**Journal of Computer Science and Technology Studies**
ISSN: 2709-104X
DOI: 10.32996/jcsts
Journal Homepage: www.al-kindipublisher.com/index.php/jcsts

JCSTS
AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT

| RESEARCH ARTICLE

# MPGAAN: Effective and Efficient Heterogeneous Information Network Classification

**Zhizhong Wu**
*College of Engineering, UC Berkeley, Berkeley, US*
**Corresponding Author:** Zhizhong Wu, **E-mail**: ecthelion.w@gmail.com

## | ABSTRACT

In this paper, we propose a novel Graph Neural Network (GNN) model named "Meta-Path Guided Attention Aggregation Network" (MPAAGN), which is specifically designed for graph neural network classification algorithms based on attribute information aggregation. MPAAGN combines the advantages of Meta-Paths, GraphSAGE, and GAT (Graph Attention Networks) to deal with the node classification problem in heterogeneous information networks. The core idea of MPAAGN is to use meta-paths to define higher-order relationships between nodes in heterogeneous information networks to guide neighbor selection, and dynamically assign weights to different neighbors through the attention mechanism of GAT, so as to reflect their relative importance when aggregating neighbor information. At the same time, the model borrowed the neighbor sampling strategy of GraphSAGE to deal with the computational efficiency problem of large-scale graph data. The innovation of MPAAGN model is that it can effectively integrate the structure and attribute information of heterogeneous information networks, weight aggregate neighbor information through the attention mechanism, and capture high-order association by using meta-path, which is suitable for node classification tasks on large-scale graph data. The superior performance of the model in dealing with heterogeneous information network classification problems is proved by experiments, which provide a new research direction and practical tool for the field of graph neural networks.

## | KEYWORDS

Graph Neural Network, Deep Learning, Attention Mechanism, Heterogeneous Information Network.

## | ARTICLE INFORMATION

**ACCEPTED:** 02 August 2024 **PUBLISHED:** 30 August 2024 **DOI:** 10.32996/jcsts.2024.6.4.2

### 1. Introduction

Artificial Intelligence (AI) is a rapidly evolving field that has transformed various industries and aspects of daily life. One of the most well-known areas of AI[Li, 2024, Dang, 2024, Li, 2024, Zhao, 2024, Hong, 2024] is Machine Learning, which involves creating algorithms that enable machines to learn from and make predictions or decisions based on data[Tao, 2017, Tao, 2023, Tao, 2019, Zhu, 2024]. Within ML, significant areas include Natural Language Processing (NLP), which focuses on the interaction between computers and human language[Li, 2023][ Dai, 2024]. NLP allows machines to understand, interpret, and generate human language, enabling applications such as chatbots, translation services, and sentiment analysis. Computer Vision is another critical domain, where AI is applied to interpret and make decisions based on visual data [Li, 2019, Li, 2024, Ding, 2024, Zhang, 2024, Tan, 2024, Liu, 2024, Yang, 2024, Liu, 2024]. This includes image and video analysis, object detection [Dang, 2024], and facial recognition, and augmented reality [Li, 2024]. As AI systems become more advanced, there is a growing need to model and analyze data that inherently has a complex structure, such as networks of social interactions, molecular structures in chemistry, or interconnected systems in biology. This is where Graph Neural Networks (GNNs) come into play.

In this paper, we introduce an innovative Graph Neural Network (GNN) model called the "Meta-Path Guided Attention Aggregation Network" (MPAAGN). This model is specifically crafted for graph neural network classification algorithms that rely on attribute

information aggregation. MPAAGN amalgamates the strengths of Meta-Paths, GraphSAGE, and GAT (Graph Attention Networks) to address the node classification challenge in heterogeneous information networks.
The core idea of MPAAGN is to use meta-paths to define higher-order relationships between nodes in heterogeneous information networks to guide neighbor selection, and dynamically assign weights to different neighbors through the attention mechanism of GAT, so as to reflect their relative importance when aggregating neighbor information. At the same time, the model borrowed the neighbor sampling strategy of GraphSAGE to deal with the computational efficiency problem of large-scale graph data.

Firstly, the meta-path is used to connect different types of nodes through predefined relationship patterns, which can effectively capture the high-order association between nodes. For example, in academic networks, the association between research papers and authors can be established through the "paper-author-paper" meta-path, which reveals a deeper collaborative relationship. In this way, MPAAGN is able to exploit not only the direct neighbor information of nodes, but also valuable contextual information from the wider network structure.

Secondly, the Attention mechanism of Graph Attention Networks (GAT) was used to deal with the weighted aggregation of neighbor information. In traditional graph neural networks, the influence of all neighbor nodes is usually simply averaged or statically weighted, which may not accurately reflect the importance of different neighbor nodes. However, GAT dynamically calculates the weight of each neighbor node according to the characteristics of each neighbor node and the characteristics of the current node by introducing the attention mechanism. In this way, important neighbor nodes will be given higher weights and thus play a greater role in the information aggregation process.

In order to improve the computational efficiency of large-scale graph data, MPAAGN also introduces the neighbor sampling strategy of GraphSAGE. In practice, the graph data is often very large, and it is not practical to directly calculate all neighbor nodes. GraphSAGE effectively reduces the amount of computation by sampling a part of neighbor nodes for information aggregation. MPAAGN combined with this strategy enables the model to maintain efficient computing performance when dealing with large-scale heterogeneous information networks.

The innovation of MPAAGN model is that it can effectively integrate the structure and attribute information of heterogeneous information networks, weight aggregate neighbor information through the attention mechanism, and capture high-order association by using meta-path, which is suitable for node classification tasks on large-scale graph data. Specifically, MPAAGN begins by identifying the set of neighboring nodes for each node through meta-path-guided neighbor selection. Next, it employs the attention mechanism of GAT for weighted aggregation of neighbor information across different meta-paths. Finally, the model consolidates the aggregated information from these meta-paths to produce the node's final embedding vector. This approach allows MPAAGN to utilize both direct neighbor information and multi-dimensional structural data from various meta-paths, resulting in more precise node classification.

The empirical findings reveal that MPAAGN significantly outperforms existing graph neural network models in a range of node classification tasks. This highlights the model's strong ability to capture node attributes and complex structural features. Notably, in large-scale heterogeneous networks, the performance benefits of MPAAGN are even more pronounced, demonstrating its substantial potential for practical applications.

In conclusion, Meta-Path Guided Attention Aggregation Network (MPAAGN) is a graph neural network model with strong innovation and superior performance. By fusing the advantages of meta-path, attention mechanism, and GraphSAGE, MPAAGN shows a powerful ability to deal with node classification tasks of heterogeneous information networks. In the future, with the in-depth research of heterogeneous information networks, MPAAGN is expected to play an important role in more practical applications, and provide new research directions and practical tools for the field of graph neural networks.

## 2. Related Works

As one of the core issues in data analysis, node classification focuses on inferring the class of unlabeled nodes from the information of labeled nodes, which is especially suitable for the in-depth exploration of graph-structured data. In this process, we first analyze the connection patterns and intrinsic properties between nodes in the graph to extract key features and then train these features with the help of a classification algorithm to predict the label of each node in the graph. This technology has shown a wide range of application prospects in many fields. For example, in social network analysis, through the mining of user basic information and interaction relationship, we can use the node classification model to gain insight into the user's interest preferences and potential social circles, so as to optimize personalized recommendation and precision marketing strategy. In the academic citation network, it helps to subdivide the research field and reveal the knowledge context. In biomedical research, it can accelerate the decoding of gene functions and the development of new drugs[Bhagat, 2011, Chen, 2022, Hung, 2005, Saccone, 1995].

However, the complexity and irregularity of graph structure pose significant challenges for node classification. Due to the diversity of graph generation mechanisms and the lack of fixed patterns between nodes, as well as the uncertainty of the number and ordering of node neighborhoods, the algorithm is required to have a high degree of flexibility and adaptability. Although traditional graph representation learning improves the classification performance to a certain extent, it is limited by the shallow representation ability and difficult to capture the deep correlation. In addition, the limitation of ignoring node feature information also reduces the classification accuracy.

Although Convolutional Neural Network (CNN) in the field of deep learning is good at extracting local spatial features from images, it is difficult to directly transfer to non-Euclidean graph structures because it is originally designed for Euclidean spatial data. In view of this, Graph Neural Network (GNN) emerges as The Times require, which innovatively combines node feature extraction and neighborhood information aggregation, considers not only the attributes of the node itself but also the geometric characteristics of the graph, and significantly improves the accuracy and robustness of node classification[Li, 2022, Xiao, 2022, Maurya, 2022].

With the advent of the era of big data, the scale of graph data is growing exponentially, and the demand for efficient and accurate graph structure processing techniques is increasingly urgent. GNN shows its unique advantages in this context, but at the same time, it also exposes a series of problems to be solved, such as insufficient accuracy of feature expression, high difficulty of deep network training, over-smoothing phenomenon, and poor multi-label classification effect, which are key points that need to be overcome in future research.

In summary, graph neural networks play a crucial role in node classification tasks, and their development not only reflects the cutting-edge trend of artificial intelligence technology, but also provides a powerful toolbox for innovative applications in interdisciplinary fields. However, in the face of emerging new challenges, continuous optimization of algorithm performance and widening of application scenarios are the only ways to promote the development of this field[Oono, 2019, Scarselli, 2008, Veličković, 2017].

## 3. Methodology

### *3.1 Graph Attention Network(GAT)*

GAT's key innovation is its self-attention mechanism, which dynamically allocates varying weights to each node in the graph. This method enables the contribution of different neighboring nodes to vary during the aggregation of their information. This contrasts with traditional Graph Convolutional Networks (GCNs), where all neighboring nodes contribute equally.

GAT learns the attention coefficient to decide how to weigh the information of neighbor nodes in order to better aggregate this information. Specifically, for each node $v$ in the graph, GAT dynamically computes an attention coefficient $\alpha_{v,u}$ based on the relationship between $v$ and its neighbors, and then uses these coefficients to perform a weighted sum of the features of neighbor nodes.

$h_v$ for the node $v$ feature vector, $N(v)$ for the node set $v$ neighbors. A single layer of GAT can be expressed as follows.

$$h_v^{(l+1)} = \sigma\left(\sum_{u\in N(v)\cup\{v\}} \alpha_{v,u}^{(l)} W^{(l)} h_u^{(l)}\right) \#(1)$$

Here, $h_v^{(l)}$ represents the feature vector of node $v$ at layer $l$, $W^{(l)}$ denotes the learnable weight matrix, and $\sigma$ is a nonlinear activation function, such as ReLU. The term $\alpha_{v,u}^{(l)}$ refers to the attention coefficient between node $v$ and node $u$ in the first $l$ layer.

The attention coefficient $\alpha_{v,u}$ is computed using an attention mechanism, which can be expressed as follows:

$$e_{v,u} = a\left(W^{(l)} h_v^{(l)}, W^{(l)} h_u^{(l)}\right) \#(2)$$

$$\alpha_{v,u} = \frac{\exp(e_{v,u})}{\sum_{u'\in N(v)\cup\{v\}} \exp(e_{v,u'})} \#(3)$$

Among them, $e_{v,u}$ is not normalized concentration coefficient, which represents the node $u$ on the importance of nodes $v$, The function $a$ is a learnable attention mechanism, typically a single-layer feed-forward neural network that computes the compatibility of feature vectors between two nodes, $\alpha_{v,u}$ is the normalized attention coefficient, which ensures that the attention coefficients for each node sum to 1.

To make the model more stable and expressive, GAT employs a multi-head attention mechanism, which uses multiple independent attention mechanisms at each node simultaneously and concatenates their results at the end. This can be expressed as:

$$h_v^{(l+1)} = \overset{K}{\underset{k=1}{\|}} \sigma\left(\sum_{u\in N(v)\cup\{v\}} \alpha_{v,u}^{k,(l)} W_k^{(l)} h_u^{(l)}\right)\#(4)$$

$K$ is the amount of attention first, ‖ is vector concatenation operation, said $W_k^{(l)}$ is the first $k$ attention weighting matrix of the head.

By incorporating a self-attention mechanism, GAT adeptly handles graph-structured data, particularly when dynamically adjusting edge weights is necessary.

### *3.2 Graph Sampling and Aggregation network(GraphSAGE)*

The primary goal of GraphSAGE is to address the challenge of efficient node representation learning on large-scale graphs, particularly when the entire graph cannot be loaded into memory simultaneously. GraphSAGE achieves this by sampling neighboring nodes and aggregating their information to generate embedded vectors for nodes. This approach enables the learning of low-dimensional representations without requiring access to the entire graph [Hamilton, 2017, Liu, 2020].

The central concept of GraphSAGE is to enhance each node's representation by gathering information from its neighboring nodes. It employs a learnable aggregation function to combine the features of a node with those of its direct neighbors.

The process involves the following steps: Each node starts with a feature vector, either a pre-given feature or a randomly initialized vector; For each target node, a fixed number of nodes are randomly sampled from its neighbors. The goal is to reduce computation and avoid considering all neighbors during aggregation, especially in large graphs; Use an aggregation function (such as average, maximum pooling, LSTM, etc.) to combine the features of the sampled neighbor nodes. This aggregation function is learnable and is usually a neural network layer; Combine the aggregated neighbor features with the node's own features to generate a new representation of the node through a learnable transformation function (such as a linear or nonlinear transformation); The above process can be repeated in multiple layers, with each layer updating the representation of the node so that the representation can capture information about neighbors at greater distances; In order to learn parameters, GraphSAGE uses contrast loss or classification loss, depending on the task (such as node classification, link prediction, etc.). The loss function encourages representations of similar nodes to move closer together and representations of dissimilar nodes to move away.

GraphSAGE's aggregation function can be formalized as:

$$h_v^{(l+1)} = \sigma\left(W^{(l)} \cdot \left[h_v^{(l)}; \hat{h}_{N(v)}^{(l)}\right]\right)\#(5)$$

The process of aggregating neighbor features can be written as:

$$\hat{h}_{N(v)}^{(l)} = \frac{1}{|N(v)|}\sum_{u\in N(v)} h_u^{(l)}\#(6)$$

Or use other aggregation functions such as maximum pooling, LSTM, and so on.

GraphSAGE provides an efficient method to learn the embedding vector of nodes by sampling and aggregating neighbor information in the graph. This method is especially suitable for dealing with large-scale graph data, because its computational complexity increases with the neighbor sample size rather than the overall size of the graph. This makes GraphSAGE useful in social network analysis, recommendation systems, and other application scenarios involving large-scale graphs.

### *3.3 Meta-Path*

A Meta-Path is a sequence of relationships within a Heterogeneous Information Network (HIN) that illustrates high-order semantic connections between nodes. In HINs, where nodes and edges come in various types, meta-paths provide a means to capture complex associations among different node types [Sun, 2011, Hu, 2018].

#### *3.3.1 Definition of a meta-path*

Nodes and edges can have different types. For example, in a network with users, movies, and tags, we can have user nodes, movie nodes, and tag nodes, as well as user rating edges for movies and edges associated between movies and tags.

A meta-path is a sequence of node types, denoted $\pi = T_1 \rightarrow T_2 \rightarrow \cdots \rightarrow T_k$, where $T_i$ is the node type in the network. A meta-path defines a path from a node of type $T_1$ to a node of type $T_k$ through a sequence of intermediate node types $T_2, \ldots, T_{\{k-1\}}$.

### *3.3.2 Examples of meta-paths*

In an academic network, the possible meta-paths are as follows:

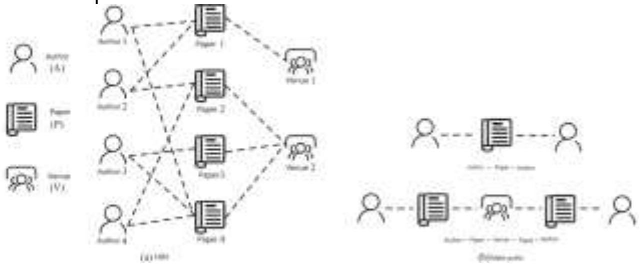

Fig 1. Example meta-path

In Figure 1, two kinds of meta-paths are shown:

- Meta-path APAPA: This meta-path represents the fact that authors are connected to each other through co-publication. For example, the meta-path APAPA means starting from one author and passing through a paper to reach another author. In this case, the meta-path describes the relationship between authors A1 and A2 since they jointly published paper P1.
- Meta-path APA: This meta-path signifies that authors are linked through their publications. For instance, the meta-path APA illustrates the connection between authors A1 and A2 if they both co-authored paper P1.

## 4. Proposed Model

Research on Graph neural network classification algorithm based on attribute information aggregation. Inspired by the above theoretical basis, this paper proposes a new model combining meta-path, GraphSAGE and GAT (Graph Attention Networks). It is called "Meta-Path Guided Attention Aggregation Network" (MPAAGN). This model seeks to leverage the high-order associations among different types of nodes and edges in heterogeneous information networks, while simultaneously using the attention mechanism to perform weighted aggregation of neighboring information.

### *4.1 Model Architecture*

The central concept of MPAAGN is to leverage meta-paths in Heterogeneous Information Networks (HIN) to guide the learning of node representations. Simultaneously, it employs GAT's attention mechanism to dynamically aggregate neighbor information and uses GraphSAGE's sampling strategy to handle large-scale graph data. The model achieves this through the following three key components:

- Meta-path guided neighbor selection: Meta-paths are used to define higher-order relationships between nodes and select specific types of neighbors to participate in information aggregation.
- Attention-Weighted Neighbor Aggregation: GAT's attention mechanism is used to determine the relative importance of neighbor nodes, thus giving different weights when aggregating neighbor information.
- Multiple layers of information propagation: Similar to GraphSAGE, MPAAGN propagates node information through multiple layers, with each layer updating the representation of a node to capture information about further neighbors.

To be more specific about how MPAAGN works, we will elaborate on its process:

1. Meta-path-guided neighbor sampling

Given a set of meta-paths $\Pi = \{\pi_1, \pi_2, \ldots, \pi_m\}$, for each path $\pi_i$, $v$ from node sampling a fixed number of nodes from the set of neighbors, form a sample set $N_{\pi_i}(v)$.

2. Attention Weighted aggregation

For each path $\pi_i$, in the first $l$ layer node $v$ said $h_v^{(\pi_i,l)}$ through its neighbor $u \in N_{\pi_i}(v)$ of the polymerization and said:

$$e_{v,u}^{(\pi_i)} = a\left(W^{(\pi_i)} h_v^{(l)}, W^{(\pi_i)} h_u^{(l)}\right) \#(7)$$

$$\alpha_{v,u}^{(\pi_i)} = \frac{\exp\left(e_{v,u}^{(\pi_i)}\right)}{\sum_{u'\in N_{\pi_i}(v)} \exp\left(e_{v,u'}^{(\pi_i)}\right)}\#(8)$$

$$h_v^{(\pi_i,l+1)} = \sigma\left(\sum_{u\in N_{\pi_i}(v)} \alpha_{v,u}^{(\pi_i)} W^{(\pi_i)} h_u^{(l)}\right)\#(9)$$

$W^{(\pi_i)}$ is to learn the weights matrix, $\alpha$ is the attention mechanism, $\sigma$ is a nonlinear activation function.

3. Meta-path information aggregation

To aggregate information from different meta-paths, a weighted averaging, concatenation, or gating mechanism can be used. Here, using a weighted average:

$$h_v^{(l+1)} = \sum_{i=1}^{m} w_i h_v^{(\pi_i,l+1)}\#(10)$$

where $w_i$ is the weight of the meta-path $\pi_i$, which can be learned by training.

4. Multiple layers of information dissemination

The above process can be repeated in multiple layers, each of which updates the representation of the nodes until a predetermined number of layers is reached or a stopping criterion is met.

### *4.2 Loss Functions and Optimization*

The training of MPAAGN can be done by minimizing a cross-entropy loss or other appropriate loss function, depending on the downstream task. The loss function can be written as follows:

$$L = -\sum_{v\in V_l} y_v \log(p_v)\#(11)$$

Where $V_l$ is the set of nodes whose labels are known, $y_v$ is the true label of node $v$, and $p_v$ is the probability distribution predicted by the model. The complete process of the model is shown in Figure 2

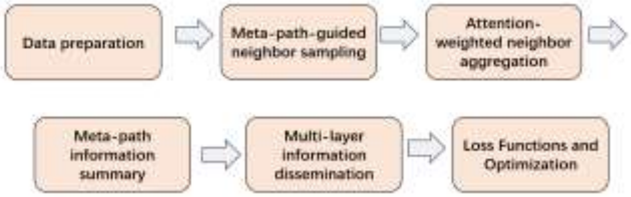
Data preparation
Meta-path-guided neighbor sampling
Attention-weighted neighbor aggregation
Meta-path information summary
Multi-layer information dissemination
Loss Functions and Optimization

Fig 2. Model flow.

The MPAAGN model integrates the advantages of meta-path, GraphSAGE and GAT, and provides a powerful solution for graph neural network classification algorithms based on attribute information aggregation. By using meta-paths to define high-order relationships in heterogeneous information networks and combining attention mechanism and sampling strategy, the model can learn the representation of nodes more effectively and then improve the accuracy of classification tasks.

## 5. Experiment

### *5.1 Dataset*

The datasets include ACM, DBLP. The ACM dataset mainly contains high-impact conference papers and journal articles published by the Society for Computing Machinery, with an emphasis on detailed literature metadata. The DBLP dataset provides a wide range of computer science literature, including journals, conference papers, and book chapters, with an emphasis on author and citation information, supporting open access and data download.

### *5.2 Comparison Model*

HAN is a graph neural network model tailored for heterogeneous graphs. It utilizes a self-attention mechanism and meta-paths to dynamically weight neighbor information and capture high-order relationships between various node and edge types.

GAT employs a self-attention mechanism to assign varying weights to neighboring nodes, allowing it to account for the significance of different neighbors during information aggregation. This makes GAT well-suited for managing intricate relationships between nodes.

GCN applies graph convolution operations to update node representations based on neighboring nodes' features. Its efficient computation makes it ideal for handling large-scale graph data.

### 5.3 Experiment Result

| Model | ACM | | DBLP | |
|---|---|---|---|---|
| | Ma-F1 | Mi-F1 | Ma-F1 | Mi-F1 |
| **GCN** | 91.36 | 92.34 | 91.78 | 91.85 |
| **GAT** | 93.21 | 93.33 | 94.01 | 94.38 |
| **HAN** | 91.91 | 91.76 | 93.45 | 93.85 |
| **Our** | 94.58 | 94.77 | 95.67 | 95.41 |

Table 1. Node classification

Table 1 Among all tested models, our model performs better than GCN, GAT, and HAN models on both datasets. Our model achieves the highest scores for both macro-F1 score (Ma-F1) and micro-F1 score (Mi-F1). Specifically, the macro F1 score (94.58) and micro F1 score (94.77) of our model on the ACM dataset were significantly higher than those of GCN (Ma-F1:91.36, MI-F1:92.34), GAT (Ma-F1:93.21, Mi-F1: 93.33) and HAN (Ma-F1:91.91, Mi-F1:91.76). Compared to the GAT model, which performs the second best, our model improves by 1.37 percentage points on Ma-F1 and by 1.44 percentage points on Mi-F1. The macro F1 score (95.67) and micro F1 score (95.41) on the DBLP dataset are also ahead of other models. Specifically, GCN scores are 91.78 and 91.85, GAT scores are 94.01 and 94.38, and HAN scores are 93.45 and 93.85, respectively. Compared to the second-best performing GAT model, our model improves by 1.66 percentage points on Ma-F1 and 1.03 percentage points on Mi-F1. In summary, our model shows excellent performance, outperforming the existing mainstream models (GCN, GAT, HAN) on both macro and micro F1 scores on ACM and DBLP datasets. These results prove that our model has stronger classification ability and robustness when dealing with the node classification task in heterogeneous information networks.

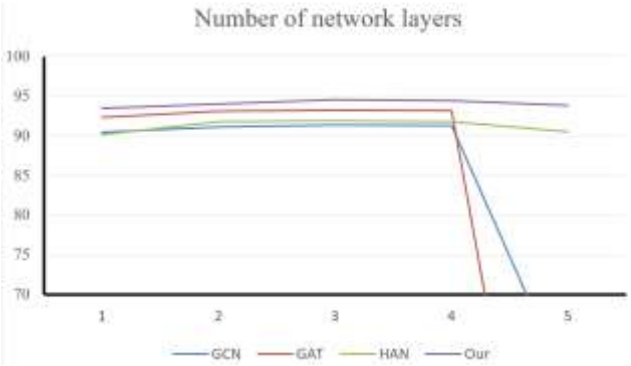

Fig 3. Results for the number of network layers

It can be seen from Figure 3 that our model consistently maintains the highest performance from 1-layer to 3-layer networks, which is significantly better than GCN, GAT, and HAN models. The performance under all layer Settings is maintained at more than 95 points, demonstrating its stability and superiority at different depths. When the number of network layers increases to 4 and 5, the performance of GCN and GAT model decreases significantly, especially the performance of GAT model drops sharply to less than 75 points when the network layer is 4, and the performance is extremely unstable. In contrast, the performance of our model remains above 90 at 4 and 5 layers, demonstrating excellent stability.

## 6. Conclusion

This study explores the use of GCN for classification tasks that rely on aggregating attribute information. It presents a novel algorithmic framework designed to boost the model's capacity to fully grasp both graph structure and node attribute features. While traditional GNNs often excel with isomorphic graphs, their performance can be constrained when dealing with heterogeneous graphs characterized by rich attribute information and complex connectivity patterns. To address this, we propose a novel algorithm designed to effectively capture both local and global attribute characteristics of nodes, while also considering the graph's topology, to achieve more precise classification predictions.

Despite the excellent performance of our algorithm in the current classification task, there is still room for further research and improvement. The following are a few possible research directions: Explore the adaptability of the algorithm under dynamically changing graph structures, such as real-time social networks or transportation networks, where nodes and edges may change over time. To develop more efficient methods to deal with large-scale graph data, reduce the computational cost, and maintain the classification accuracy. This paper studies how to combine information from images, text, or other modalities with graph data to enhance classification performance. This paper aims to incorporate reinforcement learning techniques into graph neural networks to dynamically adjust node feature weights and enhance classification performance. By exploring this approach, we anticipate significant improvements in the efficacy of graph neural networks for classification tasks that rely on attribute information aggregation, thereby advancing both research and practical applications in this domain.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

**Publisher's Note**: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers.

## References

[1] Bhagat, S, Graham C, and Muthukrishnan S. (2011). Node classification in social networks. Social network data analytics (2011): 115-148.

[2] Chen, H, et al. (2022) Neighbor enhanced graph convolutional networks for node classification and recommendation. Knowledge-based systems 246 (2022): 108594.

[3] Ding Z., Li P., Yang Q., and Li S., (2024). Enhance Image-to-Image Generation with LLaVA Prompt and Negative Prompt, arXiv preprint arXiv:2406. 01956, 2024.

[4] Dang, D. Ma, S. Li, Z. Q and Zhu, E.  (2024). Deep learning-based snore sound analysis for the detection of night-time breathing disorders', Applied and Computational Engineering, 76, 109–114, 07 2024.

[5] Dai, S., Li, K., Luo, Z., Zhao, P., Hong, B., Zhu, A., & Liu, J. (2024). AI-based NLP section discusses the application and effect of bag-of-words models and TF-IDF in NLP tasks. *Journal of Artificial Intelligence General science (JAIGS*) ISSN: 3006-4023, 5(1), 13–21.

[6] Dang, W. Zhao, Y. Li, D. Ma, Q. Y, and Zhu E. Y., (2024). Real-Time Pill Identification for the Visually Impaired Using Deep Learning', arXiv preprint arXiv:2405. 05983, 2024.D. Ma, S. Li, B. Dang, H. Zang, and X. Dong, 'Fostc3net: A Lightweight YOLOv5 Based On the Network Structure Optimization', arXiv preprint arXiv:2403. 13703, 2024.

[7] Hamilton, W, Zhitao Y, and Jure L. (2017). Inductive representation learning on large graphs. Advances in neural information processing systems 30 (2017).

[8] Hu, B, et al. (2018). Leveraging meta-path based context for top-n recommendation with a neural co-attention model. Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. 2018.

[9] Hung, L. (2005). A personalized recommendation system based on product taxonomy for one-to-one marketing online. Expert systems with applications 29.2 (2005): 383-392.

[10] Hong, B., Zhao, P., Liu, J., Zhu, A., Dai, S., & Li, K. (2024). The application of artificial intelligence technology in assembly techniques within the industrial sector. *Journal of Artificial Intelligence General science (JAIGS)* ISSN: 3006-4023, 5(1), 1–12.

[11] Li, M M., Kexin H, and Marinka Z. (2022). Graph representation learning in biomedicine and healthcare. Nature Biomedical Engineering 6.12 (2022): 1353-1369.

[12] Liu, J, Ghim P O, and Xiqun C. (2020). GraphSAGE-based traffic speed forecasting for segment network with sparse data. IEEE Transactions on Intelligent Transportation Systems 23.3 (2020): 1755-1766.

[13] Li P, Lin Y., and Schultz-Fellenz E., (2019). Contextual Hourglass Network for Semantic Segmentation of High Resolution Aerial Imagery', arXiv preprint arXiv:1810. 12813, 2019.

[14] Li P., Abouelenien M., Mihalcea R., Ding Z., Yang Q., and Zhou Y., (2023). Deception Detection from Linguistic and Physiological Data Streams Using Bimodal Convolutional Neural Networks', arXiv preprint arXiv:2311. 10944, 2023.

[15] Li P., Yang Q., Geng X., Zhou W., Ding Z., and Nian Y., (2024). Exploring Diverse Methods in Visual Question Answering', arXiv preprint arXiv:2404. 13565, 2024.

[16] Li S., Dong X., Ma D., Dang B., Zang H., and Gong Y., (2024). Utilizing the LightGBM algorithm for operator user credit assessment research', Applied and Computational Engineering,. 75, 1, 36–47, Jul. 2024.

[17] Li, K., Zhu, A., Zhou, W., Zhao, P., Song, J., & Liu, J. (2024). Utilizing deep learning to optimize software development processes. arXiv preprint arXiv:2404.13630.

[18] Li, K., Xirui, P., Song, J., Hong, B., & Wang, J. (2024). The application of augmented reality (ar) in remote work and education. arXiv preprint arXiv:2404.10579.

[19] Liu, X., Qiu, H., Li, M., Yu, Z., Yang, Y., & Yan, Y. (2024). Application of Multimodal Fusion Deep Learning Model in Disease Recognition. arXiv preprint arXiv:2406.18546.

[20] Liu, X., & Wang, Z. (2024). Deep Learning in Medical Image Classification from MRI-based Brain Tumor Images. arXiv preprint arXiv:2408.00636.

[21] Maurya, S K, Xin L, and Tsuyoshi M. (2022). Simplifying approach to node classification in graph neural networks. *Journal of Computational Science* 62 (2022): 101695.

[22] Oono, K, and Taiji S. (2019). Graph neural networks exponentially lose expressive power for node classification. arXiv preprint arXiv:1905.10947 (2019).

[23] Scarselli, F, et al. (2008). The graph neural network model. IEEE transactions on neural networks 20.1 (2008): 61-80.

[24] Sun, Y, et al. (2011). Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. Proceedings of the VLDB Endowment 4.11 (2011): 992-1003.

[25] Saccone, C., et al. (1995). Molecular classification of living organisms. *Journal of molecular evolution* 40 (1995): 273-279.

[26] Tao, Y. (2017). SQBA: sequential query-based attack. In Fifth International Conference on Artificial Intelligence and Computer Science (AICS 2023) (pp. 128032Q). SPIE.

[27] Tao, Y. (2023). Meta Learning Enabled Adversarial Defense. In 2023 IEEE International Conference on Sensors, Electronics and Computer Engineering (ICSECE) (pp. 1326-1330).

[28] Tao, Y., Jia, Y., Wang, N., & Wang, H. (2019). The FacT: Taming Latent Factor Models for Explainability with Factorization Trees. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 295–304). Association for Computing Machinery.

[29] Tan, L., Liu, S., Gao, J., Liu, X., Chu, L., & Jiang, H. (2024). Enhanced Self-Checkout System for Retail Based on Improved YOLOv10. arXiv preprint arXiv:2407.21308.

[30] Veličković, P, et al. (2017) Graph attention networks. arXiv preprint arXiv:1710.10903 (2017).

[31] Xiao, S, et al. (2022) Graph neural networks in node classification: survey and evaluation. Machine Vision and Applications 33.1 (2022): 4.

[32] Yang, Y., Qiu, H., Gong, Y., Liu, X., Lin, Y., & Li, M. (2024). Application of Computer Deep Learning Model in Diagnosis of Pulmonary Nodules. arXiv preprint arXiv:2406.13205.

[33] Zhang, C, et al. (2019). Heterogeneous graph neural network. Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2019.

[34] Zhao, P., Li, K., Hong, B., Zhu, A., Liu, J., & Dai, S. (2024). Task allocation planning based on hierarchical task network for national economic mobilization. *Journal of Artificial Intelligence General science (JAIGS)* ISSN: 3006-4023, 5(1), 22–31.

[35] Zhu, M., Zhang, Y., Gong, Y., Xing, K., Yan, X., & Song, J. (2024). Ensemble methodology: Innovations in credit default prediction using lightgbm, xgboost, and localensemble. *arXiv preprint arXiv:2402.17979*.

[36] Zhang, Y., Gong, Y., Cui, D., Li, X., & Shen, X. (2024). Deepgi: An automated approach for gastrointestinal tract segmentation in mri scans. *arXiv preprint arXiv:2401.15354*.