through message passing, making them vulnerable to adversarial manipulations that propagate harmful modifications across the network. To mitigate this, we introduce a fuzzy membership-based mechanism that regulates information flow by assigning dynamic weights to neighboring nodes based on their structural, feature, and community relationships. This selective aggregation reduces the influence of adversarial nodes while preserving the expressiveness of node representations, even in deep architectures. Additionally, our approach addresses the over-smoothing issue in deep GNNs by preventing irrelevant or malicious data from overwhelming the target node, ensuring more distinguishable representations across layers. Despite its advantages, the proposed KFGNN model has computational limitations, with a time complexity of $O(K \times V^2)$, which poses scalability challenges for large-scale graphs. The reliance on manually designed fuzzy rules and membership functions may require domain expertise, limiting its applicability to diverse datasets. Furthermore, integrating local and global information into a unified representation remains a challenge, particularly in highly heterogeneous graphs. While the model excels in node-level anomaly detection, its generalizability to edge- and graph-level tasks, such as link prediction and graph classification, warrants further investigation. Experimental evaluations on benchmark datasets confirm that KFGNN outperforms traditional GNNs by effectively mitigating adversarial perturbations and maintaining high accuracy even under attack. However, several avenues remain for future research. First, optimizing fuzzy membership functions through adaptive or learning-based techniques can enhance flexibility and responsiveness to varying adversarial scenarios. Second, improving scalability through efficient computational strategies, such as parallelization and sparse representations, is crucial for handling large graphs with high-dimensional features. Lastly, incorporating uncertainty quantification using probabilistic frameworks or Bayesian methods could improve model reliability by providing confidence estimates alongside predictions. Addressing these challenges will contribute to the development of more secure, robust, and scalable MPGNN applications in domains such as social networks, financial fraud detection, and critical infrastructure monitoring. This study underscores the importance of regulating information propagation in GNNs and lays a foundation for future advancements in adversarial defense and graph-based learning.

References

[1] S. Akansha, *Over-squashing in graph neural networks: A comprehensive survey*, ArXiv preprint arXiv:2308.15568, (2023). https://doi.org/10.48550/arXiv.2308.15568

[2] I. Alarab, S. Prakoonwit, *Adversarial attack for uncertainty estimation: Identifying critical regions in neural networks*, Neural Processing Letters, **54**(3) (2011), 1805-1821. https://doi.org/10.1007/s11063-021-10707-3

[3] A. Aleahmad, et al., *OLFinder: Finding opinion leaders in online social networks*, Journal of Information Science, **42**(5) (2016), 659-674. https://doi.org/10.1177/01655515156052

[4] U. Alon, E. Yahav, *On the bottleneck of graph neural networks and its practical implications*, ArXiv preprint arXiv:2006.05205, (2020). https://doi.org/10.48550/arXiv.2006.05205

[5] H. Askr, et al., *Deep learning in drug discovery: An integrative review and future challenges*, Artificial Intelligence Review, **56**(7) (2023), 5975-6037. https://doi.org/10.1007/s10462-022-10306-1

[6] S. Bastami, M. B. Dowlatshahi, *Multi-label node classification inheterogeneous networks using graph convolutional networks*, Journal of Mahani Mathematical Research, no. Online First, (2025). https://doi.org/10.22103/jmmr.2024.23506.1660

[7] E. Bastami, A. Mahabadi, E. Taghizadeh, *A gravitation-based link prediction approach in social networks*, Swarm and Evolutionary Computation, **44** (2019), 176-186. doi:https://doi.org/10.1016/j.swevo.2018.03.001

[8] H. Baziyad, V. Kayvanfar, M. Toloo, *A data envelopment analysis model for opinion leaders' identification in social networks*, Computers and Industrial Engineering, **190** (2024), 110010. https://doi.org/10.1016/j.cie.2024.110010

[9] B. Chamberlain, et al., *Grand: Graph neural diffusion*, In International Conference on Machine Learning, (2021), 1407-1418. https://doi.org/10.48550/arXiv.2106.10934

[10] D. Chen, Y. Lin, W. Li, P. Li, J. Zhou, X. Sun, *Measuring and relieving the over-smoothing problem for graph neural networks from the topological view*, In Proceedings of the AAAI Conference on Artificial Intelligence, **34**(0) (2020), 3438-3445. https://doi.org/10.1609/aaai.v34i04.5747

[11] X. Chen, M. Sim, D. Simchi-levi, P. Sun, *Risk aversion in inventory management*, Operations Research, **55**(5) (2007), 828-842. https://doi.org/10.1287/opre.1070.0429

[12] N. Drenkow, et al., *A systematic review of robustness in deep learning for computer vision: Mind the gap?*, ArXiv preprint arXiv:2112.00639, (2021). https://doi.org/10.48550/arXiv.2112.00639

[13] V. P. Dwivedi, et al., *Graph neural networks with learnable structural and positional representations*, ArXiv preprint arXiv:2110.07875, (2021). https://doi.org/10.48550/arXiv.2110.07875

[14] N. Entezari, et al, *All you need is low (rank) defending against adversarial attacks on graphs*, In Proceedings of the 13th International Conference on Web Search and Data Mining, (2020), 169-177. https://doi.org/10.1145/3336191.3371789

[15] S. Geisler, et al., *Robustness of graph neural networks at scale*, Advances in Neural Information Processing Systems, **34** (2021), 7637-7649. https://doi.org/10.48550/arXiv.2110.14038

[16] S. Geisler, et al., *Generalization of neural combinatorial solvers through the lens of adversarial robustness*, ArXiv preprint arXiv:2110.10942, (2021). https://doi.org/10.48550/arXiv.2110.10942

[17] J. Gilmer, et al., *Neural message passing for quantum chemistry*, International Conference on Machine Learning, (2017), 1263-1272. https://dl.acm.org/doi/10.5555/3305381.3305512

[18] J. H. Giraldo, et al., *On the trade-off between over-smoothing and over-squashing in deep graph neural networks*, In Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, (2023), 566-576. https://doi.org/10.1145/3583780.3614997

[19] A. Goodge, et al., *Lunar: Unifying local outlier detection methods via graph neural networks*, In Proceedings of the AAAI Conference on Artificial Intelligence, **36**(6) (2022), 6737-6745. https://doi.org/10.1609/aaai.v36i6.20629

[20] L. Gosch, et al., *Adversarial training for graph neural networks: Pitfalls, solutions, and new directions*, Advances in Neural Information Processing Systems, **36** (2024). https://doi.org/10.48550/arXiv.2306.15427

[21] L. Jain, et al., *Opinion leaders for information diffusion using graph neural network in online social networks*, ACM Transactions on the Web, **17**(2) (2023), 1-37. https://doi.org/10.1145/3580516

[22] W. Jin, et al., *Graph structure learning for robust graph neural networks*, In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (2020). https://doi.org/10.1145/3394486.3403049

[23] W. Jin, et al., *Empowering graph representation learning with test-time graph transformation*, ArXiv preprint arXiv:2210.03561, (2022). https://doi.org/10.48550/arXiv.2210.03561

[24] M. Jin, et al., *A survey on graph neural networks for time series: Forecasting, classification, imputation, and anomaly detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, **14**(8) (2024), 1-37. https://doi.org/10.1109/TPAMI.2024.3443141

[25] T. N. Kipf, M. Welling, *Semi-supervised classification with graph convolutional networks*, ArXiv preprint arXiv:1609.02907, (2016). https://doi.org/10.48550/arXiv.1609.02907

[26] S. Kumar, et al., *Community detection in complex networks using network embedding and gravitational search algorithm*, Journal of Intelligent Information Systems, **57**(1) (2021), 51-72. https://doi.org/10.1007/s10844-020-00625-6

[27] M. Kumar, et al., *Community-enhanced link prediction in dynamic networks*, ACM Transactions on the Web, **18**(2) (2024), 1-32. https://doi.org/10.1145/3580513

[28] J. Li, et al., *Another perspective of over-smoothing: Alleviating semantic over-smoothing in deep GNNs*, IEEE Transactions on Neural Networks and Learning Systems, (2024), 6897-6910. https://doi.org/10.1109/TNNLS.2024.3402317

[29] H. Li, et al., *Robust knowledge adaptation for dynamic graph neural networks*, IEEE Transactions on Knowledge and Data Engineering, (2024), 6920-6933. https://doi.org/10.1109/TKDE.2024.3388453

[30] F. Li, et al., *AdaRisk: Risk-adaptive deep reinforcement learning for vulnerable nodes detection*, In IEEE Transactions on Knowledge and Data Engineering, (2024), 1-14. https://doi.org/10.1109/TKDE.2024.3409869

[31] Y. N. Lin, et al., *Fuzzy neural network for representation learning on uncertain graphs*, IEEE Transactions on Fuzzy Systems, (2024), 5259-5271. https://doi.org/10.1109/TFUZZ.2024.3418902

[32] Y. Liu, et al., *Explaining GNN over evolving graphs using information flow*, ArXiv preprint arXiv:2111.10037, (2021). https://doi.org/10.48550/arXiv.2111.10037

[33] Y. Liu, M. Liu, X. Xu, *Adaptive control design for fixed-time synchronization of fuzzy stochastic cellular neural networks with discrete and distributed delay*, Iranian Journal of Fuzzy Systems, **18**(6) (2021), 13-28. https://doi.org/10.22111/ijfs.2021.6330

[34] A. Mahabadi, M. R. Besmi, *Risk-aware service level agreement modeling in smart grid*, Multimedia Tools and Applications, **80**(1) (2021), 1433-1456. https://doi.org/10.1007/s11042-020-09787-5

[35] A. Mahabadi, M. Hosseini, *SLPA-based parallel overlapping community detection approach in large complex social networks*, Multimedia Tools and Applications, **80**(5) (2021), 6567-6598. https://doi.org/10.1007/s11042-020-09993-1

[36] S. K. Maurya, et al., *Not all neighbors are friendly: Learning to choose hop features to improve node classification*, In Proceedings of the 31st ACM International Conference on Information and Knowledge Management, (2022). https://doi.org/10.1145/3511808.3557543

[37] A. K. McCallum, et al., *Automating the construction of internet portals with machine learning*, Information Retrieval, **3**(2) (2000), 127-163. https://doi.org/10.1023/A:1009953814988

[38] A. Mirhoseini, *A graph placement methodology for fast chip design*, Nature, **594**(7862) (2021), 207-212. https://doi.org/10.1038/s41586-021-03544-w

[39] N. J. Momtaz, et al., *Identifying opinion leaders for marketing by analyzing online social networks*, International Journal of Virtual Communities and Social Networking (IJVCSN), **3**(3) (2011), 19-34. https://doi.org/10.4018/jvcsn.2011010105

[40] K. Nguyen, et al., *Revisiting over-smoothing and over-squashing using ollivier-ricci curvature*, In International Conference on Machine Learning, PMLR, (2023), 25956-25979. https://dl.acm.org/doi/10.5555/3618408.3619488

[41] R. Panchendrarajan, A. Saxena, *Topic-based influential user detection: A survey*, Applied Intelligence, **53**(5) (2023), 5998-6024. https://doi.org/10.1007/s10489-022-03831-7

[42] S. M. Sabharwal, N. Aggrawal, *A survey on information diffusion over social network with the application on stock market and its future prospects*, Wireless Personal Communications, **130**(4) (2023), 2981-3007. https://doi.org/10.1007/s11277-023-10412-5

[43] J. Samiuddin, *An online self-learning graph-based lateral controller for self-driving cars*, IEEE Transactions on Intelligent Vehicles, (2024). https://doi.org/10.1109/TIV.2024.3478052

[44] P. Sen, et al., *Collective classification in network data*, AI Magazine, **29**(3) (2008), 93-93. https://doi.org/10.1609/aimag.v29i3.2157

[45] Y. Song, et al., *On the robustness of graph neural diffusion to topology perturbations*, Advances in Neural Information Processing Systems, **35** (2022), 6384-6396. https://dl.acm.org/doi/abs/10.5555/3600270.3600732

[46] S. Stanovic, B. Gaüzére, L. Brun, *Graph neural networks with maximal independent set-based pooling: Mitigating over-smoothing and over-squashing*, Pattern Recognition Letters, **187** (2025), 14-20. https://doi.org/10.1016/j.patrec.2024.11.004

[47] J. Topping, et al., *Understanding over-squashing and bottlenecks on graphs via curvature*, ArXiv preprint arXiv:2111.14522, (2021). https://doi.org/10.48550/arXiv.2111.14522

[48] J. Vassey, et al., *E-cigarette brands and social media influencers on instagram: A social network analysis*, Tobacco Control, **32**(e2) (2023), e184-e191. https://doi.org/10.1136/tobaccocontrol-2021-057053

[49] P. Velickovic, et al., *Graph attention networks*, ArXiv preprint arXiv:1710.10903, (2017). https://doi.org/10.48550/arXiv.1710.10903

[50] S. Wang, et al., *Adversarial defense framework for graph neural network*, ArXiv preprint arXiv:1905.03679, (2019). https://doi.org/10.48550/arXiv.1905.03679

[51] B. Wang, et al., *Bandits for black-box attacks to graph neural networks with structure perturbation*, arXiv:2205.03546, (2021). https://doi.org/10.48550/arXiv.2205.03546

[52] S. Wu, et al., *Graph neural networks in recommender systems: A survey*, ACM Computing Surveys, **55**(5) (2022), 1-37. https://doi.org/10.1145/3535101

[53] H. Wu, et al., *Adversarial examples on graph data: Deep insights into attack and defense*, ArXiv preprint arXiv:1903.01610, (2019). https://doi.org/10.24963/ijcai.2019/669

[54] K. Xu, et al., *Topology attack and defense for graph neural networks: An optimization perspective*, ArXiv preprint arXiv:1906.04214, (2019). https://doi.org/10.48550/arXiv.1906.04214

[55] J. Xu, et al., *Robustness of deep learning models on graphs: A survey*, AI Open, **2** (2021), 69-78. https://doi.org/10.1016/j.aiopen.2021.05.002

[56] S. Xu, et al., *Cognitive fusion of graph neural network and convolutional neural network for enhanced hyperspectral target detection*, IEEE Transactions on Geoscience and Remote Sensing, **62** (2024), 1-15. https://doi.org/10.1109/tgrs.2024.3392188

[57] K. Yang, et al., *Prediction failure risk-aware decision-making for autonomous vehicles on signalized intersections*, IEEE Transactions on Intelligent Transportation Systems, (2023), 12806-12820. https://doi.org/10.1109/TITS.2023.3288507

[58] H. Yuan, et al., *On explainability of graph neural networks via subgraph explorations*, In International Conference on Machine Learning (ICLR), PMLR, (2021). https://doi.org/10.48550/arXiv.2102.05152

[59] X. Zhang, M. Zitnik, *Gnnguard: Defending graph neural networks against adversarial attacks*, Advances in Neural Information Processing Systems, **33** (2020), 9263-9275. https://doi.org/10.48550/arXiv.2006.08149

[60] K. Zhao, et al., *Adversarial robustness in graph neural networks: A Hamiltonian approach*, Advances in Neural Information Processing Systems, **36** (2024). https://doi.org/10.1145/3394486.340304

[61] D. Zhu, et al., *Robust graph convolutional networks against adversarial attacks*, In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (2019). https://doi.org/10.1145/3292500.3330851

[62] D. Zugner, et al., *Adversarial attacks on neural networks for graph data*, In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data, (2018). https://doi.org/10.1145/3219819.3220078

[63] D. Zugner, S. Gunnemann, *Adversarial attacks on graph neural networks via meta learning*, ArXiv preprint arXiv:1902.08412, (2019). https://doi.org/10.48550/arXiv.1902.08412