Deep Attentional Implanted Graph Clustering Algorithm for the Visualization and Analysis of Social Networks

Dr. Fernando Escobedo[1*], Dr. Henry Bernardo Garay Canales[2],
Dr. Eddy Miguel Aguirre Reyes[3], Carlos Alberto Lamadrid Vela[4],
Oscar Napoleón Montoya Perez[5] and Grover Enrique Caballero Jimenez[6]

[1*]Professor, Universidad César Vallejo, Tarapoto, Perú. jescobedog@ucv.edu.pe,
https://orcid.org/0000-0002-6443-1497

[2]Professor, Universidad Nacional de Tumbes, Tumbes, Perú. hgarayc@untumbes.edu.pe,
https://orcid.org/0000-0003-2323-1103

[3]Professor, Universidad Nacional de Tumbes, Tumbes, Perú. eaguirrer@untumbes.edu.pe,
https://orcid.org/0000-0003-1304-7601

[4]Professor, Universidad Nacional de Tumbes, Tumbes, Perú. colono222@htomail.com,
https://orcid.org/0000-0003-4011-3301

[5]Professor, Universidad Nacional de Tumbes, Tumbes, Perú. napo.mp@gmail.com,
https://orcid.org/0000-0001-6464-8189

[6]Professor, Universidad Nacional de Tumbes, Tumbes, Perú. enriqgcj5@hotmail.com,
https://orcid.org/0009-0009-7317-5729

Received: October 20, 2023; Accepted: December 24, 2023; Published: February 29, 2024

Abstract

As the user base expands, social network data becomes more intricate, making analyzing the interconnections between various entities challenging. Various graph visualization technologies are employed to analyze extensive and intricate network data. Network graphs inherently possess intricacy and may have overlapping elements. Graph clustering is a basic endeavor that aims to identify communities or groupings inside networks. Recent research has mostly concentrated on developing deep learning techniques to acquire a concise representation of graphs, which is then utilized with traditional clustering methods such as k-means or spectral clustering techniques. Multiplying these two-step architectures is challenging and sometimes results in unsatisfactory performance. This is mostly due to the lack of a goal-oriented graph encoding developed explicitly for the clustering job. This work introduces a novel Deep Learning (DL) method called Deep Attentional Implanted Graph Clustering (DAIGC), designed to achieve goal-oriented clustering. Our approach centers on associated graphs to thoroughly investigate both aspects of data in graphs. The proposed DAIGC technique utilizes a Graph Attention Autoencoder (GAA) to determine the significance of nearby nodes about a target node. This allows encoding a graph's topographical structure and node value into a concise representation. Based on this representation, an interior product decoder has been trained to rebuild the graph structure. The performance of the proposed approach has been evaluated on four distinct types and sizes of real-world intricate networks, varying in vertex count from $N = 10^2\ to\ N = 10^7$. The performance of the suggested

Journal of Internet Services and Information Security (JISIS), volume: 14, number: 1 (February), pp. 153-164.
DOI: *10.58346/JISIS.2024.I1.010*

*Corresponding author: Professor, Universidad César Vallejo, Tarapoto, Perú.

methods is evaluated by comparing them with two established and commonly used graph clustering techniques. The testing findings demonstrate the effectiveness of the proposed method in terms of processing speed and visualization compared to the state-of-the-art algorithms.

Keywords: Goal-oriented Graph Encoding, Clustering, Social Networks, Visualization, Analysis, Deep Learning.

1 Introduction to Graph Clustering

The presence of a social structure is seen as a crucial characteristic of complicated networks found in real-world scenarios. Key topological characteristics of sophisticated networks are a power-law dispersion of node degrees, a high clustering factor, and a reduced separation degree. Due to their topological properties, nearly all complicated networks in the actual world exhibit a mesoscopic level of connections known as communities (Sun et al., 2020). The society is an associated subgraph in which most internal nodes are more strongly connected than external ones. Within neighborhoods, nodes are expected to possess shared traits or fulfill comparable jobs within a network. By recognizing and evaluating the structure of the underpinning communities, one can get valuable insights into the intricate network systems (Jin et al., 2021).

Link weights (w) are commonly used in real-world networks to quantify the proximity, strength, or capability between nodes. Networks that have linkages with assigned weights are referred to as weighted networks. The weights of links play a crucial role in determining the operational and structural characteristics of complex networks found in the real world. Within social networks, nodes symbolize individuals, while the links connecting these nodes symbolize their interpersonal connections (Gong et al., 2020). Link weights increase as the strength of the relationship between persons increases and decreases as the strength of the relationship falls. (Kolomeets et al., 2021).

Conversely, in non-social networks, link weights represent connections' specific roles. For instance, connection weights in a food web network can indicate the varying power levels or carbon circulation between predator-prey groups. In metabolic networks, link weights can reflect the differing amounts of fluid along specific reaction routes. In transit networks, link weights can represent the traffic moving along connections. Lastly, in neural networks, link weights can signify the number of organic processes and gap junctions involved (Knyazeva & Talalaeva 2021).

Graph clustering seeks to divide the nodes in the graph into separate and non-overlapping clusters. Common uses of this technology are identifying communities, segmenting groups, and discovering functional groups within workplace social networks (Yang et al., 2022). For associated graph clustering, a crucial issue is how to effectively retain the structural link and utilize the data contained in the nodes.

To address this issue, recent research has employed deep learning methods to acquire concise representations that effectively utilize the abundant information in the material and structural data (Wu et al., 2020). Utilizing the acquired graph embedding, basic clustering methods like k-means are employed. The autoencoder is a widely used option for embedding-based approaches (Dehmamy et al., 2019). It is particularly useful in unsupervised contexts because it can learn hidden representations.

However, it is important to note that all these methods that rely on embeddings follow a two-step process. An issue arises when the acquired embedding may not be the most suitable for the succeeding graph clustering job, and the graph clustering task does not provide any advantages to the graph embedding learning process. A goal-oriented training framework is widely sought to attain reciprocal advantages for these two stages. However, conventional goal-oriented training methods are mostly utilized for classification tasks—for example, authors in (Ding et al., 2021) introduced graph convolutional networks for interconnected data. As far as we know, a limited number of works on goal-oriented embedding approaches for graph clustering exist. In light of these facts, we present a methodology in this research that utilizes a goal-oriented Graph Attentional Autoencoder (GAA) to perform related graph clustering.

2 Related Works on Graph Clustering, Graph Autoencoders, and Graph Clustering Attention Methods

This section introduces graph clustering, autoencoders, and standard graph clustering attention methods.

2.1. Graph Clustering

This process groups nodes in a graph based on similarities or connections. The Gaussian Pigeon-Oriented Graph Clustering (GPOGC) method by Sun and colleagues is designed for social networks (Sun et al., 2019). The Gaussian pigeon-inspired method maximizes social network graph clustering. This is implemented in real-world social networks. The algorithm improves clustering performance, as shown by cluster quality metrics like modularity and silhouette score. This method improves social network clustering, especially with complex interactions. Limitations include being sensitive to parameter adjustment and not applying to all graph architectures.

The Hard Sample Aware Network for Contrastive Deep Graph Clustering was introduced by Liu and colleagues (Liu et al., 2023). The implementation involves building a graph data network sensitive to difficult samples. In contrastive deep graph clustering, the output shows improved clustering accuracy and the model's excellent handling of difficult data. Benefits include improved efficiency in complex graph topologies. Higher computing complexity and dataset peculiarities may be drawbacks.

Liu and colleagues propose Simple Contrastive Graph Clustering for graph clustering. The method involves creating a simple but effective contrastive learning framework for graph clustering (Liu et al., 2023). Results show improved clustering accuracy and the technique's simplicity and efficacy. Benefits include direct and effective graph clustering. Complex network topologies may present management challenges.

Müller uses graph neural networks to cluster graphs. Neural networks are used to learn graph node knowledge representations and cluster them. The output shows improved clustering results, including purity and normalized mutual information (Müller 2023). Neural networks can record complex graph topologies. Hyperparameter adjustment and high computing requirements are drawbacks. These sources show how to cluster graphs in social networks, deep graph clustering, and using Gaussian-inspired optimization and Graph Neural Networks. Each method has advantages and drawbacks depending on the graph data and clustering goal.

2.2. Graph Autoencoder

Liu and colleagues use Graph Autoencoders to hide social network communities. A graph autoencoder is trained to create concise network representations while preserving community structures (Liu et al., 2022). The output shows how to hide communities in graph representations, and the values may include metrics for community concealment. Community structure maintenance is efficient with graph autoencoders. Fine-tuning hyperparameters and managing large networks may be difficult.

Liu and colleagues present Siga, a new social network rating prediction method that combines Social Influence Modeling and Graph Autoencoder (Liu et al., 2023). The process uses graph autoencoders to model social impact patterns. The application estimates ratings using the Siga algorithm on real-world datasets. Forecasting ratings are more accurate, and the values include MSE and RMSE. Social impact modeling improves rating prediction accuracy. It may be susceptible to noisy data and require careful parameter tuning.

Yang and colleagues propose using Multi-Modal Graph Edge Variational Autoencoders to learn about social network relationships (Yang et al., 2020). Integration of variational autoencoders into a multi-modal framework captures complex social network relationships. The output shows improved relationship learning, and the values may include metrics to assess relationship quality. Representing and analyzing social network relationships with multiple modes or types is an advantage. Complex computations and relationship management may be drawbacks.

These references show how graph autoencoders can conceal communities, predict ratings, and learn relationships in social networks. Each method improves representation learning and prediction precision, but drawbacks vary by job and dataset.

2.3. Attention Mechanisms

Human eyesight has a unique signal-processing system called the attention mechanism. The human visual system rapidly scans the image, known as the focus of attention, to find the region of interest and eliminate irrelevant information. Attention models, first used in machine translation, are now used in natural language processing, picture recognition, and recommender systems. Visual attention processes can quickly identify Image regions of interest (Fu et al., 2017).

Graph Attention Networks (GAN) (Casanova et al., 2018) utilize the self-attention process to aggregate information from neighboring nodes in graph clustering. This enhances the capacity to represent node attributes and enables adaptive weighting of various neighbors. Graph Convolutional Network with Attention Graph Clustering (GCAGC) is an adaptive graph convolution network incorporating attention graph clustering (Zhang et al., 2020). The Attention-driven Graph Clustering Network (ADGCN) (Peng et al., 2021) is an innovative deep clustering technique using a heterogeneous fusion module to combine node characteristics and topological information dynamically. Additionally, it utilizes the attention mechanism to merge matching features dynamically.

3 Deep Attentional Implanted Graph Clustering (DAIGC) to Achieve Goal-oriented Clustering

This paper introduces a novel framework called Deep Attentional Implanted Graph Clustering (DAIGC) for achieving goal-oriented clustering. A GAA capable of learning latent representations has been created to leverage the interconnectedness of different graph data types. The encoder utilizes a GAN to leverage the graph's layout and the information that resides in the nodes. Several encoders are stacked to construct a deep architecture for learning embeddings. The decoder, located on the opposite end, reconstructs the topographical graph data and changes the implicit graph model. In addition, we utilize a self-training module that uses the "assured" clustering allocations as soft tags to direct the optimization process. This focused clustering component combines implanting training and clustering into a single framework, enhancing clustering efficiency by optimizing the clustering distribution to match an improbable improved distribution. This approach differs from the two-step implanting learning-based methods illustrated in Figure 1.

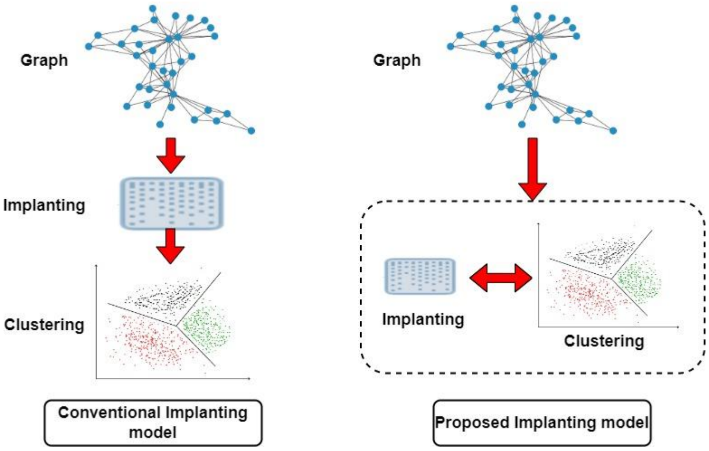

Figure 1: Conventional two-step implanting learning-based methods and the proposed implanting model

This paper focuses on the clustering task applied to attributed graphs. A graph is denoted as $G = (W;\ E;\ Y)$, where $W = \{w_j\}, j = 1,2, \ldots\ldots n$ represents a collection of nodes and $W = \{w_{ji}\}$ represents a collection of edges connecting the nodes. The graph G's topographical structure can be depicted using an adjacency matrix B, where $B_{j,i} = 1,$ if $[w_j, w_i] \in E$; otherwise $B_{j,i} = 0$. The attribute values $Y = \{y_1; y_2 \ldots\ldots y_n\}$ consist of real-value attribute vectors $y_i \in S^m$ associated with vertex w_j.

Graph clustering is the process of dividing the nodes in graph G into l separate groups, denoted as $\{G_1, G_2, \ldots\ldots, G_l\}$. The objective is to ensure that nodes within a particular cluster are usually adjacent in graph design but distant from nodes in other clusters. Additionally, nodes within a similar cluster are more likely to have identical attribute values. The framework depicted in Figure. 2 comprises a GAA and a self-learning clustering unit.

GAA utilizes a graph attention mechanism to process the values of attributes and the graph's layout as input. It then learns the implicit embedding by reducing the restoration loss.

The self-learning unit conducts clustering using the acquired illustration and adjusts the implicit representation based on the present clustering outcome.

DAIGC simultaneously learns the graph implanting and conducts clustering within a cohesive framework, allowing each element to enhance the other.

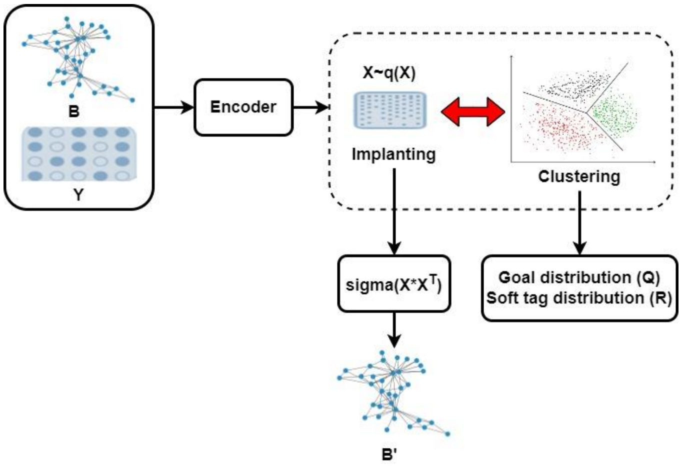

Figure 2: Proposed DAIGC framework

The theoretical structure of DAIGC has been depicted in Figure 2. DAIGC is a method that utilizes a GAA to learn a hidden depiction X from a graph $G = (W;\ E;\ Y)$. It then changes this representation using a self-learning clustering unit, which is optimized alongside the autoencoder and performs clustering during learning.

3.1. GAA

The GAA model integrates graph structure and attribute information to cluster attributed graphs effectively. The encoder captures the attention weights for neighboring nodes and aggregates their attribute information:

$$h_j = sigma(\sum_{i=1}^{n} softmax\left(\alpha_{ji}\right). W_1 . y_i) \tag{1}$$

Attention coefficients $\left(\alpha_{ji}\right)$ are computed using a shared self-attention mechanism:

$$\alpha_{ji} = \frac{exp(LeakyReLU(\left[a^T . \left[W_2 . y_j \middle| |W_3 . y_i\right]\right]))}{\sum_{l=1}^{n} exp(LeakyReLU([\left[a^T . \left[W_2 . y_j \middle| |W_3 . y_l\right]\right]))} \tag{2}$$

|| depicts the concatenation, W_1, W_2, W_3 are the learning weight matrices and a^T is the attention parameter. The decoding of GAA involves the reconstruction of the input graph using the trained node implanting and is given as:

$$\hat{Y} = \sigma(W_4 . h_j) \tag{3}$$

h_j is the trained implanting that is utilized for clustering. The attention technique included in the GAA enables the model to concentrate on pertinent surrounding nodes throughout the encoding process. This ensures that the acquired embeddings effectively capture the graph's structure and attribute information. The decoder reconstructs the graph using these embeddings, while the clustering unit utilizes them to create significant clusters. This integration enables efficient clustering in graphs with attributes.

The encoding process of the GAA captures the attention weights of adjacent nodes. It combines their attribute information, enabling a thorough representation of the graph's structure and the nodes' properties. The attention mechanism is characterized by attention coefficients, denoted as α_{ji}, which are calculated via a shared self-attention process. This technique includes a weight matrix that can be learned (denoted as W_1) and a shared attention parameter (denoted as a^T). These components enable the model to concentrate on pertinent information when encoding.

The decoding procedure uses the acquired node embeddings to recreate the input graph. The attribute values that have been rebuilt ($\hat{Y}$) are acquired by utilizing a decoding layer (W_4) and a sigmoid activation function. The acquired embedding (h_j), functions as a comprehensive depiction of network nodes, encompassing structural and attribute-based data.

Pseudocode 1: Proposed DAIGC to achieve goal-oriented clustering

```
# Graph Clustering
Number of clusters: G = {G1, G2, ..., Gl}
# Components
DAIGC: Integrate GAA and Self-learning Clustering Unit
# Algorithm
def graph_attentional_autoencoder(G, l):
 # Encoding Phase
 W1, W2, W3, W4, a = initialize_weights()
 H = graph_attention_encoding(G, W1, W2, a)
 # Decoding Phase
 Y_hat = graph_attention_decoding(H, W3, W4)
 # Self-learning Clustering
 clusters = self_learning_clustering(H, l); return Y_hat, clusters
# Graph Attention Encoding
def graph_attention_encoding(G, W1, W2, a)
 # Compute attention coefficients
 α = calculate_attention_coefficients(G, W1, a)
 # Aggregate neighboring nodes' information
 H = aggregate_information(G, α, W2; return H
# Graph Attention Decoding
def graph_attention_decoding(H, W3, W4):
 # Reconstruct attribute values
 Y_hat = reconstruct_attributes(H, W3, W4) ; return Y_hat
# Self-learning Clustering
def self_learning_clustering(H, l):
# Perform clustering using learned embeddings
clusters = k_means_clustering(H, l); return clusters
# Helper Functions
def initialize_weights():
 # Initialize learnable weight matrices
 W1, W2, W3, W4 = initialize_parameters()
 # Shared attention parameter
 a = initialize_attention_parameter(); return W1, W2, W3, W4, a
def calculate_attention_coefficients(G, W1, a):
# Calculate attention coefficients using a shared self-attention mechanism
α = attention_mechanism(G, W1, a); return α
def aggregate_information(G, α, W2):
# Aggregate neighboring nodes' information using attention coefficients
 H = aggregate_with_attention(G, α, W2); return H
def reconstruct_attributes(H, W3, W4):
 # Reconstruct attribute values
 Y_hat = reconstruct(H, W3, W4); return Y_hat
def k_means_clustering(H, l):
 # Perform K-means clustering
clusters = k_means(H, l); return clusters
```

The self-learning clustering unit employs the acquired embeddings to cluster the attributed graph, creating significant clusters. The effectiveness of the DAIGC in improving clustering performance in attributed graphs is demonstrated by the integrated method (GAA+ Self-learning Clustering Unit) seen in Pseudocode 1. This technique combines graph structure with attribute information. The suggested model is a potential approach to enhance clustering tasks where graph topology and node properties are significant factors.

4 Results and Discussion

To assess the effectiveness of the suggested community discovery method, we conducted tests on several types and scales of real-world complex networks. We have chosen varying-sized networks, ranging from 30 to 1,40,6135 nodes. Before conducting tests, we transformed all unweighted networks into weighted networks by allocating a weight of 1 to all edges.

Table 1: The database used in the experiment

Database	**Nodes**	**Edges**	**Description**
Facebook	900	73295	This network is specifically designed to exchange confidential messages among its users.
Social Networks	1901	19986	This social networking platform, similar to Facebook, originated from a community of users.
Elegans	305	2458	The dataset comprises neural network data.
Bn-fly	1805	32587	The dataset comprises brain networks.

The experiments were conducted on a standalone computer machine equipped with a 2.5GHz quad-core CPU, 10MB cache, 128GB RAM, and running the Microsoft Windows operating system. The database used in the experiment is shown in Table 1.

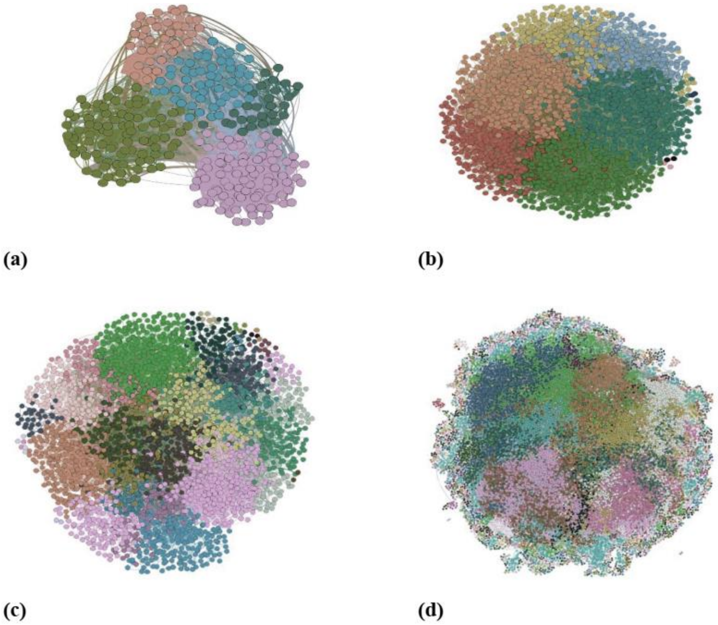

Figure 3: Visual representations of communities in various networks. (a) Elegans, (b) Social Networks, (c) Bn-fly, and (d) Facebook

Figure 3 displays visual representations of communities in various networks using our suggested methodology. To differentiate the communities, distinct colors have been assigned to them throughout the process of producing their representations. The visualization results demonstrate that the proposed method successfully identified communities of enough size in varying-sized networks. Greater community size correlates directly with less visual complexity.

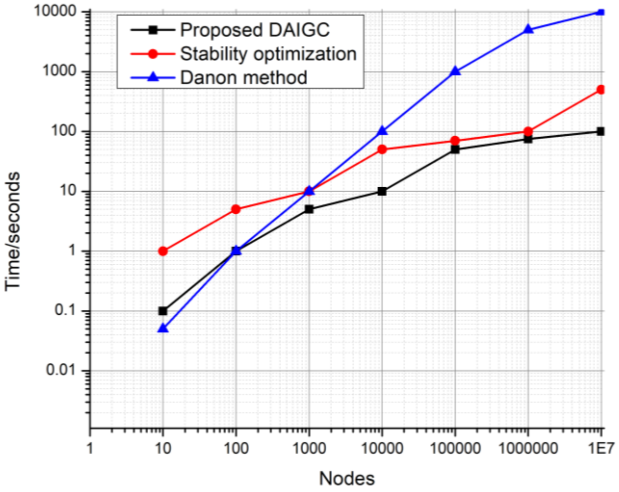

Figure 4: Performance comparison of various algorithms regarding processing time (s)

Figure 4 depicts a performance comparison among three algorithms: the proposed DAIGC, Stability Optimization (Mavroeidis & Marchiori 2014), and the Danon approach (Danon et al., 2006). The comparison is based on the processing time (in seconds) for different numbers of nodes. With increasing nodes, the proposed DAIGC constantly surpasses previous approaches, demonstrating its superior efficiency in managing larger-scale networks. With a network size of 1 million nodes, the proposed DAIGC algorithm takes 75 seconds to complete. In comparison, the Stability Optimization algorithm takes 100 seconds, while the Danon approach takes 500 seconds. This suggests that the proposed DAIGC has greater processing efficiency, making it an attractive option for situations involving large node populations.

5 Conclusion

This study presents a new DL technique named Deep Attentional Implanted Graph Clustering (DAIGC), which aims to accomplish goal-oriented clustering. We focus on utilizing connected graphs to analyze comprehensively both aspects of data presented in graphs. The proposed DAIGC approach employs a Graph Attention Autoencoder (GAA) to ascertain the importance of neighboring nodes concerning a target node. This enables encoding a graph's topographical structure and node value into a brief form. A trained interior product decoder can reconstruct the graph structure using this format. The efficacy of the suggested methodology has been assessed on four different categories and magnitudes of complex real-world networks, ranging in vertex quantity from $N=10^2$ to $N=10^7$. The efficacy of the proposed approaches is assessed by contrasting them with two well-established and widely employed graph clustering techniques. With a network size of 1 million nodes, the proposed DAIGC algorithm takes 75 seconds to complete.

In comparison, the Stability Optimization algorithm takes 100 seconds, while the Danon approach takes 500 seconds. This suggests that the proposed DAIGC has greater processing efficiency, making it an attractive option for situations involving large node populations. The testing results illustrate the efficacy of the suggested approach in terms of its processing speed and visualization capabilities when compared to the most advanced algorithms currently available.

References

[1] Casanova, P.V.G.C.A., Lio, A.R.P., & Bengio, Y. (2018). Graph attention networks. *ICLR. Petar Velickovic Guillem Cucurull Arantxa Casanova Adriana Romero Pietro Liò and Yoshua Bengio*.

[2] Danon, L., Diaz-Guilera, A., & Arenas, A. (2006). The effect of size heterogeneity on community identification in complex networks. *Journal of Statistical Mechanics: Theory and Experiment*, *2006*(11). https://doi.org/10.1088/1742-5468/2006/11/P11010

[3] Dehmamy, N., Barabási, A.L., & Yu, R. (2019). Understanding the representation power of graph neural networks in learning graph topology. *Advances in Neural Information Processing Systems*, *32*.

[4] Ding, Y., Tian, L.P., Lei, X., Liao, B., & Wu, F.X. (2021). Variational graph auto-encoders for miRNA-disease association prediction. *Methods*, *192*, 25-34.

[5] Fu, J., Zheng, H., & Mei, T. (2017). Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 4438-4446.

[6] Gong, Z., Wang, H., Guo, W., Gong, Z., & Wei, G. (2020). Measuring trust in social networks based on linear uncertainty theory. *Information Sciences*, *508*, 154-172.

[7] Jin, D., Yu, Z., Jiao, P., Pan, S., He, D., Wu, J., & Zhang, W. (2021). A survey of community detection approaches: From statistical modeling to deep learning. *IEEE Transactions on Knowledge and Data Engineering*, *35*(2), 1149-1170.

[8] Knyazeva, I., & Talalaeva, O. (2021). Topological data analysis approach for weighted networks embedding. In *Networks in the Global World V: Proceedings of NetGloW 2020,* Springer International Publishing, 81-100.

[9] Kolomeets, M., Chechulin, A., & Kotenko, I.V. (2021). Bot detection by friends graph in social networks. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 12*(2), 141-159.

[10] Liu, D., Chang, Z., Yang, G., & Chen, E. (2022). Community hiding using a graph autoencoder. *Knowledge-Based Systems*, *253*. https://doi.org/10.1016/j.knosys.2022.109495.

[11] Liu, J., Xiao, Y., Zheng, W., & Hsu, C.H. (2023). Siga: social influence modeling integrating graph autoencoder for rating prediction. *Applied Intelligence*, *53*(6), 6432-6447.

[12] Liu, Y., Yang, X., Zhou, S., Liu, X., Wang, S., Liang, K., & Li, L. (2023). Simple contrastive graph clustering. *IEEE Transactions on Neural Networks and Learning Systems*, 1-12.

[13] Liu, Y., Yang, X., Zhou, S., Liu, X., Wang, Z., Liang, K., & Chen, C. (2023). Hard sample aware network for contrastive deep graph clustering. *In Proceedings of the AAAI conference on artificial intelligence*, *37*(7), 8914-8922.

[14] Mavroeidis, D., & Marchiori, E. (2014). Feature selection for k-means clustering stability: theoretical analysis and an algorithm. *Data Mining and Knowledge Discovery*, *28*, 918-960.

[15] Müller, E. (2023). Graph clustering with graph neural networks. *Journal of Machine Learning Research*, *24*, 1-21.

[16] Peng, Z., Liu, H., Jia, Y., & Hou, J. (2021). Attention-driven graph clustering network. *In Proceedings of the 29th ACM international conference on multimedia*, 935-943.

[17] Sun, H., Du, H., Huang, J., Li, Y., Sun, Z., He, L., & Zhao, Z. (2020). Leader-aware community detection in complex networks. *Knowledge and Information Systems*, *62*, 639-668.

[18] Sun, Y., Yin, S., Li, H., Teng, L., & Karim, S. (2019). GPOGC: Gaussian pigeon-oriented graph clustering algorithm for social networks cluster. *IEEE Access*, *7*, 99254-99262.

[19] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Philip, S.Y. (2020). A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, *32*(1), 4-24.

[20] Yang, C., Zhang, J., Wang, H., Li, S., Kim, M., Walker, M., & Han, J. (2020). Relation learning on social networks with multi-modal graph edge variational autoencoders. *In Proceedings of the 13th International Conference on Web Search and Data Mining*, 699-707.

[21] Yang, T., Yang, C., Zhang, L., Shi, C., Hu, M., Liu, H., & Wang, D. (2022). Co-clustering interactions via attentive hypergraph neural network. *In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 859-869.

[22] Zhang, K., Li, T., Shen, S., Liu, B., Chen, J., & Liu, Q. (2020). Adaptive graph convolutional network with attention graph clustering for co-saliency detection. *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9050-9059.

Authors Details

Dr. Fernando Escobedo, Doctor in Public Management and Governance, Master in Public Management, Bachelor in Administration, RENACYT Research Professor, Coordinator of Research, Development and Innovation, Head of Research of the Professional School of Administration, with proactive capacity and initiative linked to the achievement of objectives.

Dr. Henry Bernardo Garay Canales, Doctor in education, master in public management, master in education, chartered public accountant, university educator, undergraduate and graduate specialist in public management, thesis advisor and speaker at academic events. academic events.

Dr. Eddy Miguel Aguirre Reyes, Doctor in Education Administration, Master in Education with mention in Teaching and Educational Management and Master in Public Management, Chartered Public Accountant and regular teacher of the National University of Tumbes. Professional in constant training, which allows the fulfillment of objectives and adapt to the dizzying changes that are the result of globalization.

Carlos Alberto Lamadrid Vela, Doctor student in Public and Private Planning, Master in Business Administration and Management, Public Accountant, Professor at IES CAP FAP Quiñones, Head of the Quality Unit of IES.

Oscar Napoleón Montoya Perez, Certified Public Accountant, Master in Public Management and already developing a doctorate in Public and Private Planning; with experience in the public and private sector, with mastery in: intermediate office (Word and Excel), Logistics (purchases and records) personnel management, with practical knowledge in taxation issues, basic level of English and Portuguese, with the experience and ability to develop teaching in higher education.

Grover Enrique Caballero Jimenez, Master in Public Management, Bachelor in Administration, with experience in business organization and management, oriented to continuous improvement and the achievement of objectives in multidisciplinary teams.