Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/mbe

MBE, 19(12): 12146–12159.
DOI: 10.3934/mbe.2022565
Received: 16 May 2022
Revised: 05 August 2022
Accepted: 11 August 2022
Published: 19 August 2022

Research article

Classification of vertices on social networks by multiple approaches

Hacı İsmail Aslan[1], **Hoon Ko**[2,*] **and Chang Choi**[1,*]

[1] Department of Computer Engineering, Gachon University, Seongnam-daero, Sujeong-gu, Seongnam-si, Gyeonggi-do, Republic of Korea

[2] Research Institute of Computer, Information, Communication, Chungbuk National University, 8-7, Chungdae-ro 1, Seowon-Gu, Cheongju-si, 28644, Chungcheongbuk-do, Republic of Korea

* **Correspondence:** Email: skoh21@chungbuk.ac.kr, changchoi@gachon.ac.kr; Tel: +82432613140, +82317505325; Fax: +82432633140.

Abstract: Due to the advent of the expressions of data other than tabular formats, the topological compositions which make samples interrelated came into prominence. Analogically, those networks can be interpreted as social connections, dataflow maps, citation influence graphs, protein bindings, etc. However, in the case of social networks, it is highly crucial to evaluate the labels of discrete communities. The reason for such a study is the importance of analyzing graph networks to partition the vertices by only using the topological features of network graphs. For each interaction-based entity, a social graph, a mailing dataset, and two citation sets are selected as the testbench repositories. The research mainly focused on evaluating the significance of three artificial intelligence approaches on four different datasets consisting of vertices and edges. Overall, one of these methods so-called "harmonic functions", resulted in the best form to classify those constituents of graph-shaped datasets. This research not only accessed the most valuable method but also determined how graph neural networks work and the need to improve against non-neural network approaches which are faster and computationally cost-effective. Also in this paper, we will show that there is a limit to be accessed by prospective graph neural network variations by using the topological features of trialed networks.

Keywords: graph neural networks; graph attention networks; harmonic functions; node classification; social network; semi-supervised learning

1. Introduction

In the current era of social networking, knowledge exchange in any form of information exhibits liaisons between individuals. These social networks may include physical contacts, messages sent, collaborative studies between peers, sentimentally closeness, etc. While the information flow is con-

structing a network, the format appears as a knowledge map [1] which implies a graph structure. Hence, analysis of graph formatted data is essential to obtain more details regarding social networks.

Typically, the data representations tend to be in a tabular or non-Euclidean format to make the data analyzed in a structured way [2]. Those data which were expressed in terms of samples and attributions related to every single sample can be imagined as rows and columns intersecting. The former can not imply any relatedness between samples, since tabular-formatted data structure will not provide such an ability. However, linkages in the real world are significant enough to be taken into account while working with samples that have interdependencies in aspects of many topics possibly. Thus, graph networks are the key players here to make significance by their abilities to show related samples with links in between. In addition to mentioning graph networks, the mathematical foundations of graph networks will be explained in the later sections briefly.

As a variant of the neural network family which has the ability to handle non-Euclidian data, a Graph Neural Network [3], called GNN for the sake of simplicity, is a brand new method in use. Namely said GNNs can take grid-wise graph inputs which show the intercorrelations between samples and evaluate many tasks which will be mentioned later. Having said that, GNNs are getting more popular in aspects of their usage. While its interpretations lead to new approaches, many models are derived depending on GNNs. Its most common fields of achievement are molecular biology [4–6], network sociology [7–9], knowledge graphs [10, 11], road traffic [12], natural language processing [13, 14], and even computer vision [15, 16]. Adding that, recent development in varied versions of GNNs such as GCN [17] which brought convolutional neural networks to the graph domain, GraphSAGE [18] which utilized aggregation functions during the message passing, APPNP [19] which derived a method from message passing to personalized PageRank [20] algorithm by fusing the famous PageRank with GNNs, SGC [21] which reduced the excess complexity by removing nonlinearities and simplified GCNs, GAT [22] which derives inspiration from attention mechanisms as in deep learning approaches, and DGI [23] which uses unsupervised methods to learn graph representations led this area of neural networks to grow and disseminate, chronologically. Two of these, GCN and GAT, need to be paid attention to in terms of their exemplarity for the other methods since one leads the convolutional approaches whilst the other leads the attention-based mechanisms. Other than variations of GNNs, there should be mentioned other learning methods such as random walks [24], spectral graph theory applications [25], the nearest neighbor approach, and harmonic functions [26].

The problematic features of classifying vertices over a graph by variations of GNNs, whereas non-neural network models are overlooked, influenced this paper. To solve this, GCN and GAT models have been benchmarked against a semi-supervised learning method called harmonic functions. Hereby this article, GCN, GAT, and harmonic functions have been used and the accuracies have been investigated under the same circumstances for each method. Resultingly, this research contributes to understanding the limit to improve graph neural networks and the effectiveness of topological structures while the classification of nodes is aimed.

Our contributions are as follows: (i) We show that topological connections are useful even without node features by using three ML methods for the node classification task while we utilized only topological structures of graph-represented data. (ii) We propose that there is a certain limit of accuracy score to be accessed by prospective modulations of GNNs to be accepted as progressive methods. (iii) We prove the effectiveness of harmonic functions against GCN and GAT when it comes to homogeneous graphs without node and edge features.

2. Materials and methods

2.1. Network datasets and related insights

The selection of adequate sets to be subject to processing by particular algorithms is substant for evaluating the throughput of the research. Therefore, the first trials for entire algorithms were performed on a dataset called "Zachary's Karate Club Dataset", created by Zachary et al. in 1977 [27]. The ground basis for such a selection relies on the simplicity of the dataset mentioned in the previous line. Zachary's Karate Club Dataset, which will be abbreviated as "Karate Dataset" in this article consists of 34 individuals from a local karate club. The dataset shows the members via vertices and the social interactions between club members using edges, basically. Later in time, the group was split into two subsections because of an internal argument between officials and the task here is to predict every student's final decision on whom to get the course from. As depicted in Figure 1, class tutors are portrayed by nodes number 0 and 33.

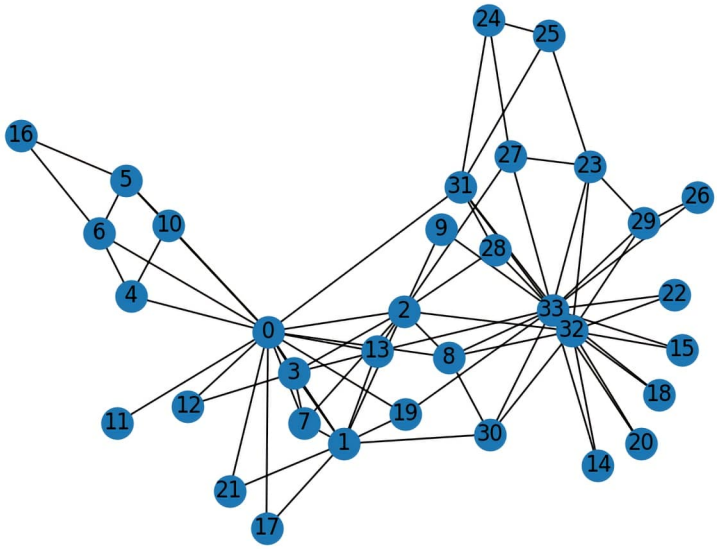

Figure 1. Visualization of Zachary's karate club dataset.

Another data that was extracted from a large European research institution was adopted to be used as a testbench dataset by using a network repository SNAP [28]. According to the analysis done by Bharali et al. [29], the "Email-Eu-Core network" is found to be a Small-World network that exhibits power law-regime with an assortative mixing pattern on the degree of nodes. In the same research paper, the network was said to be robust against random failures but vulnerable to targeted attacks. Besides, the average degree which means the number of edges compared to the number of nodes was relatively huge considering the same criterion for other datasets introduced hereby in this paper. Thus, the graph-network abbreviated as "Email Dataset" which consists of an internal mailing network had some differences to be under investigation for node classification. The challenging structure of the dataset led it to be taken as one of these benchmark datasets in this study.

Lastly in this section, two datasets called "Cora" and "Pubmed" datasets should be mentioned since these are covered in the introduced research. Both sets are consisting of scientific writings and the citation linkages of these articles. As might be expected, papers refer to nodes whereas citations are presented as edges for both citation networks. To fetch these datasets in the form of adjacency matrices, a particular python library, Spektral library [30], has been used.

Those citation networks which are commonly used in the graph theory area were quite important to generalize the outputs of classification results discussed in the subsequent "Results" part. Here, there was a hardship in the case of the Pubmed dataset due to its average clustering coefficient. The average clustering coefficient (CC) is a phenomenon that refers to the possibility to draw a triangle by a node and its two distinct neighbors. CC can also be interpreted as the fraction of the number of closed triplets by the number of all triplets in the network graph.

In an outlined form, one can refer to Table 1 as a summary of the datasets utilized during the experiments.

Table 1. Summary of the datasets.

	Karate	Email	Cora	Pubmed
No. Nodes	34	1005	2708	19,717
No. Edges	78	25,571	5429	44,338
No. Clusters	2	42	7	3
No. Training Nodes	2	804	2166	5000
Average Clustering Coefficient	0.57	0.4	0.24	0.06
Average Degree	4.59	25.44	3.9	4.5

2.2. Preprocessing node and edge embeddings

Most generally, when one's working on graph data without node and edge attributions, the preprocessing schedule should be short. For instance, in the case of the karate dataset, there was no action needed to process the inputs since those were already in an abstract form. Similarly, the email dataset didn't demand anything besides turning the data list to tuples to be handled.

However, for the Cora dataset, there were some necessary transformations to be considered. The main reason why it matters to rebuild the Cora dataset is that the harmonic function takes the node inputs as integers and those integers should be in an adjacent format. Additionally, GAT and GCN take the label arguments as integers as well. Contrary to the requirements, the Cora dataset accommodates vertices in a non-consecutive order. Having said that, particular numbers of nodes are not varying in between the range of numbers of samples which means node numbers should be arranged in order to have those nodes adjacently increasing from zero to the number of total vertices in the set. As $V_i \rightarrow V_i^{'}$ and $V_j \rightarrow V_j^{'}$ (where V_i and V_j refers to the vertices having an interconnection) the edge connection should remain the same as $Edge(V_i, V_j) \rightarrow Edge(V_i^{'}, V_j^{'})$ while $Edge$ showing the link between two vertices.

The aforementioned situation of labels of the Cora dataset has been solved by mapping the clique texts to specific integers. One can refer to Table 2 for the encoding details per each label name.

Table 2. Original label names and related integers while preprocessing.

Original Label	Encoded Target
Rule Learning	0
Neural Networks	1
Theory	2
Case Based	3
Probabilistic Methods	4
Genetic Algorithms	5
Reinforcement Learning	6

2.3. Graph Convolutional Networks and Graph Attention Networks

To fulfill the aim of this study, two networks were applied to the datasets and described in the following subsections. It should be kept in mind that many other models could be used to achieve classification tasks. However, in this paper, when comparing harmonic functions with GCN and GAT, it is aimed to set a higher limit in terms of accuracy than the lower limit that harmonic functions will define for future neural network models to achieve.

2.3.1. Graph Convolutional Networks

Proposed by Kipf et al., Graph Convolutional Networks [31] can be seen as the enhancements to neural networks that operate on graphs which had been previously introduced by Gori et al. [32]. For convenience, it would be nifty to show the mathematical foundations of GCNs so that the surrounding dialectics can be understood well.

Prescribed as inputs $\boldsymbol{A}$, the adjacency matrix in the shape of $\boldsymbol{N} \times \boldsymbol{N}$, and $\boldsymbol{H}$, the feature matrix in the shape of $\boldsymbol{N} \times \boldsymbol{F}$, where each vertex has 2-D feature vector in the shape of $\boldsymbol{1} \times \boldsymbol{F}$, the basic propogation rule for a GCN is given by Eq (2.1).

$$H' = \sigma(AHW) \tag{2.1}$$

Noting that W is a the weight matrix in the shape of $\boldsymbol{F} \times \boldsymbol{F'}$, $\boldsymbol{H'}$ stands for the newly generated node features, and $\boldsymbol{\sigma(\cdot)}$ implements the non-linearity function, the node i was focused to get the following equation to simply sump up the transformed features of all connecting nodes. Equation (2.2) depicts the correlation in between a node's features h_i and the features of the connected nodes h_j to the particular node i. Please note that $\boldsymbol{V(i)}$ is the group of node i's neighboring nodes.

$$h_i = \sigma\left(\sum_{j \in V(i)} W^T h_j\right) \tag{2.2}$$

Conflicting with the expression made, the previous propagation rule which is depending on the sum-pool doesn't perform solid. The reason which leads to such an outcome is how the propagation rule introduced in Eq (2.2) just sums up feature vectors. This may lead the repeated applications to increase the scale of the features. To overcome that obstacle, another update rule that can be simulated

by Eq (2.3) satisfies the normalization of the adjacency matrix by multiplying it by the inverse of the diagonal degree matrix $\boldsymbol{D}$. Consequently, a newly updated node feature vector is shown by Eq (2.4).

$$H' = \sigma(D^{-1}AHW) \tag{2.3}$$

$$h_i^{'} = \sigma\left(\sum_{j\in V(i)} \frac{1}{|V(i)|} W^T h_j\right) \tag{2.4}$$

The previous propagation rules were presented in order to explain how GCN works. Now, the rule which is derived from the previous equations, so-called symmetric normalization should be visited as Eq (2.5) shows, to see how it multiplies the adjacency matrix by the square root of the inverse of the diagonal degree matrix. Resultingly, the node features are up to be changed according to Eq (2.6)

$$H' = \sigma(D^{-1/2}AD^{-1/2}HW)) \tag{2.5}$$

$$h_i^{'} = \sigma\left(\sum_{j\in V(i)} \frac{1}{\sqrt{|V(i)|}\sqrt{|V(j)|}} W^T h_j\right) \tag{2.6}$$

The implementation of GCN in the case of this research has exploited the theoretical foundations expressed in this section. Even so, this study aimed to work with topological bonds of the graph-structured data without fetching any information related to nodes. Hence, features for each node were set to zero initially.

2.3.2. Graph Attention Networks

Graph Attention Networks (GATs) were introduced in Veličković et al. as a novel approach that benefits the GCN's background but adds particular "attention" mechanisms. The mathematical pinnings of GAT can be viewed on its inventor's paper [22], though it is still needed to outline the unique annexes of GAT in the context of this paper.

Unlike GCN, the coefficients in GAT are not constant due to the relaxation of the coefficients being dependent on the current input. The idea explained is the motivation of attention mechanisms for such a graph neural network. Having said that we have non-constant coefficients, the attention coefficient α_{ij} while the j is the sender and i is the receiver nodes were computed as Eq (2.7) shows.

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(a[W^T h_i \| W^T h_j]))}{\sum_{k\in V(i)} \exp(\text{LeakyReLU}(a[W^T h_i \| W^T h_k]))} \tag{2.7}$$

Theoretically, a one-layered MLP, which is symbolized by **a**, has been applied on concatenated messages $W^T h_i$ and $W^T h_i$ by the activation function which was propositioned as the LeakyReLU function. Over and above that, GAT exerts multi-head attention which means each GAT layer has a fixed number of independent duplicates. Those outputs obtained through each duplicated layer of GAT, then, were concatenated to produce a finalized feature vector. As Eq (2.8) expresses, the normalized attention coefficients are used to determine the features corresponding to them, and to assign the final output features for every node.

$$h_i^{'} = \sigma\left(\sum_{j\in V(i)} \alpha_{ij} W^T h_j\right) \tag{2.8}$$

2.4. Harmonic functions

As stated by Zhe et al. [33], the semi-supervised learning method using Gaussian fields and harmonic functions is applicable for networks whose topology is known. With the help of the NetworkX library's useful application programming interface [34], the classification method has been applied to the datasets accordingly. One can refer to the Eq (2.9) which explains what harmonic functions are where a function $h : V \to \mathbb{R}$ is called harmonic function so that the graph $G = (V, E)$ is harmonic. For convenience, d_i refers to the degree of the vertex. Sufficient information can be revisited from He et al. [35] to understand the basis of classification by harmonic functions.

$$h(V_i) = \frac{1}{d_i} \sum_{(V_i, V_j) \in V} h(V_j) \tag{2.9}$$

3. Proposed method

As its subcomponents are defined in the previous section, basically, the proposed method depends on accomplishing node classification tasks through both GCN, GAT, and harmonic functions. Since one of the motivations of this article is to show the impact of topological bindings on classification, the very first step of our proposed method to conduct this research is to sift network data from node and edge features to access only topologic attributions. Consequently, there will be a drastic decrease which may lead to more efficient use of memory in terms of the amount of data.

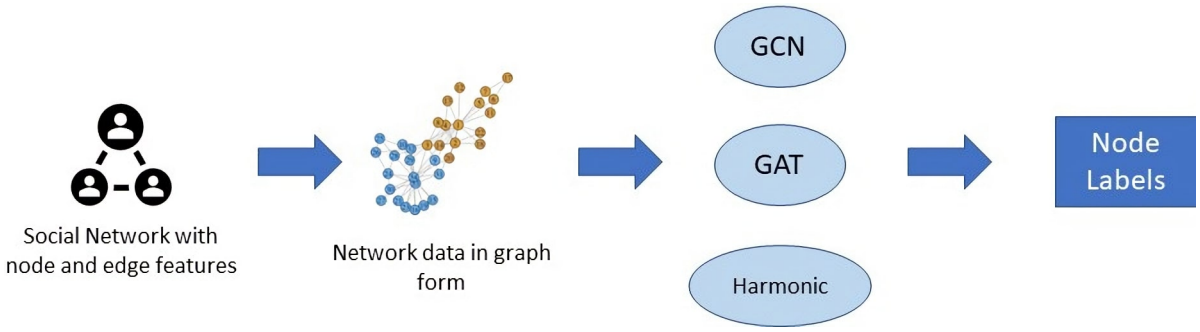

Figure 2. Workflow of the proposed method.

After the extraction of topologic bindings, the aforementioned classification methods were subject to be used separately. Predicted node labels were noted for every classification method to evaluate the importance of the topological structure of graphs for vertex classification and the best model overall for such an application. Figure 2 outlines the projected research method which leads this study. After following those steps depicted in Figure 2, the rest is to interpret the results accordingly. To have a better perception, Algorithm 1 has been extracted to outline the general workflow of the proposed study.

As it can be deducted from the Algorithm 1, the entire process depends on sifting the node features out and evaluating the node classification accuracies over three models. That is a naive approach to perceiving how GCN, GAT, and harmonic functions perform on the node classification task when the

Algorithm 1 Concise algorithm of the proposed method

Require: $G(V, E, F)$
Ensure: $acc1, acc2, acc3$
  function PREPROCESS($G(V, E, F)$):
    $i \leftarrow 0$
    while $V_i \in V$ **do**
      delete F_i
      $i \leftarrow i + 1$
    end while
    return G(V, E)
  end function
  function CLASSIFY($G(V, E)$):
    $acc1 \leftarrow GCN(G(V, E))$
    $acc2 \leftarrow GAT(G(V, E))$
    $acc3 \leftarrow Harmonic(G(V, E))$
    return acc1, acc2, acc3
  end function
  $acc1, acc2, acc3 \leftarrow classify(preprocess(G(V, E, F)))$
  $best\_model \leftarrow max(acc1, acc2, acc3)$

graph is fully homogeneous. One can apply our model from scratch by following the footsteps of the mentioned algorithm.

4. Results

4.1. Environmental Setup

While the learning rate is 0.01, the optimizer is Adam, loss function depends on a negative log-likelihood approach, out of 10 training sessions with 500 epochs in every case, GCN and GAT methods were trialed whereas harmonic function was only trained within 100 iterations because, after a particular number of iterations, the harmonic function doesn't perform any better or worse compared to previous iterations. At the end of those 10 experiments for each dataset and method, the standard deviations were noted since the training data were sampled randomly. For a better generalization, it has been thought that random sampling would give a better idea. Because serving as a validation dataset for the algorithms, the only case in which the training data was not set randomly was the karate dataset case, which took only 2 node assignments as the training data.

In terms of computing architecture, a cloud computing service so-called Google Colab has been utilized in this research because of its simplicity. Python 3 Google Compute Engine backend (TPU) served as the processor, while 35 GB of memory has been subject to be used.

4.2. Research results

Ensuring the construction of methods and datasets, the very explainable sequel occurred for the karate dataset. Hence, the social interaction between the individuals herewith helped the experiment

to distinguish those two distinct cliques. Figure 3 illustrates the classified labels per node by GCN and nodes' corresponding ground-truth assignments.

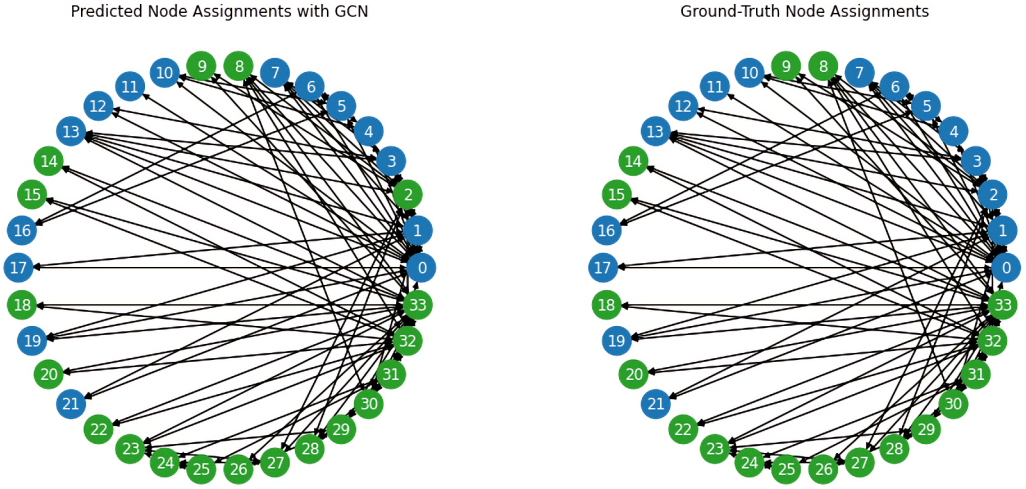

Figure 3. Classification results of intentionally mistuned GCN for karate dataset.

In the case of the karate dataset, it was observed that the classification accuracy could hit the 100% quite easily without the need for complex parameter tuning since the dataset is quite small and partaking in a particular clique is linearly dependent on the social interactions of the club members. Therefore, to highlight the difference in terms of accuracy between classified labels and ground-truth assignments on Figure 3, the GCN was trained during 20 epochs intentionally. In the case of proper training, full accuracy was observed.

In addition to computing the standard deviation, the mean accuracy levels have been measured by simply taking the average of the results in 10 experiments. The train-to-test ratio was 0.8 for every case but Pubmed-GCN and Pubmed-GAT duos. The reason is that the out-of-memory error occurs since the number of processed nodes is too much in the training for the Pubmed dataset. To avoid such an error, the Pubmed dataset has been trained with 5000 vertices which means almost 25% of the entire network.

Interpreting the tables in this section is only possible once one knows the train-test split method that has been used in the current paper. The foremost fact about measuring the test accuracy here depends on the training backbones of semi-supervised learning by maximizing the negative log-likelihood of the known node assignments. Here, the mentioned "known node assignments" can be elucidated as train data, and resultingly, an analogy between unlabelled data and test data can be constructed. Therefore, once the model is fed with a certain number of samples whose ground-truth labels are known, the whole graph is taken as an entire set that includes both test and train data, and the accuracy for the entire set can be recorded by Table 3 and abbreviated as ACC. Yet, the observed accuracy contains both train and test accuracy. To evaluate the test accuracy for unlabelled data, a simple fraction of the train to test is used by assuming the training accuracy 100% as explained in Eq (4.1).

$$\text{Test Accuracy} = \frac{\text{ACC} - (\text{Training Samples Ratio}\%)}{1 - (\text{Training Samples Ratio}\%)} \tag{4.1}$$

According to the latter formula, accuracies of prediction for unlabelled nodes have been shown in Table 3.

Table 3. Comparison of methods used in this research.

Name of the dataset	Accuracy for full dataset (ACC)		
	GCN	GAT	Harmonic
Karate	100%	100%	100%
Email	94.42 ± 0.52%	93.59 ± 0.47%	94.14 ± 0.68%
Cora	96.29 ± 0.3%	96.61 ± 0.29%	97.27 ± 0.23%
Pubmed	85.29 ± 0.26%	85.55 ± 0.24%	96.47 ± 0.07%
	Accuracy for unlabelled data (test accuracy)		
	GCN	GAT	Harmonic
Karate	100%	100%	100%
Email	72.1 ± 0.52%	67.95 ± 0.47%	70.7 ± 0.68%
Cora	81.45 ± 0.3%	83.05 ± 0.29%	86.35 ± 0.23%
Pubmed	80.39 ± 0.26%	80.73 ± 0.24%	82.35 ± 0.07%

5. Discussion

As it predominantly appeared, the harmonic function outputs slightly better than the others for node classification tasks on Cora and Pubmed datasets in terms of mean accuracy levels. Having said that, the standard deviation occurred smaller than it occurred for the other datasets which means harmonic functions are more robust against different training samples. When the number of samples is higher, the standard deviation goes lower as expected theoretically. The same or the opposite can not be said for the mean accuracy levels since they depend on internal knowledge in each graph. The amount of samples is only one fact but not the most crucial. When it comes to graph-structured data, the heterogeneity [36] issue emerges as the key.

Known as message passing algorithms, both GCN and GAT depend on the same theory but differ on update and aggregation rules. Contingent with these methods, varied versions of usage-specific graph neural networks have been deployed in recent years as stated in the Introduction section. Hence, GCN and GAT served as seed points for many others to initiate fresh methods to overcome graph-related tasks in artificial intelligence. Though graph neural networks show impressive performance on graph-related tasks in the machine learning field, they still need to exceed certain limits. For sure, most of the cutting-edge techniques have already gone beyond the limits previously achieved by GCNs and GATs. However, most of the novel approaches in GNNs still depend on GCN and GAT theory. Intuitively, there must be a comparison level for these new models to be accepted as an improvement. In this research, the limit has been proposed as classification by harmonic functions as introduced by Zhu et al. For the popular datasets, such as Cora and Pubmed datasets, this theory seems to work fine. Unlike the motivation of this theory, GCN outperformed harmonic functions for the Email dataset, which showed us this limit may not be applicable for every dataset.

Whereas harmonic functions slightly outperform GCN and GAT, it still lacks the ability to handle feature vectors related to entries. Thus, classification with harmonic functions is only possible with topological correlations of nodes in the graph. This disability of not being able to work with feature

matrix can be offered as a soft spot for harmonic functions which need to be fixed by new modifications. Recalling the fact that both GCN and GAT can work with feature matrices as introduced by Eq (2.1), merging the node update rule of harmonic functions and the feature update rule of GCN or GAT would devise a new and even better method for node classification tasks. Even so, this idea needs confirmation by further research but is still worth noting here in this section. Moreover, due to heterogeneity in graphs, generalization of the limit to be accessed by new approaches as the output of harmonic functions is not possible. However, it can still be set as a limit for particular datasets as has been shown in this paper. Having said that, one should note that only the topological attributes have been used which makes the case out of heterogeneous graphs. For further studies, node features will be included for a larger set of test cases. Moreover, edges with attributes may also be included to enhance the topological information yielded by a graph structure.

6. Conclusions

Hereby this paper, while using three methods on four sample spaces, the topological bindings of individuals in each sample space have been processed to classify nodes into related communities/labels. The contributions, as stated in the introduction section, were achieved to some extent. The main goal was to reach a certain limit of accuracy scores in order to be accessed by prospective modulations of GNNs. Moreover, after sifting the data to obtain its topological structure, we were able to show that featureless graph data was still useful on its own for graph neural networks and harmonic functions. Lastly, as our main goal depicts, harmonic functions appeared more effective against GCN and GAT while working with graphs without node and edge attributions. The novelty of this article is twofold. First, classification by harmonic functions was compared against state-of-the-art models and achieved significant success. To the best of our knowledge, it was not the focus of previous research in this area. Second, we have shown that the graph data is still useful without node features. That also brings an efficient use of memory since features may or may not be important depending on the case. Our paper shows using the graph data without utilizing features must be considered even if this is contrary to intuition. Upon exploring the situation from multiple perspectives, this research may serve as an artificial intelligence backbone for social network-based studies. In particular, Yao et al. [37,38] depict the semantic analysis based on traditional network analysis techniques to understand user behavior in certain clusters. We believe such applications could be revamped by innovations in the graph machine learning field.

7. Future work

The process of preparing the current research article led to many ideas regarding Graph Neural Networks. As it was divided into details, GNNs' handiness to obtain node embeddings like a feature extractor or as an encoder influenced us to investigate if GNN models could be merged with other ML techniques for industrial applications. Especially, the banking transaction datasets that consist of licit and illicit transactions attracted our attention, since these sorts of data refer to time-series data and such applications can be surveyed in the security area. We will look for potential advancements in GNNs by fusing GNNs with other ML or graph theory techniques to solve industry-related issues.

Acknowledgments

This work was supported by the Gachon University research fund of 2021 (GCU-202110260001), and the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. 2021R1I1A3040361).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. A. Hogan, E. Blomqvist, M. Cochez, C. D'amato, G. De Melo, C. Gutierrez, et al., Knowledge graphs, *ACM Comput. Surv.*, **54** (2022), 1–37. https://doi.org/10.1145/3447772
2. M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, P. Vandergheynst, Geometric deep learning: going beyond euclidean data, *IEEE Signal Process Mag.*, **34** (2017), 18–42. https://doi.org/10.1109/MSP.2017.2693418
3. Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, P. S. Yu, A comprehensive survey on graph neural networks, *IEEE Trans. Neural Networks Learn. Syst.*, **32** (2021), 4–24. https://doi.org/10.1109/TNNLS.2020.2978386
4. S. Kearnes, K. McCloskey, M. Berndl, V. Pande, P. Riley, Molecular graph convolutions: moving beyond fingerprints, *J. Comput.-Aided Mol. Des.*, **30** (2016), 595–608. https://doi.org/10.1007/s10822-016-9938-8
5. A. Fout, J. Byrd, B. Shariat, A. Ben-Hur, Protein interface prediction using graph convolutional networks, in *Advances in Neural Information Processing Systems*, **30** (2017), 6533–6542. Available from: https://proceedings.neurips.cc/paper/2017/file/f507783927f2ec2737ba40afbd17efb5-Paper.pdf.
6. E. Choi, Z. Xu, Y. Li, M. Dusenberry, G. Flores, E. Xue, et al., Learning the graphical structure of electronic health records with graph convolutional transformer, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **34** (2020), 606–613. https://doi.org/10.1609/aaai.v34i01.5400
7. M. Zhang, Y. Chen, Link prediction based on graph neural networks, in *Advances in Neural Information Processing Systems*, **31** (2018), 5171–5181. Available from: https://proceedings.neurips.cc/paper/2018/file/53f0d7c537d99b3824f0f99d62ea2428-Paper.pdf.
8. C. Li, D. Goldwasser, Encoding social information with graph convolutional networks forPolitical perspective detection in news media, in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, **2019** (2019), 2594–2604. https://doi.org/10.18653/v1/P19-1247
9. T. Bian, X. Xiao, T. Xu, P. Zhao, W. Huang, Y. Rong, et al., Rumor detection on social media with Bi-directional graph convolutional networks, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **34** (2020), 549–556. https://doi.org/10.1609/aaai.v34i01.5393
10. M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, M. Welling, Modeling relational data with graph convolutional networks, in *The Semantic Web*, (2018), 593–607. https://doi.org/10.1007/978-3-319-93417-4_38 .

11. N. Park, A. Kan, X. L. Dong, T. Zhao, C. Faloutsos, Estimating node importance in knowledge graphs using graph neural networks, in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, (2019), 596–606. https://doi.org/10.1145/3292500.3330855

12. Z. Cui, K. Henrickson, R. Ke, Y. Wang, Traffic graph convolutional recurrent neural network: a deep learning framework for network-scale traffic learning and forecasting, *IEEE Trans. Intell. Transp. Syst.*, **21** (2020), 4883–4894. https://doi.org/10.1109/TITS.2019.2950416

13. H. Wu, L. Cheng, J. Jin, F. Yuan, Dialog acts classification with semantic and structural information, in *2019 International Conference on Intelligent Computing, Automation and Systems (ICICAS)*, (2019), 438–442. https://doi.org/10.1109/ICICAS48597.2019.00098

14. Y. T. Lin, M. T. Wu, K. Y. Su, Syntax-aware natural language inference with graph matching networks, in *2020 International Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, (2020), 85–90. https://doi.org/10.1109/TAAI51410.2020.00024

15. Z. Wang, H. Chang, 3D mesh deformation using graph convolution network, in *2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)*, (2019), 375–378. https://doi.org/10.1109/CCOMS.2019.8821790

16. P. Pradhyumna, G. P. Shreya, Mohana, Graph Neural Network (GNN) in image and video understanding using deep learning for computer vision applications, in *2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)*, (2021), 1183–1189. https://doi.org/10.1109/ICESC51422.2021.9532631

17. S. Zhang, H. Tong, J. Xu, R. Maciejewski, Graph convolutional networks: a comprehensive review, *Comput. Social Networks*, **6** (2019), 11. https://doi.org/10.1186/s40649-019-0069-y

18. W. Hamilton, Z. Ying, J. Leskovec, Inductive representation learning on large graphs, in *Advances in Neural Information Processing Systems*, **30** (2017). Available from: https://proceedings.neurips.cc/paper/2017/file/5dd9db5e033da9c6fb5ba83c7a7ebea9-Paper.pdf.

19. J. Gasteiger, A. Bojchevski, S. Günnemann, Predict then propagate: graph neural networks meet personalized pageRank, preprint, arXiv:1810.05997.

20. L. Page, S. Brin, R. Motwani, T. Winograd, The pageRank citation ranking: bringing order to the web, 1999. Available from: http://ilpubs.stanford.edu:8090/422/.

21. F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, K. Weinberger, Simplifying graph convolutional networks, in *Proceedings of the 36th International Conference on Machine Learning*, **97** (2019), 6861–6871. Available from: https://proceedings.mlr.press/v97/wu19e.html.

22. P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, preprint, arXiv:1710.10903.

23. P. Veličković, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, R. D. Hjelm, Deep graph infomax, preprint, arXiv:1809.10341.

24. G. Nikolentzos, M. Vazirgiannis, Random walk graph neural networks, in *Advances in Neural Information Processing Systems*, **33** (2020), 16211–16222. Available from: https://proceedings.neurips.cc/paper/2020/file/ba95d78a7c942571185308775a97a3a0-Paper.pdf.

25. B. Nica, A brief introduction to spectral graph theory, 2016. Available from: https://arxiv.org/pdf/1609.08072.pdf.
26. I. Benjamini, L. Lovász, Harmonic and analytic functions on graphs, *J. Geom.*, **76** (2003), 3–15. https://doi.org/10.1007/s00022-033-1697-8
27. W. W. Zachary, An information flow model for conflict and fission in small groups, *J. Anthropol. Res.*, **33** (1977), 452–473. https://doi.org/10.1086/jar.33.4.3629752
28. J. Leskovec, A. Krevl, SNAP datasets: Stanford Large Network Dataset Collection, 2014. Available from: http://snap.stanford.edu/data.
29. A. Bharali, An analysis of Email-Eu-Core network, *Int. J. Sci. Res. Math. Stat. Sci.*, **5** (2018), 100–104. https://doi.org/10.26438/ijsrmss/v5i4.100104
30. D. Grattarola, C. Alippi, Graph neural networks in TensorFlow and Keras with Spektral, *IEEE Comput. Intell. Mag.*, **16** (2021), 99–106. https://doi.org/10.1109/MCI.2020.3039072
31. T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, preprint, arXiv:1609.02907.
32. F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, *IEEE Trans. Neural Networks*, **20** (2009), 61–80. https://doi.org/10.1109/TNN.2008.2005605
33. X. Zhu, Z. Ghahramani, J. D. Lafferty, Semi-supervised learning using Gaussian fields and harmonic functions, in *Proceedings of the Twentieth International Conference on Machine Learning*, (2003), 912–919. Available from: https://dl.acm.org/doi/10.5555/3041838.3041953.
34. A. Hagberg, D. S. Chult, P. J. Swart, Exploring network structure, dynamics, and function using networkx, in *Proceedings of the 7th Python in Science conference (SciPy 2008)*, (2008), 11–15. Available from: https://conference.scipy.org/proceedings/scipy2008/paper_2/.
35. L. He, C. T. Lu, J. Ma, J. Cao, L. Shen, P. S. Yu, Joint community and structural hole spanner detection via harmonic modularity, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (2016), 875–884. https://doi.org/10.1145/2939672.2939807
36. S. Luan, C. Hua, Q. Lu, J. Zhu, M. Zhao, S. Zhang, et al., Is heterophily a real nightmare for graph neural networks to do node classification? preprint, arXiv:2109.05641.
37. Q. Yao, R. Y. M. Li, L. Song, Construction safety knowledge sharing on YouTube from 2007 to 2021: Two-step flow theory and semantic analysis, *Saf. Sci.*, **153** (2022), 105796. https://doi.org/10.1016/j.ssci.2022.105796
38. Q. Yao, R. Y. M. Li, L. Song, M. J. C. Crabbe, Construction safety knowledge sharing on Twitter: a social network analysis, *Saf. Sci.*, **143** (2021), 105411. https://doi.org/10.1016/j.ssci.2021.105411

AIMS Press