Proceedings of the 10th World Congress on Electrical Engineering and Computer Systems and Sciences (EECSS'24)
Barcelona, Spain - August 19-21, 2024
Paper No. MVML 113
DOI: 10.11159/mvml24.113

Classification of complex network using bidirectional LSTM

María del Carmen Soto Camacho[1], Aldo Ramirez-Arellano[1]
[1]Sección de Estudios de Posgrado e Investigación-Unidad Profesional Interdisciplinaria de Ingeniería y Ciencias Sociales y Administrativas /Instituto Politécnico Nacional
Alcaldía Iztacalco, Ciudad de México, México
msotoc0600@alumno.ipn.mx; aramirezar@ipn.mx

Abstract **–** Classifying complex networks has become highly significant in network analysis. Graph Neural Networks (GNNs) have successfully succeeded in this particular task. However, GNNs suffer from limited network representation as they rely solely on scalar-based features for node properties. GNNs' message-passing methods suffer from oversmoothing and lack global information on complex networks. Data augmentation is also impossible for GNN, so obtaining reasonable classifications in small datasets is an open issue. Deng's entropy of complex networks captures the network's topology and the valuable information generated by the nodes and edges to solve the problems arising from complex networks and unlock their full potential. Our proposed method utilizes Deng's entropy to calculate an entropy sequence incorporating local and global features at multiple scales. We then combine the entropy sequences for nodes and edges into a matrix fed into a bidirectional bLSTM network to perform complex network classification.

Keywords**:** LSTM, Deng entropy, complex network, classification.

1. Introduction

Complex network classification is a process that involves associating a given complex network with a label. A function needs to be estimated that maps each complex network to its corresponding label [1]. In supervised classification, a model is trained using labeled complex networks, and then the same model is used to assign category labels to unlabeled complex networks[2]-[4].

Calculating topological metrics, such as diameter, assortativity, degree, and cluster coefficient, becomes increasingly challenging as the number of complex networks to be classified grows[5]. In addition, the effectiveness of the topological features in classifying complex networks is highly dependent on the domain of the complex network[6].

One approach to classifying complex networks is based on kernel function. It involves collecting a set of complex network features or frequent subnetworks and then using a similarity measure to compare two complex networks [1], [7], [8]-[9]. It has been observed that the limited availability of labeled complex networks used for training results in poor classification performance due to an increase in metrics compared to the number of complex networks[5]. A detailed survey and experimental analysis of the current complex network kernels indicate that their performance is influenced by various features of complex networks, including density, size, number of complex networks, global structure, and node attributes[10].

Graph neural networks (GNN) are a type of neural network that can depict network topology and generate information aggregation operations. The architecture of GNN consists of a convolutional and pooling layer. The convolutional layer extracts the local structure of the complex network to add node features, and the pooling layer selects these features to be embedded in a complex network representation for the classification task[3]. Complex networks are non-Euclidean objects made up of a set of nodes and their relationships (edges), and as a result, the GNN can lose local and global information when convolution and pooling operations are used[11]. Various approaches have been designed[12] to address these complications. For example, neighbor aggregation (message passing), message passing via recurrent neural networks, and attention mechanisms have been proposed[13]. Other approaches employ vector representation of complex networks using Word2Vec and Doc2Vec to extract relevant features from subnetworks to build the feature vector representation used for the classification task[14]-[15].

It is crucial to have high-quality complex network representations that capture the various topology structures while accounting for the uncertainty in each complex network to classify complex networks effectively. Additionally, the

representations should preserve the features of the nodes and edges from both local and global perspectives. To address these requirements, we have proposed a complex network classification method that utilizes Deng entropy, which is calculated by applying a defined mass of function of nodes and edges. This approach feeds into the classification task's bidirectional long short-term memory (bLSTM). Our study's main contributions can be summarized as follows:

- The topology of the complex network is represented by two data sequences that correspond to the calculation of the Deng entropy of nodes and edges. To this end, a mass is introduced for nodes and edges.
- The mass functions are based on an algorithm called box-covering. It involves using different sizes of boxes to analyze the complex network, with the diameter of the complex network plus one being used as the box size $\varepsilon = 1\ to\ \varepsilon = \Delta$. By computing from lower ε values to the maximum one, the technique can capture the local and global topology of the complex network at different scales.
- We tested our approach on real-world and synthetic complex networks to evaluate the effectiveness of DE-bLSTM classification on different complex network domains. Our results demonstrate that the approach is effective for intra-domain complex network classification and cross-domain classification.

The article is structured as follows: Section 2 provides an overview of the related work, while Section 3 outlines the background information necessary to introduce the DE-bLSTM approach in Section 4. The results of the complex network classification using De-bLSTM are presented in Section 5, and finally, Section 6 concludes our findings.

2. Related work

Complex network classification is a crucial task that assigns class labels to complex networks using models based on training data[4]. It helps solve real-life classification problems[5] and has many models for predicting the class of unknown complex networks or understanding complex structures[9]. In this section, we will briefly review network classification methods.

2.1. Kernel methods

Kernel functions are popular for measuring similarity between complex networks. Various complex network kernels have been proposed for different applications[10]. In bioinformatics, protein function can be predicted using secondary and tertiary protein structures represented by plane-labeled geometric complex networks [2]. A classification prediction model based on these structures can help identify basic protein functions.

These methods have been developed to calculate complex network similarity and predict possible complex network labels. The kernel methods are designed under the R-convolution framework[5], [16]. However, some methods only consider local complex network properties, such as subtrees, and cannot capture global complex network properties, such as connected components and cycles. Wasserstein's WL method was proposed. K-WL is based on the k-dimensional Wes-Feiler-Lehman algorithm, which balances global and local subtree kernels to analyze the entire complex network[17]-[18]. A survey of kernel approaches and their performance comparison can be found in[10].

2.2. GNN methods

Graph Neural Network (GNN) is a powerful tool for learning node and complex network embeddings. GNN operates through message passing (MPNN) between neighboring nodes, enabling it to learn complex network structures [19]. However, GNN's performance tends to be poor when predicting tasks requiring long-range interactions, and the structure of the complex network results in long-range neighbors that grow exponentially. It also suffers from the over-smoothing problem and lack of global structural information[3].

Researchers have improved GNN methods for complex network classification by proposing different methods, such as Graph Convolutional Networks (GCN) and Graph Attention Networks (GAT)[17], [21]. These approaches learn effective node-level representations and use centrality-based edge-importance for complex network compression, which filters out trivial structures and perturbations in the input complex networks. To achieve complex network classification tasks, researchers employ graph-level embeddings through maximum or average pooling[18].

2.3. Graph pooling methods

This method extracts essential structural information with different granularities using multiple complex network convolution and pooling operations. It considers both node features and complex network topology. GNN methods for complex network classification can be categorized into global and hierarchical pooling[20].

Global pooling methods are used in GNN to calculate the complex network representation vector. However, the expressive power of these methods is limited, and they can result in inaccurate classification[22]. A new global pooling method has been proposed to address these issues and improve global representation and classification accuracy, increasing attention to modeling sequence data[23].

Hierarchical pooling methods reduce a complex network size by deleting or fusing nodes. It condenses local neighborhoods into a single node[20], extracting important structural and node information. An extra pooling operation reduces complex network size, and the updated features and reconstructions of the adjacency matrix improve the complex network representations[22]. It captures hierarchical information compositionally by aggregating messages on an increasingly coarser complex network.

2.4. Capsule network method

Zuo [3] and Yin [24] , have proposed new methods for graph classification using capsule networks. Zuo's method converts node features into capsules and uses dynamic routing to generate graph capsules, while Yin's CapsualGNN (GCN) groups nodes as capsules and generates multiple embeddings for each graph. Both methods introduce techniques to capture global and structural information between nodes.

2.5. Graph pyramid method

Ji [25] and Lu [20] proposed novel graph-based neural network architectures for different tasks. Ji utilized a Pyramid Graph Transformer (PyGT) for handwritten Chinese character recognition, while LU proposed a Feature Pyramid-based Graph Convolutional Neural Network for Graph Classification (FPGCN-GC) method. Both approaches improved the accuracy of their respective tasks by incorporating graph attention mechanisms and residual connections.

3. Preliminaries

This section will define the notations used in each of the sections of the paper and explain the main terms. We will use "graph" and "complex network" interchangeably. However, the main difference is that complex networks have unique topological features such as small-world property[26], scale-free degree distribution[27], and fractality[28]. They are the backbone of complex systems, including social, technological, and biological networks[29].

A complex network is defined as:

$$G= (V, E),$$

where V is a finite set of nodes and E is a symmetric and reflexive relation on V. Entropy is an instrument for measuring the complexity of systems, including complex networks, so the following section introduces the concept of entropy.

3.1. Shannon entropy

Shannon entropy is obtained from a probability distribution $P = \{p_1, p_2, \dots, p_N\}$, under a probability space $X = \{x_1, x, \dots, x_N\}$, by:

$$I = -\sum_{i=1}^{N} p_i \, ln \, ln \, p_i \,, \tag{1}$$

where N is the total number of probabilities p_i and $\sum_{i=1}^{N} \quad p_i = 1$. The maximum Shannon entropy is achieved when P is observed to be a uniform distribution; consequently, when $p_i = \frac{1}{N} \quad I_{max} = ln\ ln\ N$

3.2. Deng entropy

Deng entropy is a probabilistic measure of uncertainty that considers non-specificity and discord in basic probability assignment (BPA). We can define a set X consisting of N mutually exclusive and collectively exhaustive events denoted by $X = \{\theta_1\ , \theta_2, \dots, \theta_N\}$, where X is defined as a frame of discernment. The power set of X is:

$$2^X = \{\emptyset, \dots, \theta_1\}, \dots, \{\theta_N\}, \{\theta_1, \theta_2\}, \dots, \{\theta_1, \theta_2, \dots, \theta_{N-1}\}, \dots, X$$

The frame of discernment is denoted by X, and the mass function is represented by the map m from 2^X to [0, 1]. The mass function is also known as BPA, and it must fulfil the following conditions:

$$m(\emptyset) = 0, \sum_{A \in 2^X} \quad m(A) = 1. \tag{2}$$

where A indicates a focal element of m and $A \in 2^X$.

Having m_1 and m_2 as two BPA; to join the pieces Dempster's combination rule is applied [30]

$$m(A) = \frac{1}{1-K} \sum_{B \cap C = A} \quad m_1(B) m_2(C), \tag{3}$$

Where B is the focal element of m_1, C is the focal element of m_2, and the conflict coefficient of the two BPA's is K, which is given by:

$$K = \sum_{B \cap C = 0} \quad m_1(B) m_2(C), \tag{4}$$

Only when $K < 1$ Eq. (3) is applicable.

Deng's entropy is a measure of uncertainty for BPA[31], It is defined as follows:

$$I_D = -\sum_{A \in 2^x} \quad m(A) \frac{m(A)}{2^{|A|} - 1} = \sum_{A \in 2^x} \quad m(A) \left(2^{|A|} - 1\right) - \sum_{A \in 2^x} \quad m(A) m(A) \tag{5}$$

Where m is the mass function, that is defined in the frame X, A is a focal element of m, and $|A|$ denotes A's cardinality. When the mass function is

$$m(A) = \frac{2^{|A|} - 1}{\sum_{A \in 2^x} \quad 2^{|A|} - 1} \tag{6}$$

Deng's entropy aims its maximum[32]

$$I_{Dmax} = log\ log\ (\sum_{A \in 2^x} \quad 2^{|A|} - 1) \tag{7}$$

The Shannon entropy is obtained by degenerating the BPA into a probability distribution. This verifies that the focal element is a singleton.. Consequently, $|A| = 1$. From Eq.(5), the term $\sum_{A \in 2^x} \quad m(A)(2^{|A|} - 1)$ is the measure of total non-specificity, and the measure of discord in the mass function m among various focal elements is the term $-\sum_{A \in 2^x} \quad m(A) m(A)$. Accordingly, Deng's entropy is a composite measure.

4. Deng entropy and complex network classification

4.1. Deng entropy of networks

To adopting I_D from Eq. (5), It is possible to obtain the Deng entropy of networks by:

$$I_D(\varepsilon) = \sum_{A_i \in X}^{N_b(\varepsilon)} \sum_{i=1} m(A_i) \frac{m(A_i)}{2^{/A_i/} - 1} \tag{8}$$

Where X represents the set of nodes or edges that are partitioned into subsets or boxes A_i, $N_b(\varepsilon)$ is the minimum number of boxes that cover the network for the size of ε, and $/A_i/$ is the cardinality of A_i. Boxes are discovered by the coloring box-covering algorithm. The boxes of A_i consist of different nodes for each ε.

The mass function of BPA for nodes is determined by:

$$m_V(A_i) = \frac{/A_i/}{N_V} \tag{9}$$

Where $/A_i/$ represents the number of nodes in A_i and N_V is the number of nodes contained in the network. The box-covering algorithm does not allow overlapping boxes, so each A_i is a mutually exclusive subset of X, thus, $\sum_{A_i \in X} m_V(A_i) = 1$.

For edges, the mass function is defined by

$$m_E(A_i) = \frac{inDe(A_i) + \frac{outDe(A_i)}{2}}{N_E} \tag{10}$$

Where $inDe(A_i)$ represents the number of edges that connect the nodes in A_i, and $outDe(A_i)$ represents the number of edges that connect the nodes of the box A_i with nodes in another box A_j, for $i = 1, 2, \dots, N_b(\varepsilon)$ and $i \neq j$, and N_E contains the total edges in the network.

Since $I_D(\varepsilon)$ is computing in the range $\varepsilon = [1, \Delta]$, where Δ is the diameter of the network plus one, the output is a dispersion ε vs $I_D(\varepsilon)$ that can be stored as a vector containing an entropy series.

4.2. Bidirectional Long Short-Term Memory for complex network classification

A recent architecture called De-bLSTM, which uses Deng entropy and bLSTM, is used for complex network classification. Input sequences are obtained from the input sequence layer, which includes the Deng entropy of nodes and edges. This layer normalizes each sequence, which is used as the input to the bLSTM layer with 350 hidden units. An entirely connected layer follows the bLSTM, and the softmax layer ensures that the class probabilities add up to one. Figure 1 depicts the training and testing process for the bLSTM architecture.

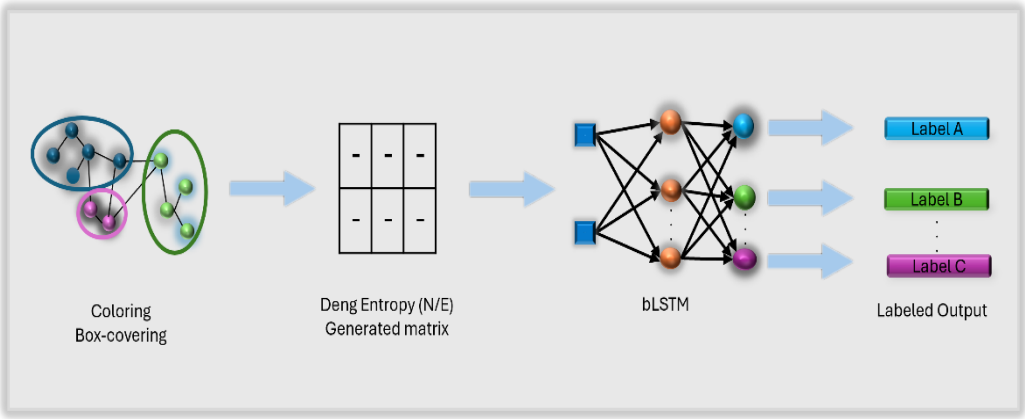

Fig. 1: The training and testing process of De-bLSTM

The experiments were carried out with the use of MATLAB R2023a on a CPU with Intel Core i9 12900KF, 128 GB RAM, and two GPU GeForce RTX 4090 with 24 GB RAM each.

5. Experiments

Benchmark datasets for complex networks were used to perform experiments that labeled each network in the dataset. A bLSTM network was tuned for each dataset, but the architecture is identical for all bLSTM. Table 1 shows how to classify synthetic complex networks by identifying the model used to generate them. The text mentions four models of networks: Barabasi-Albert (BA) [27] for self-similar networks, Song, Halvin, and Makse (SHM) [33] for fractals, Watts and Strogatz (WS) [26] for networks that have the small-world property, and (u,v)-flowers (UVF) [34] for scale-free deterministic networks.

The experiments were conducted using 10-fold cross-validation, and the performance was averaged across the 10-fold. Due to the combination of real-world and synthetic datasets, which results in unbalanced classes, the reported performance measures include Matthew's correlation coefficient (MCC) and the Area Under the Receiver Operating Characteristics Curve (AUC), in addition to accuracy (ACC).

Table 1: Features of synthetic networks are grouped by the network. The average number of Edges and Nodes is presented in the respective columns.

Dataset	Model	# Complex Networks	Edges	Nodes
BA	Barabasi-Albert[27]	255	12,790	3,400
SHM	Song, Halvin and Makse [33]	218	54,023	2,568.8
UVF	(u,v)-flower [34]	255	3,093.8	2,678.4
WS	Watts and Strogatz [26]	216	18,750	3,125

5.1. Real and Synthetic networks

Our experiments aimed to categorize real networks in various domains, including the brain, food, infrastructural, Cheminformatics, PPI, and Social sectors. We classified a given network according to its corresponding ground truth domain. Then, the synthetic networks were categorized based on the true model that created them. Next, the real-world and synthetic networks were combined into a dataset, and classification was performed to determine their respective domains or models. No data augmentation was applied during these experiments. Table 2 summarises these experiments AUC, MCC, and ACC using De-bLSTM and classification applying the AdaBoosty and random tree algorithm (De-AdaBoost), which performed the best among many other algorithms. We have also studied whether the bLSTM algorithm performs the best on the Deng entropy-based embedding or any other advanced machine learning algorithm that could achieve such high performance using this network representation.

Table 2: ACC, UAC, and MCC scores of the De-bLSTM and De-AaBoost algorithms are compared in classifying real-world, synthetic, and combined datasets. Standard deviation values are also included in parentheses.

Algorithm	Real-world (6 classes)			Synthetic (4 Classes)			Real-world and Synthetic (10 classes)		
	AUC	MCC	ACC	AUC	MCC	ACC	AUC	MCC	ACC
De-bLSTM	0.99 (0.01)	0.95 (0.04)	97.24 (2.50)	1.00 (0.00)	1.00 (0.00)	100 (0.00)	0.99 (0.01)	0.98 (0.01)	97.07 (2.09)
De-AdaBoost	0.98 (0.01)	0.83 (0.06)	91.55 (3.34)	1.00 (0.00)	0.99 (0.01)	97.73 (1.86)	0.99 (0.00)	0.84 (0.05)	99.77 (1.39)

As shown in Table 2, although the De-AdaBoost methods demonstrate remarkable performance, the De-bLSTM outperforms in all classification tasks. When unbalanced, the De-bLSTM method achieves high accuracy, such as in a real-world dataset. When the synthetic and real-world datasets are brought together, something similar happens.

6. Conclusion

This study proposes a new method for classifying complex networks using the entropy of nodes and edges in a network called DE-bLSTM. The approach combines two concepts, namely Deng entropy and bLSTM networks, to achieve this goal. Additionally, the De-bLSTM approach was tested on both synthetic and real-world complex networks and proved effective. The De-bLSTM demonstrated high accuracy in classifying different networks belonging to various domains. Finally, a statistical test was performed on 10-fold cross-validation results to determine if there was a significant difference between our approach and the best GNN for a given dataset.

The De-bLSTM method has shown remarkable accuracy in classifying problems that involve two or more classes. Therefore, De-bLSTM can be utilized in domains that possess these features. However, one limitation of De-bLSTM is that it does not take into account the labels of the nodes or edges, which could potentially enhance the classification performance. Ongoing research aims to extend the algorithm by including these labels in the data structure to train the bLSTM. Additionally, classifying nodes and dynamic complex networks will be a future extension of De-bLSTM.

Acknowledgments

This research was partially supported by Secretaría de Investigación y Posgrado, Instituto Politécnico Nacional (Grant: 20240278). The work of María del Carmen Soto Camacho has been supported by the Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCYT, 651b2ff07392111317337653).

References

[1] Smalter, Aaron, Huan, Jun, Jia, Yi, and Lushington, Gerald, "GPD: A Graph Pattern Diffusion Kernel for Accurate Graph Classification with Applications in Cheminformatics," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 7, no. 2, pp. 197–207, 2010, doi: 10.1109/TCBB.2009.80.

[2] Han, Donghong, Hu, Yachao, Ai, Shuangshuang, and Wang, Guoren, "Uncertain graph classification based on extreme learning machine," *Springer*, vol. 7, pp. 346--358, 2015.

[3] Xianglin Zuo, Hao Yuan, Bo Yang, Hongji Wang, and Ying Wang, "Exploring graph capsual network and graphormer for graph classification," *Information Sciences*, vol. 640, p. 119045, 2023, doi: https://doi.org/10.1016/j.ins.2023.119045.

[4] Yongjiao Sun, Boyang Li, Ye Yuan, Xin Bi, Xiangguo Zhao, and Guoren Wang, "Big graph classification frameworks based on Extreme Learning Machine," *Neurocomputing*, vol. 330, pp. 317–327, 2019, doi: https://doi.org/10.1016/j.neucom.2018.11.035.
[5] Mansurul Bhuiyan and Mohammad Al Hasan, "Representing Graphs as Bag of Vertices and Partitions for Graph Classification," *Data Science and Engineering*, vol. 3, 2018, doi: 10.1007/s41019-018-0065-5.
[6] Nagy, Marcell and Molontay, Roland, "Network classification-based structural analysis of real networks and their model-generated counterparts," *Network Science*, vol. 10, no. 2, pp. 146–169, 2022, doi: 10.1017/nws.2022.14.
[7] Michel Neuhaus, Horst Bunke, and Kaspar Riesen, "Novel kernels for error-tolerant graph classification," *Brill*, vol. 22, pp. 425–441, 2009, doi: 10.1163/156856809789476119.
[8] Riesen, Kaspar and Bunke, Horst, "Graph Classification by Means of Lipschitz Embedding," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, pp. 1472–1483, 2009, doi: 10.1109/TSMCB.2009.2019264.
[9] Tinghuai Ma, Wenye Shao, Yongsheng Hao, and Jie Cao, "Graph classification based on graph set reconstruction and graph kernel feature reduction," *Neurocomputing*, vol. 296, pp. 33–45, 2018, doi: https://doi.org/10.1016/j.neucom.2018.03.029.
[10] Kriege, Nils M, Johansson, Fredrik D, and Morris, Christopher, "A survey on graph kernels," *SpringerOpen*, vol. 5, pp. 1--42, 2020.
[11] Bai, Lu, Jiao, Yuhang, Cui, Lixin, and Hancock, Edwin R, "Learning aligned-spatial graph convolutional networks for graph classification".
[12] Bouritsas, Giorgos, Frasca, Fabrizio, Zafeiriou, Stefanos, and Bronstein, Michael M., "Improving Graph Neural Network Expressivity via Subgraph Isomorphism Counting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, pp. 657–668, 2023, doi: 10.1109/TPAMI.2022.3154319.
[13] Wei Dong, Junsheng Wu, Zongwen Bai, Weigang Li, and Wei Qiao, "Design of affinity-aware encoding by embedding graph centrality for graph classification," *Neurocomputing*, vol. 387, pp. 321–333, 2020, doi: https://doi.org/10.1016/j.neucom.2020.01.010.
[14] Tinghuai Ma, Qian Pan, Hongmei Wang, Wenye Shao, Yuan Tian, and Najla Al-Nabhan, "Graph classification algorithm based on graph structure embedding," *Expert Systems with Applications*, vol. 161, p. 113715, 2020, doi: https://doi.org/10.1016/j.eswa.2020.113715.
[15] Yu Xie, Chuanyu Yao, Maoguo Gong, Cheng Chen, and A.K. Qin, "Graph convolutional networks with multi-level coarsening for graph classification," *Knowledge-Based Systems*, vol. 194, p. 105578, 2020, doi: https://doi.org/10.1016/j.knosys.2020.105578.
[16] Wang, Shui-Hua, Zhou, Jin, and Zhang, Yu-Dong, "Community-acquired pneumonia recognition by wavelet entropy and cat swarm optimization," *Springer*, pp. 1--18, 2022.
[17] Gao, Jianliang, Gao, Jun, Ying, Xiaoting, Lu, Mingming, and Wang, Jianxin, "Higher-Order Interaction Goes Neural: A Substructure Assembling Graph Attention Network for Graph Classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, pp. 1594–1608, 2023, doi: 10.1109/TKDE.2021.3105544.
[18] Jinyin Chen, Haiyang Xiong, Haibin Zheng, Dunjie Zhang, Jian Zhang, Mingwei Jia, Yi Liu, "EGC2: Enhanced graph classification with easy graph compression," *Information Sciences*, vol. 629, pp. 376–397, 2023, doi: https://doi.org/10.1016/j.ins.2023.01.114.
[19] Do, Manh Tuan, Park, Noseong, and Shin, Kijung, "Two-stage training of graph neural networks for graph classification," *Springer*, vol. 55, pp. 2799--2823, 2023.
[20] Mingming Lu, Zhixiang Xiao, Haifeng Li, Ya Zhang, and Neal N. Xiong, "Feature pyramid-based graph convolutional neural network for graph classification," *Journal of Systems Architecture*, vol. 128, p. 102562, 2022, doi: https://doi.org/10.1016/j.sysarc.2022.102562.
[21] Ye, Yang and Ji, Shihao, "Sparse Graph Attention Networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, pp. 905–916, 2023, doi: 10.1109/TKDE.2021.3072345.

[22] Wei, Lanning, Zhao, Huan, He, Zhiqiang, and Yao, Quanming, "Neural Architecture Search for GNN-Based Graph Classification," *Assoc. Comput. Mach.*, vol. 42, no. January 2024, p. 29, 2023, doi: 10.1145/3584945.
[23] Yang, Yukun, Ma, Bo, Liu, Xiangdong, Zhao, Liang, and Huang, Shoudong, "GSAP: A Global Structure Attention Pooling Method for Graph-Based Visual Place Recognition," *Remote Sensing*, vol. 13, no. 8, 2021, doi: 10.3390/rs13081467.
[24] Ying Wang, Hongji Wang, Hui Jin, Xinrui Huang, and Xin Wang, "Exploring graph capsual network for graph classification," *Information Sciences*, vol. 581, pp. 932–950, 2021, doi: https://doi.org/10.1016/j.ins.2021.10.001.
[25] Ji Gan, Yuyan Chen, Bo Hu, Jiaxu Leng, Weiqiang Wang, and Xinbo Gao, "Characters as graphs: Interpretable handwritten Chinese character recognition via Pyramid Graph Transformer," *Pattern Recognition*, vol. 137, 2023, doi: 10.1016/j.patcog.2023.109317.
[26] Watts, Duncan J. and Strogatz, Steven H., "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, p. 440, 1998.
[27] Barabási, A. L. and Albert, R., "Emergence of scaling in random networks," *Science*, vol. 286, pp. 509–512, 1999.
[28] Song, Chaoming, Havlin, Shlomo, and Makse, Hernan, "Self-similarity of complex networks," *Nature*, vol. 433, p. 392, 2005.
[29] Aldo Ramirez-Arellano, "Students learning pathways in higher blended education: An analysis of complex networks perspective," *Computers \& Education*, vol. 141, p. 103634, 2019.
[30] A. P. Dempster, "Upper and Lower Probabilities Induced by a Multivalued Mapping," *Inst. Math. Stat.*, vol. 38, pp. 325-- 339, 1967.
[31] Yong Deng, "Deng entropy," *Chaos, Solitons \& Fractals*, vol. 91, pp. 549–553, 2016, doi: 10.1016/j.chaos.2016.07.014.
[32] Deng, Yong, "Information volume of mass function," *International Journal of Computers Communications \& Control*, vol. 15, no. 6, 2020.
[33] Song, Chaoming, Havlin, Shlomo, and Makse, Hernán, "Origins of fractality in the growth of complex networks," *Nature physics*, vol. 2, no. 4, p. 275, 2006.
[34] Hernán D Rozenfeld, Shlomo Havlin, and Daniel ben-Avraham, "Fractal and transfractal recursive scale-free nets," *New Journal of Physics*, vol. 9, p. 175, 2007, doi: 10.1088/1367-2630/9/6/175.