Residual Attention Augmentation Graph Neural Network for Improved Node Classification

Muhammad Affan Abbas
Department of Electrical and Information Engineering, Control Science and Engineering, Tianjin University, China
affan_abbas8@yahoo.com

Waqar Ali
Department of Environmental Sciences, Informatics, and Statistics, Ca' Foscari University of Venice, Italy
waqar.ali@unive.it

Florentin Smarandache
Mathematics, Physics, and Natural Science Division, University of New Mexico, USA
smarand@unm.edu

Sultan S. Alshamrani
Department of Information Technology, College of Computer and Information Technology, Taif University, Saudi Arabia
susamash@tu.edu.sa

Muhammad Ahsan Raza
Department of Information Sciences, University of Education Lahore, Multan Campus, Pakistan
ahsan.raza@ue.edu.pk

Abdullah Alshehri
Department of Information Technology, Faculty of Computing and Information, Al-Baha University, Saudi Arabia
aashehri@bu.edu.sa

Mubashir Ali
Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, China
mubashir@siat.ca.cn (corresponding author)

Received: 31 December 2023 | Revised: 23 January 2024 | Accepted: 24 January 2024

ABSTRACT

Graph Neural Networks (GNNs) have emerged as a powerful tool for node representation learning within graph structures. However, designing a robust GNN architecture for node classification remains a challenge. This study introduces an efficient and straightforward Residual Attention Augmentation GNN (RAA-GNN) model, which incorporates an attention mechanism with skip connections to discerningly weigh node features and overcome the over-smoothing problem of GNNs. Additionally, a novel MixUp data augmentation method was developed to improve model training. The proposed approach was rigorously evaluated on various node classification benchmarks, encompassing both social and citation networks. The proposed method outperformed state-of-the-art techniques by achieving up to 1% accuracy improvement. Furthermore, when applied to the novel Twitch social network dataset, the proposed model yielded remarkably promising results. These findings provide valuable insights for researchers and practitioners working with graph-structured data.

Keywords-graph neural networks; node classification; over-smoothing; citation and social networks; mixup data augmentation

I. INTRODUCTION

Graphs are used in many different disciplines, such as social networks, biological systems, and recommendation engines, to depict intricate relationships and structures. To properly interpret and utilize graph data, it is critical to learn meaningful node representations within these complex network topologies. In light of this, Graph Neural Networks (GNNs) have become a powerful paradigm that presents a hopeful resolution to this problem [1]. GNNs facilitate efficient node classification, graph classification, and link prediction, among other tasks, by encoding both the local and global graph structure [2]. However, creating effective GNN architectures that meet the unique requirements of node classification is still a challenging issue. Numerous GNN variations have been proposed, each with a unique architectural design and components, creating a vast array of alternatives [3]. The ongoing research for the most effective and efficient GNN architectures that can function effectively on a range of real-world graph data is highlighted by this diversity.

This study aims to offer a thorough and workable approach for enhancing GNN performance in the context of node classification in light of these difficulties. The proposed method aims to improve the capabilities of GNN designs while making them simpler, drawing inspiration from recent developments in the field. This approach combines graph convolutional layers with fully connected layers in a simplified architectural layout. It uses Attention Mechanism (AM) [4] and Data Augmentation (DA) strategies, namely MixUp [5], to further enhance the performance of GNNs in node classification. Strategically incorporated into the GNN architecture, these strategies help improve the generalization of the GNN model. Skip Connections (SCs) are used to reduce the accuracy loss caused by over-smoothing. Extensive tests were performed on various node classification tasks to thoroughly evaluate the performance of the proposed strategies on known benchmarks, such as social networks [6] and citation networks [7]. Concisely, this study:

- Developed a Residual Attention Augmentation Graph Neural Network (RAA-GNN) to enhance the evaluation of the node classification task.
- Developed a novel DA method, called MixUp DA, which combines labels and node attributes to produce synthetic data points and improve the model's ability to classify nodes. Additionally, well-designed skip connections and an effective multi-head attention technique were introduced to improve information aggregation and over-smoothing issues, which together improve GNN performance for node classification.
- Evaluated the proposed method on the Twitch social network dataset, and the results showed up to a 1% gain in accuracy, providing further insights for graph-structured data applications.

II. RELATED WORKS

Several studies have investigated SCs, AMs, and DA in the context of graph-based machine learning. Although DA has been beneficial in enhancing model performance in several fields, its implementation in graph-based machine learning has encountered difficulties. Conventional augmentation methods for graph data, including noise addition or perturbing node properties [8], frequently fail because they break the natural graph structure [9]. Furthermore, the addition of synthetic noise can impede the learning and generalization of the model. The proposed MixUp DA strategy [10] provides a logical method to enhance graph data by seamlessly combining node attributes and labels.

Attention mechanisms have revolutionized information aggregation in GNNs by allowing nodes to choose to attend to the relevant neighbors [11]. However, problems with scalability and processing complexity may make them less successful. Current methods are frequently computationally intensive, and therefore, they are unfeasible for large-scale graphs. Currently, the SuperHyperGraph presents the most general form of graph [11]. These issues are addressed and make it easier to apply attention methods to larger graphs by introducing a multi-head AM that strikes a compromise between expressive capacity and computational efficiency. SCs are important in deep learning architectures because they facilitate the transfer of information between layers [12]. Applying SCs in GNNs has proven difficult, despite their usefulness. Their poor integration can cause over-smoothing, reducing classification accuracy by making nodes indistinguishable through excessive information exchange. This study introduces SCs into the GNN design to mitigate the effects of over-smoothing [13], resulting in improved performance without sacrificing accuracy. Consequently, SCs, AM, and DA [14-15] have all been crucial in the advancement of graph-based machine learning. This study addresses these systems' drawbacks by providing a computationally efficient multi-head AM, a more principled approach to DA, and a method for preventing over-smoothing using SCs. Together, these developments enhance node classification in GNNs and enable a greater variety of complicated, real-world graph data to be used in GNNs.

III. METHODOLOGY

Figure 1 shows the architecture of the proposed RAA-GNN model. In the first step, the MixUp augmentation strategy employs a feature-label augmentation method to increase the robustness of the training dataset. Then, the AM is used, which permits the adaptive weighting of pertinent neighbors, boosting the model's capacity to identify significant local structures and raising classification accuracy all around. Following this, SCs are used to solve the over-smoothing problem of GNNs for node classification.

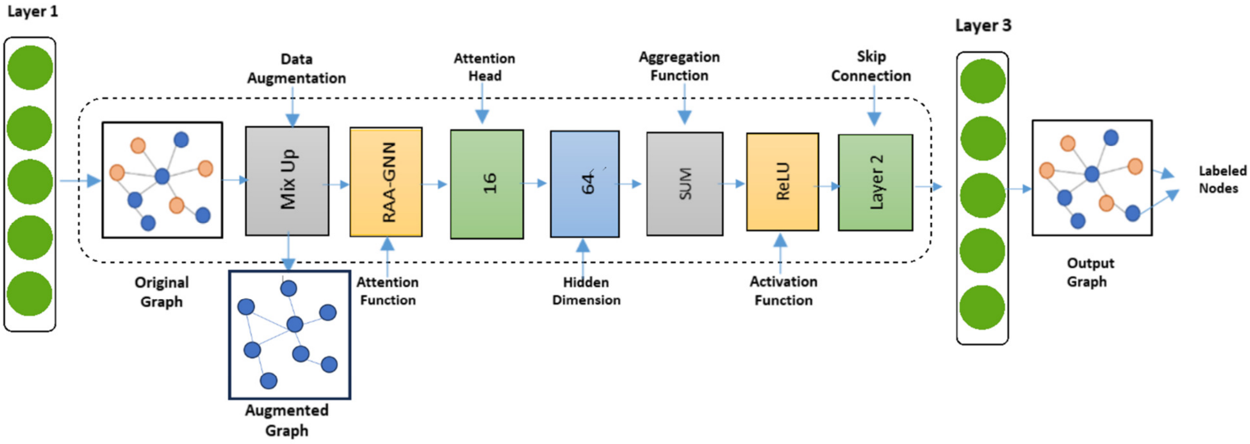

Fig. 1. The RAA-GNN architecture integrates SCs, AMs, and DA for improved node classification.

A. Node Augmentation MixUp Method

GNNs can be designed with MixUp augmentation as a practical and efficient approach to improve node classification performance. The proposed method is based on meticulous preprocessing of input data represented by *X*, which guarantees consistency and standardization. MixUp augmentation serves as a dynamic catalyst by carefully combining the training dataset to add robustness and diversity. The amount of augmentation is dynamically influenced by the mixing parameter *λ*. This coefficient highlights the controlled variability included during training and is randomly generated from a beta distribution with parameters:

$$X_{mix} = \lambda X + \ (1 - \ \lambda)_{Xshuffled} \qquad (1)$$

MixUp works in unison with the larger GNN design, where SCs and attention processes are essential building blocks to improve the model's comprehension of complex graph structures. The following mathematical formulas capture MixUp's effect on the data and explain how original and shuffled features and labels are well combined, which adds to the model's flexibility:

$$Mixed\ Data =$$

$$\lambda \cdot Batch\ Data + \ (1 - \ \lambda) \cdot Batch\ Data[Indices] \qquad (2)$$

$$Mixed\ Labels =$$

$$\lambda \cdot Batch\ Labels + (1 - \ \lambda) \cdot Batch\ Labels[Indices] \quad (3)$$

Integrating MixUp into the GNN model promotes a more robust and flexible learning process for node categorization tasks. The proposed GNN architecture is at the forefront of node classification research because of this deliberate augmentation, which also strengthens the model's ability to generalize across a variety of graph configurations and enrich the training dataset.

B. Attention Mechanism (AM) for Node Classification

RAA-GNN is used to represent the attention function, which is essential to the model's ability to concentrate on the most pertinent data inside the graph structure. With sixteen attention heads, the model captures many structural details and complex relationships, leading to a thorough comprehension. The ReLU activation function is used to combine the contributions of each attention head, represented as RAA-GNNi(*X*, *A*), to get the final attention scores. Each attention head makes a unique contribution to the overall AM. The adjacency matrix is represented by *A*. Intricate graph patterns are captured by 64 hidden dimensions, which balance model expressiveness and efficiency. The aggregation function combines data from nearby nodes by applying the summation:

$$H_{agg} = \sum_N H_N \qquad (4)$$

where *N* is for the neighbor. By introducing non-linearity, ReLU activation improves the model's capacity to learn intricate relationships:

$$H_{activated} = \ max\left(0, H_{agg}\right) \qquad (5)$$

C. Skip Connections (SCs)

The SCs, denoted by H_{skip}, enable the smooth transfer of data between the model's layers. These SCs serve to bridge the gap between subsequent layers by integrating the activated features ($H_{activated}$) with the features of the preceding layer ($H_{previous}$), therefore facilitating the transfer and retention of crucial information. By ensuring that important information from previous layers is merged, this additive process improves the model's ability to represent both local and global interdependence.

$$H_{skip} = H_a + H_p \qquad (6)$$

where *a* is the activated and *p* is the previous. The model's three layers improve node classification by capturing hierarchical representations:

$$H_{output} = GNN(H_{skip}^{(3)}) \qquad (7)$$

The output graph, which displays node classifications based on learned features, is generated by the last layer:

$$Y_{pred} = Softmax\left(H_{output}\right) \qquad (8)$$

This architecture provides a basis for strong node classification in a variety of datasets by utilizing cutting-edge methods to address graph-based learning difficulties.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

Extensive simulations were carried out to evaluate the performance of the proposed GNN with the novel features of DA, AM, and SCs. The findings demonstrate the complex relationships between these elements and the accuracy of node classification on a variety of datasets. The results in Table I shed important light on the relative importance of the various parts of the proposed GNN model. These features are crucial for accurately capturing complex relationships in graphs, as demonstrated by the model's strong performance across a range of datasets. The results advance the knowledge of GNN architectures and provide useful advice for creating powerful models that are suited to particular uses.

The experimental results demonstrate that the proposed GNN architecture changes increase the model's test set accuracy up to 1%. Adam optimizer, with a learning rate of 0.01, DA, AM, SC, and three GCN layers with the ReLU activation function were utilized in the top-performing design. Table II shows the reported mean classification accuracy for the fully supervised node classification task for various graph neural network models. The bold numbers represent the best results while the second bests are underlined. The results for GCN, Mix-Hop, and GraphSAGE were obtained from [16]. The results For GCNII, NodeAug, FSGNN, GPRGNN, and GEOM-GCN were taken from [17, 18]. Traditionally, GNN models such as GCN and GAT have more efficiency on homophily datasets, although they give poor results on datasets with heterophily. Advanced models such as WRGAT, and GPRGNN function are reasonably superior on datasets with both homophily and heterophily. The proposed model performs significantly better on heterophily datasets, particularly with a notable boost on the CiteSeer and Chameleon datasets. Improvements were also noted for the datasets from Actor, Texas, and Cornell. The proposed model achieves consistent and comparable performance to state-of-the-art methods on homophily datasets. It also performed exceptionally well in the evaluation of the new Twitch social network dataset for node classification, demonstrating its flexibility to various graph architectures [19].

TABLE I. NODE CLASSIFICATION ACCURACY (%) FOR THE PROPOSED MODEL

	Cora	**CiteSeer**	**PubMed**	**Chameleon**	**Wisconsin**	**Texas**	**Cornell**	**Squirrel**	**Actor**
Proposed	88.94	78.32	89.14	79.31	86.14	86.21	86.57	72.22	36.71
Without DA	85.33	73.52	87.37	78.30	84.90	85.12	86.21	71.39	34.99
Without AM	86.22	75.59	87.13	79.01	84.90	84.22	84.43	71.35	33.78
Without SC	85.59	77.23	86.97	78.30	83.09	84.32	85.55	71.87	32.34

TABLE II. NODE CLASSIFICATION ACCURACY (%) FOR DIFFERENT MODELS ON VARIOUS DATASETS

Model	**Cora**	**CiteSeer**	**PubMed**	**Chameleon**	**Wisconsin**	**Texas**	**Cornell**	**Squirrel**	**Actor**	**Mean Acc.**
GCN	87.28	76.68	87.38	59.82	59.80	59.46	57.03	36.89	30.26	61.62
GraphSAGE	86.90	76.04	88.45	58.73	81.18	82.43	75.95	41.61	34.23	69.50
MixHop	87.61	76.26	85.31	60.50	75.88	77.84	73.51	43.80	32.22	68.10
GEOM-GCN	85.27	77.99	90.05	60.90	64.12	67.57	60.81	38.14	31.63	64.05
GCNII	88.01	77.13	**90.30**	62.48	81.57	77.84	76.49	N/A	N/A	–
NodeAug	86.20	75.40	82.1	N/A	N/A	N/A	N/A	–	–	–
GPRGNN	88.49	77.08	88.99	66.47	85.88	**86.49**	81.89	49.03	36.04	73.37
FSGNN	87.61	77.17	89.70	78.93	**87.25**	85.90	86.23	**73.32**	34.89	77.88
RAA – GNN	**88.94**	**78.32**	89.14	**79.31**	86.14	86.21	**86.57**	72.22	**36.71**	**78.17**

TABLE III. RESULTS ON A NEW TWITCH SOCIAL NETWORKS DATASET

Dataset	**Train Loss**	**Train Accuracy**	**Validation Loss**	**Test Accuracy**
Twitch Social Networks	0.2463	0.8989	0.9186	0.9046

V. CONCLUSION

This study presents the RAA-GNN model for node classification that incorporates SC, AM, and DA, showing that these elements can work together to improve its discriminative ability. While SCs handle over-smoothing issues, the AMs specifically allow the model to perform better on graphs for node classification. DA is an essential component that adds variation to the training dataset and promotes robustness against overfitting. The experimental study highlighted each component's independent effectiveness, as well as their combined impact on overall performance. The proposed model demonstrated its adaptability by consistently outperforming state-of-the-art approaches in node classification across multiple datasets. In summary, this study extends GNN architectures and sheds light on the complex interactions between SC, DA, and AM. It also sets a new Sota node classification in graph structure learning through the first attempt to integrate SC, AM, and DA into RAA-GNN, thus advancing our understanding of GNNS.

ACKNOWLEDGMENT

The authors would like to acknowledge the Deanship of Scientific Research, Taif University for helping with this work.

REFERENCES

[1] S. Khoshraftar and A. An, "A Survey on Graph Representation Learning Methods," *ACM Transactions on Intelligent Systems and Technology*, vol. 15, no. 1, Jan. 2024, Art. no. 19, https://doi.org/10.1145/3633518.

[2] P. W. Battaglia *et al.*, "Relational inductive biases, deep learning, and graph networks." arXiv, Oct. 17, 2018, https://doi.org/10.48550/arXiv.1806.01261.

[3] J. Zhou *et al.*, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, Jan. 2020, https://doi.org/10.1016/j.aiopen.2021.01.001.

[4] A. Elhassouny and F. Smarandache, "Trends in deep convolutional neural Networks architectures: a review," in *2019 International Conference of Computer Science and Renewable Energies (ICCSRE)*, Agadir, Morocco, Jul. 2019, https://doi.org/10.1109/ICCSRE.2019.8807741.

[5] H. Guo, Y. Mao, and R. Zhang, "MixUp as Locally Linear Out-of-Manifold Regularization," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 3714–3722, Jul. 2019, https://doi.org/10.1609/aaai.v33i01.33013714.

[6] P. H. C. Avelar, A. R. Tavares, M. Gori, and L. C. Lamb, "Discrete and Continuous Deep Residual Learning Over Graphs." arXiv, Nov. 26, 2019, https://doi.org/10.48550/arXiv.1911.09554.

[7] R. Liao, Z. Zhao, R. Urtasun, and R. S. Zemel, "LanczosNet: Multi-Scale Deep Graph Convolutional Networks." arXiv, Oct. 23, 2019, https://doi.org/10.48550/arXiv.1901.01484.

[8] A. Deptuła, "Application of the Dependency Graph Method in the Analysis of Automatic Transmission Gearboxes," *Engineering, Technology & Applied Science Research*, vol. 11, no. 2, pp. 7033–7040, Apr. 2021, https://doi.org/10.48084/etasr.4098.

[9] K. Ding, Z. Xu, H. Tong, and H. Liu, "Data Augmentation for Deep Graph Learning: A Survey," *ACM SIGKDD Explorations Newsletter*, vol. 24, no. 2, pp. 61–77, Sep. 2022, https://doi.org/10.1145/3575637.3575646.

[10] X. Han, Z. Jiang, N. Liu, and X. Hu, "G-Mixup: Graph Data Augmentation for Graph Classification," in *Proceedings of the 39th International Conference on Machine Learning*, Jun. 2022, pp. 8230–8248.

[11] F. Smarandache, "Extension of HyperGraph to n-SuperHyperGraph and to Plithogenic n-SuperHyperGraph, and Extension of HyperAlgebra to n-ary (Classical-/Neutro-/Anti-)HyperAlgebra," *Neutrosophic Sets and Systems*, vol. 33, pp. 289–295, Feb. 2020, https://doi.org/10.5281/zenodo.3783103.

[12] K. Xu, M. Zhang, S. Jegelka, and K. Kawaguchi, "Optimization of Graph Neural Networks: Implicit Acceleration by Skip Connections and More Depth," in *Proceedings of the 38th International Conference on Machine Learning*, Jul. 2021, pp. 11592–11602.

[13] T. K. Rusch, M. M. Bronstein, and S. Mishra, "A Survey on Oversmoothing in Graph Neural Networks." arXiv, Mar. 20, 2023, https://doi.org/10.48550/arXiv.2303.10993.

[14] D. D. Van, "Application of Advanced Deep Convolutional Neural Networks for the Recognition of Road Surface Anomalies," *Engineering, Technology & Applied Science Research*, vol. 13, no. 3, pp. 10765–10768, Jun. 2023, https://doi.org/10.48084/etasr.5890.

[15] H. Sasaki, S. Yamamoto, A. Agchbayar, and N. Nkhbayasgalan, "Extracting Problem Linkages to Improve Knowledge Exchange between Science and Technology Domains using an Attention-based Language Model," *Engineering, Technology & Applied Science Research*, vol. 10, no. 4, pp. 5903–5913, Aug. 2020, https://doi.org/10.48084/etasr.3598.

[16] J. Zhu, Y. Yan, L. Zhao, M. Heimann, L. Akoglu, and D. Koutra, "Beyond Homophily in Graph Neural Networks: Current Limitations and Effective Designs," in *Advances in Neural Information Processing Systems*, 2020, vol. 33, pp. 7793–7804.

[17] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li, "Simple and Deep Graph Convolutional Networks," in *Proceedings of the 37th International Conference on Machine Learning*, Nov. 2020, pp. 1725–1735.

[18] S. K. Maurya, X. Liu, and T. Murata, "Simplifying approach to node classification in Graph Neural Networks," *Journal of Computational Science*, vol. 62, Jul. 2022, Art. no. 101695, https://doi.org/10.1016/j.jocs.2022.101695.

[19] B. Rozemberczki and R. Sarkar, "Twitch Gamers: a Dataset for Evaluating Proximity Preserving and Structural Role-based Node Embeddings." arXiv, Feb. 16, 2021, https://doi.org/10.48550/arXiv.2101.03091.