EAI Endorsed Transactions on
Internet of Things

Research Article

A Self-Supervised GCN Model for Link Scheduling in Downlink NOMA Networks

Caiya Zhang[1,*], Fang Fang[1], Congsong Zhang[2]

[1]Department of Computer Science, Western University, Canada
[2]Department of Computer Science, University of British Columbia Okanagan, Canada

Abstract

INTRODUCTION: Downlink Non-Orthogonal Multiple Access (NOMA) networks pose challenges in optimizing power allocation efficiency due to their complex design.
OBJECTIVES: This paper aims to propose a novel scheme utilizing Graph Neural Networks to address the optimization challenges in downlink NOMA networks.
METHODS: We transform the optimization problem into an optimal link scheduling problem by modeling the network as a bipartite graph. Leveraging Graph Convolutional Networks, we employ self-supervised learning to learn the optimal link scheduling strategy.
RESULTS: Simulation results showcase a significant enhancement in power allocation efficiency in downlink NOMA networks, evidenced by notable improvements in both average accuracy and generalization ability.
CONCLUSION: Our proposed scheme demonstrates promising potential in substantially augmenting power allocation efficiency within downlink NOMA networks, offering a promising avenue for further research and application in wireless communications.

Received on 13 December 2023; accepted on 5 May 2024; published on 13 May 2024

Keywords: GNNs, graph convolutional neural networks, Non-orthogonal multiple access (NOMA), link scheduling.

doi:10.4108/eetiot.6039

1. Introduction

Communication networks constitute fundamental infrastructure in contemporary society. According to the Cisco Annual Internet Report (2018-2023) white paper, over two-thirds of the global population will have access to the Internet by 2023 [1]. Therefore, in order to meet the rapidly increasing demand for urban communications, it is crucial to enhance power allocation efficiency and spectral efficiency. To tackle these challenges, various kinds of models for communication networks have been proposed. In the field of wireless communication, traditional wireless access approaches such as Frequency Division Multiple Access (FDMA), Time Division Multiple Access (TDMA), and Code Division Multiple Access (CDMA) have reached maturity in supporting multiple user access [2]. However, these Orthogonal Multiple Access (OMA)-based approaches have limitations in terms of spectral and power efficiency. To address these issues, Non-Orthogonal Multiple Access (NOMA) has been proposed to improve the efficiency of wireless communication systems, positioning it as a potential candidate for future upgraded communication networks [3].

NOMA networks enable multiple users to share the same time-frequency resources for data transmission and employ Successive Interference Cancellation (SIC) techniques to mitigate interference at the receiver side. Concerning power allocation, optimization strategies based on NOMA systems has garnered significant attention. Numerous techniques have been proposed to enhance power allocation efficiency in wireless communication networks, with machine learning emerging as one of the noteworthy approaches in recent decades. Traditional machine learning algorithms have proven effective in network design, traffic forecasting, and resource allocation within wireless communication contexts [4].

*Corresponding author. Email: czhan685@uwo.ca

Nonetheless, ensuring the simultaneous preservation of both the computational tractability and the veracity of the topological features in wireless networks remains a formidable task for conventional machine learning methods. Graph neural networks (GNNs) are a class of machine learning models designed to process graph-structured data by capturing and leveraging the spatial relationships between data points [5]. By harnessing the structural information inherent in wireless networks, GNNs can effectively discern the underlying patterns and dynamics of wireless channels and signals, ultimately leading to improved performance in various communication tasks. In the study conducted in [5], GNNs have been used to generalize learning outcomes across dynamic network structures, enabling effective modeling of non-Euclidean data, such as wireless communication networks. Therefore, GNNs hold significant potential to make a meaningful impact in the field of wireless communications.

Motivated by the analysis above, in this paper, we investigate the problem of maximizing the link scheduling efficiency in downlink NOMA networks. Specifically, we propose a Graph Convolutional Network (GCN) model with self-supervised learning to improve the efficiency of NOMA networks. Our proposed scheme generates near-optimal strategies and exhibits generalization ability.

The rest of the paper is organized as follows. Section 2 defines the system model and the problem formulation of this work. Section 3 provides details on the experimental data and the solution to the proposed model. Section 4 presents the results of the experiments and discusses the validity and limitations of the results. Finally, Section 5 draws the conclusions.

2. System Model and Problem Formulation

2.1. Channel Model

For wireless communication in contemporary urban areas, the Rayleigh fading channel stands out as a widely used type of channel model. This is because Rayleigh fading relies on reflections during propagation, including numerous reflections off buildings, making it well-suited for assessing radio propagation in densely populated cities. The Rayleigh fading model assumes that the signal amplitude across such communication channels will vary or fade randomly according to the Rayleigh distribution. Specifically, this distribution characterizes the radial component of the sum of two uncorrelated Gaussian random variables. Therefore, in this paper, we assume that the channels between the Base Station (BS) and all the users are Rayleigh channels and the channel model is defined as follows:

$$\mathbf{h}_k^d = \widetilde{\mathbf{h}}_k^d / \sqrt{d_{d,k}^{\alpha}}, k \in K, \tag{1}$$

where $\widetilde{\mathbf{h}}_k^d \in \mathbb{C}^{M\times 1}$ consists of M independent and identically distributed (i.i.d.) elements following a complex $\mathcal{CN}(0,\ 1)$ distribution, $d_{d,k}$ represents the distance between the BS and the k-th user, and α represents the path loss exponent [6].

2.2. System Model

We consider a downlink single-cell NOMA network with Rayleigh fading channels, as illustrated in Fig.1. In this setup, a single BS equipped with M sub-channels is positioned at the center of the cell, K single-sub-channel users are uniformly distributed within the cell [7]. The BS transmits signals to users through these sub-channels, where M≤K and K∈ {1, 2,..., K}. Since we are exclusively focusing on the downlink in this scenario, we model the wireless network as an undirected graph. In this graph, the nodes represent users and base stations with ideal communication links, while the edges signify the presence of detrimental interference links.

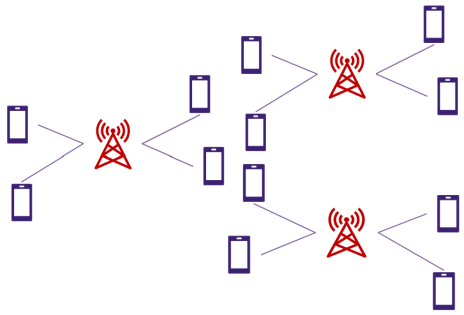

Figure 1. System Model with Three BSs.

To evaluate the generalization performance of the approach proposed in this letter, we also consider a multi-BS model that is more complex. The allocation of sub-channels and users within the cellular coverage of each BS adheres to the specifications of a single BS model, as exemplified by the three-BS configuration illustrated in Fig. 2.

2.3. Problem Formulation

After presenting the system model and the channel model, we proceed to address the optimal power allocation problem within the proposed system. In order to optimize power allocation, our goal is to maximize the sum rate achieved by each user. Therefore, we formulate the sum rate maximization problem as follows:

$$\max_{\{p_{m,n}\}} \quad EE^l = \sum_{n=1}^{N} \omega_{m,n} B_n \log_2 \left(1 + \frac{G_{m,n} p_{m,n}}{\sum_{i=1}^{m-1} p_{i,n} G_{i,n} + 1}\right) \tag{2}$$

which can be simplified as:

$$\max_{p} \quad EE^l = \sum_{l \in L} \omega B \log_2 (1 + r_l) \tag{3}$$

Since signal-to-interference-plus-noise ratio $\frac{G_{m,n} p_{m,n}}{\sum p_{i,n} G_{i,n} + 1}$ is not the focus of this work, we would like to simply replaced it with r_l in equation (3). Hence, when considering a link scheduling strategy in Equation (3), our attention is directed solely towards:

$$\max_{p} \sum_{l \in L} \omega \log_2 (1 + r_l) \tag{4}$$

where l $\in$ {1, 2, ..., L_{max}} and L_{max} represents the maximum number of iterations [8]. Consequently, the optimization problem has been defined as the binary classification problem of link connection, which we will address using a GCN in the next section.

3. GCN-based Model for Link Scheduling

In this work, we approach the link scheduling problem as a classification task. Each end-to-end pair is considered as a node, and the interference links between these pairs are represented as edges. Our objective is to map the channel matrix to binary decisions, where '1' indicates an active end-to-end pair, and '0' indicates an inactive one . To achieve this, we employ a self-supervised learning strategy. This involves random removal or modification of edges and features within the original graph, and the model is trained using unsupervised learning to predict these edges. Once trained, the model can perform node classification tasks, determining the status of each end-to-end pair. In summary, this approach provides a novel solution for the link scheduling problem in wireless communication networks.

3.1. Objective Task

The objective task at hand can be divided into two parts: link prediction and node classification. Specifically, for the link prediction task, we employ a random modification strategy to randomly alter the edges and node features within the graph [9]. This strategy aids the GCN in learning potential node representations from the graph data itself. The goal of the link prediction task is to anticipate the presence or absence of a connection between two nodes. This can be formulated as follows:

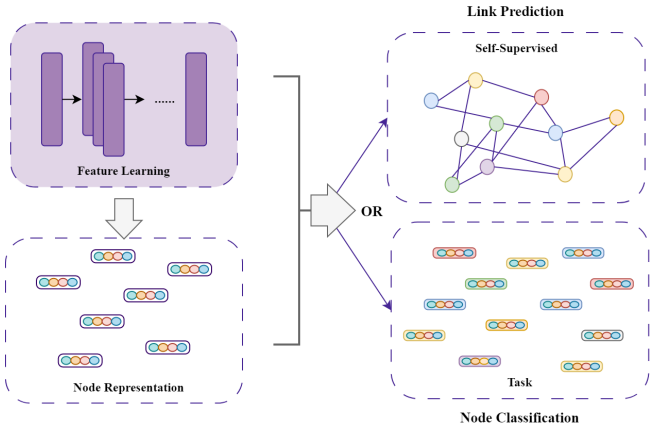

Figure 2. Overview of the Proposed Model

$$\hat{A} = sigmoid(H^i (H^i)^T), \tag{5}$$

where H^i is the node representation obtained by GCNs, $\hat{A}$ is the reconstructed adjacency matrix, and A is the adjacency matrix. We use the weighted cross-entropy as the loss function.

For the classification task, the aim is to classify the node representations learned from the feature learning module into binary classes. This module comprises several graph convolution layers and a Softmax function, which can be formulated as follows:

$$Z = softmax(H^i), \tag{6}$$

and the loss function is designed as:

$$L = -\sum_{i \in V_l} \sum_{j=1}^{N} M_{ij} log(Z_{ij}), \tag{7}$$

where M represents the label matrix.

3.2. Self-supervised Strategy

The concept of self-supervised learning involves training a model with a substantial amount of unlabeled data to acquire deep representations of input samples [10]. This learning approach has achieved remarkable performance in natural language and image learning tasks, and recently, there has been a growing interest in extending this approach to graph-related tasks using graph neural networks.

Our model encompasses two key tasks: self-supervised link prediction and node classification. In the self-supervised training phase,we introduce

random modifications to the input graph by removing certain edges and overwriting node information. Subsequently, the node representations are generated through a GCN network. These representations are then utilized to reconstruct the graph, and we employ a weighted cross-entropy loss function to fine-tune the model's parameters. As for the node classification task, it leverages the learned features from the link prediction phase to classify the nodes. Our approach can be effectively applied to a range of graph-related tasks, resulting in improved performance.

3.3. Feature Learning

The sub-module of feature learning constitutes the main component of the entire model. During the self-supervised training phase, this sub-module engages in joint training, taking the modified graph data as input. These modifications render the graph incomplete, and the primary objective of this phase is to explore the node representations within the graph, along with the embedded structural information. Through this process, the sub-module can self-supervise its learning from the modified graph data effectively. Furthermore, when the sub-module is employed for classification tasks, it undergoes end-to-end training alongside the classification module.

Specifically, the implementation of feature learning consists of several GCN layers. GCNs define the graph convolutions based on the spatial relationships between their nodes, and each convolutional layer can only handle first-order neighborhood information. By stacking multiple convolutional layers, it is possible to achieve multi-order neighborhood information transfer. We utilize this feature to understand how other nodes influence a given node.

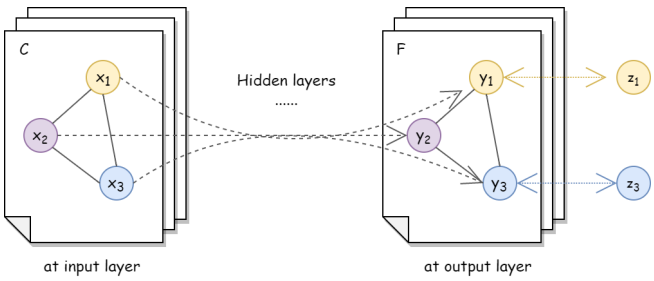

Figure 3. Operation between Layers

As shown in Fig. 3, we feed a single graph into the GCN model.As it progresses layer by layer, the features of each node transform from "x" to "y". Importantly, the connectivity between nodes remains shared and is not dependent on the number of layers. This relationship can be expressed as:

$$H^{l+1} = \sigma(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}H^{l}W^{l}) \tag{8}$$

where $\widetilde{A} = A + I$ and I is the unit matrix; $\widetilde{A}$ is the degree matrix of $\widetilde{A}$, i.e. $\widetilde{D}_{ii} = \Sigma_j \widetilde{A}_{ij}$; H represents the features at each layer; and σ is the nonlinear activation function [1]. This formula is also the foundation on which GCNs rely to extract features.

4. Results

This section presents the performance of the proposed self-supervised GCN approach. We demonstrate the superior performance of this self-supervised GCN by comparing it with two baseline models. Additionally, we analyze the robustness of the proposed model, considering that real-world application scenarios may vary.

The testing experiments were conducted utilizing an Nvidia Geforce GTX 3070 Ti GPU, and the model was implemented using Pytorch. The distance between the base station and the k-th user was generated within the range 45 to 50, with the path loss exponent α set to 2 [6]. Furthermore, we employed the Adam optimizer with a learning rate of 0.01.

In this work, the performance of proposed approach is compared with the following 2 benchmark schemes:

Supervised GCN: the structure and parameters of chosen supervised GCN are the same as those of the proposed self-supervised GCN, while the loss function is defined as the negative sum rate in [12].

Self-supervised Multilayer Perceptron (MLP): The selected MLP uses the same loss function and the same scale of network layers as proposed self-supervised GCN.

4.1. Performance with different sizes of training samples

Table 1 illustrates the test performance of proposed self-supervised GCN method with 20 BS-user groups for different numbers of training samples. It is evident that as the number of training samples increases, the test performance of the proposed approach improves. With 1000 training sample, the average accuracy reaches near 0.85, which is close to that of supervised GCN. These comparison results indicate that the proposed self-supervised GCN method for link scheduling problem performs well and approaches the results achieved through supervised learning.

4.2. Performance with different model scales

To demonstrate the generalization abilities of the proposed method, we tested models trained with 500 training samples under varying numbers of base stations. The proposed self-supervised GCN approach yields better test results for models of both smaller

Table 1. Average Test Performance Comparisons with Different Number of Training Samples

Number of Training Samples	100	200	500	1000
	Accuracy			
Self-Supervised GCN	**0.6050**	**0.7696**	**0.8204**	**0.8275**
Supervised GCN	0.6972	0.7995	0.8301	0.8566
Self-Supervised MLP	0.5767	0.6904	0.7025	0.7576

and larger scale. The test performance on the 10-group model even surpasses that of the supervised GCN, indicating the robust generalization ability of the proposed self-supervised GCN method.

Table 2. Average Test Performance Comparisons with Different Network Scales

Number of Cells	#BS=10	#BS=20	#BS=50
	Accuracy		
Self-Supervised GCN	**0.8400**	**0.8204**	**0.7863**
Supervised GCN	0.8398	0.8301	0.7975
Self-Supervised MLP	0.7286	0.7025	0.6522

4.3. Robustness to Corrupted Input Features

To show the robustness of proposed model to corrupted input features, tests with fixed missing proportion of the interference Channel State Information (CSI) are taken. The performance of the proposed GNN model with partial CSI was compared to the performance achieved with full CSI, and the ratio of the two schemes is depicted in Fig. 4. The results indicate that even with 40% of the CSI missing, the proposed GNN system exhibited an accuracy of 0.82 and a sum rate of 0.88 compared to the performance with complete CSI. These findings provide evidence of the robustness of the proposed GNN system against the graph feature corruption.

4.4. Experiments Settings

The Fig. 5 depicts the performance of proposed method under varying batch sizes and epochs. In general, larger batch sizes can provide a greater number of negative samples for comparison, which trands to lead to improved training outcomes.

What is more, lengthier training cycles can provide more negative samples for each sample by offering a greater range of data splits, which can also enhance the performance of a model. In the training process

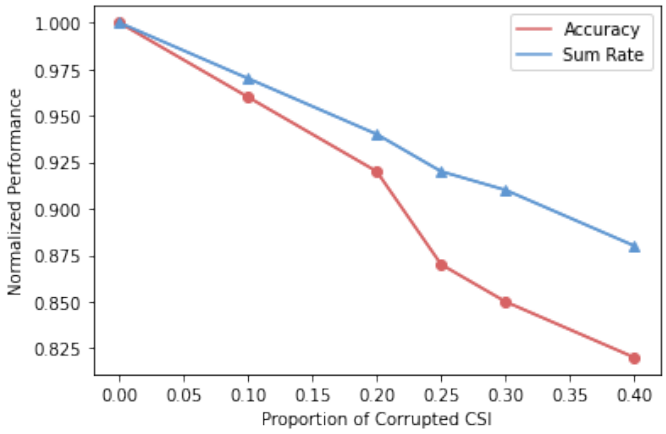

Figure 4. Operation between Layers

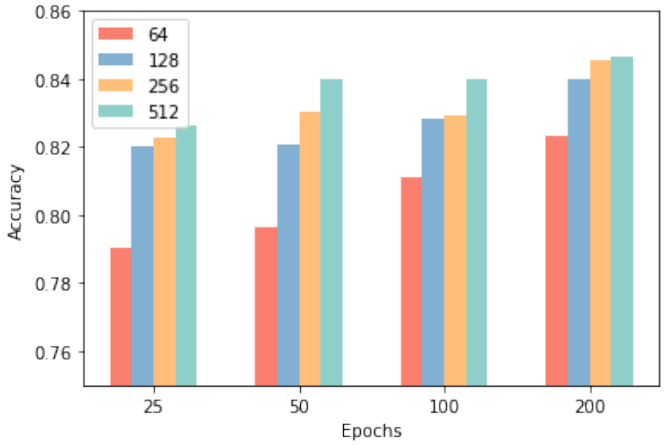

Figure 5. Performance with different batch size and epochs

of link prediction section, we adopt the commonly used batch size of 128 and epochs of 25. Nevertheless, the performance of proposed model may be further enhanced by using larger batch sizes and longer training period.

4.5. Limitations and Validity

Limitations: In real-world wireless network environments, numerous potential interferences may introduce certain biases into the data sets used for both training and testing the model. Therefore, the application of the proposed model to actual wireless networks may encounter challenges.

Validity: Machine learning-based prediction techniques can enhance link allocation efficiency in wireless networks. When integrated with NOMA systems, our approach can serve as a potent tool for predicting efficient link assignment strategies in wireless networks, thereby exhibiting promising applications. Furthermore, the proposed model demonstrates strong

performance on test data of varying scales and exhibits satisfactory generalization ability.

5. Conclusions

In this work, a graph convolutional neural network is introduced to investigate the optimal power allocation efficiency in downlink NOMA networks by optimizing the link scheduling strategy. Although our proposed self-supervised learning model has achieved competitive test accuracy with low training costs and limited computing complexity, it still relies on efficient simulation of training data and has its limitations. Nevertheless, this underscores the effectiveness and potential value of machine learning methods in addressing wireless communication challenges. In the foreseeable future, it is likely that self-supervised and even unsupervised learning frameworks, along with preprocessing methods tailored for graph data, such as diffusion convolution for both directed and un-directed graphs, will continue to make significant strides in improving wireless communication solutions.

6. Declarations of competing interest

The authors declare that they do not possess any known competing financial interests or personal relationships that might have influenced the work presented in this paper.

Acknowledgement. Thanks editors and anonymous reviewers for their valuable suggestions, which have greatly contributed to the improvement of this work.

References

[1] Jiang W. Graph-based deep learning for Communication Networks: A survey. Computer Communications. 2022 Mar;185:40–54. doi:10.1016/j.comcom.2021.12.015

[2] Reddy P.V., Reddy S., Sawale R. D., Narendar P., Duggineni C., et al. Analytical Review on OMA vs. noma and challenges Implementing Noma. 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC). 2021 Oct 7; doi:10.1109/icosec51865.2021.9591629

[3] Ding Z., Adachi F., and Poor H.V.. The application of MIMO to Non-Orthogonal Multiple Access. IEEE Transactions on Wireless Communications. 2016 Jan;15(1):537–52. doi:10.1109/twc.2015.2475746

[4] Zhang C., Patras P., and Haddadi H. Deep Learning in Mobile and wireless networking: A survey. IEEE Communications Surveys & Tutorials. 2019;21(3):2224–87. doi:10.1109/comst.2019.2904897

[5] Wu Z., Pan S., Chen F., Long G., Zhang C., and Yu P.S. A comprehensive survey on Graph Neural Networks. IEEE Transactions on Neural Networks and Learning Systems. 2021 Jan;32(1):4–24. doi:10.1109/tnnls.2020.2978386

[6] Ximing X., Shiyu J., and Zhiguo D. A Reinforcement Learning Approach for an IRS-assisted NOMA Network. 2021 Jun

[7] Fang F., Zhang H., Cheng J., Roy S., and Leung V.C. Joint user scheduling and power allocation optimization for energy-efficient NOMA systems with imperfect CSI. IEEE Journal on Selected Areas in Communications. 2017 Dec;35(12):2874–85. doi:10.1109/jsac.2017.2777672

[8] Fang F., Ding Z., Liang W., and Zhang H. Optimal Energy Efficient Power Allocation with user fairness for uplink MC-noma systems. IEEE Wireless Communications Letters. 2019 Aug;8(4):1133–6. doi:10.1109/lwc.2019.2908912

[9] Zeng J. and Xie P. Contrastive self-supervised learning for graph classification. Proceedings of the AAAI Conference on Artificial Intelligence. 2021 May 18;35(12):10824–32. doi:10.1609/aaai.v35i12.17293

[10] Thomas N.K. and Max W. Semi-Supervised Classification with Graph Convolutional Networks. 2016 Sep

[11] Qikui Z., Bo D., and Pinglun Y. Self-supervised Training of Graph Convolutional Networks. 2020 June

[12] Shen Y., Shi Y., Zhang J., and Letaief K.B. A graph neural network approach for Scalable Wireless Power Control. 2019 IEEE Globecom Workshops (GC Wkshps). 2019 Dec; doi:10.1109/gcwkshps45667.2019.9024538