ARTICLE

JOURNAL OF UNIVERSITY OF BABYLON
For Pure and Applied Sciences (JUBPAS)

Vol.30; No.4.| 2022

Improvement the Community Detection with Graph Autoencoder in Social Network Using Correlation-Based Feature Selection Method

Hawraa Zuhair Ahmed [1*] and Asia Mahdi Naser Alzubaidi [2]

1College of Science for Women, University of Babylon, hawraazuhair @gmail.com, Babylon, Iraq.

2College of Computer Science & Information Technology, University of Karbala, Asia.alzubaidi2008@gmail.com, Karbala, Iraq.

***Corresponding author email:** hawraazuhair @gmail.com; mobile: 07706045019

تحسين اكتشاف المجتمع بـ Graph Autoencoder في الشبكات الاجتماعية باستخدام طريقة تحديد الميزات القائمة على الارتباط

حوراء زهير أحمد*[1]، آسيا مهدي ناصر الزبيدي [2]

1 كلية العلوم للبنات ، جامعة بابل، hawraazuhair @yahoo.com ، بابل، العراق.

2 كلية علوم الحاسوب وتكنلوجيا المعلومات ، جامعة ;كربلاء، Asia.alzubaidi2008@gmail.com ، كربلاء، العراق.

Received:	19/9/2022	Accepted:	13/11/2022	Published:	31/12/2022

ABSTRACT

Background:

In this paper, we aim to improve community detection methods using Graph Autoencoder. Community detection is a crucial stage in comprehend the purpose and composition of social networks.

Materials and Methods:

We propose a Community Detection framework using the Graph Autoencoder (CDGAE) model, we combined the nodes feature with the network topology as input to our method. A centrality measurement-based strategy is used by CDGAE to deal with the featureless dataset by providing artificial attributes to its nodes. The performance of the model was improved by applying feature selection to node features

The basic innovation of CDGAE is that added the number of communities counted using the Bethe Hessian Matrix in the bottleneck layer of the graph autoencoder (GAE) structure, to directly extract communities without using any clustering algorithms.

Results:

According to experimental findings, adding artificial features to the dataset's nodes improves performance. Additionally, the outcomes in community detection were much better with the feature selection method and a deeper model. Experimental evidence has shown that our approach outperforms existing algorithms.

Conclusion:

In this study, we suggest a community detection framework using graph autoencoder (CDMEC). In order to take advantage of GAE's ability to combine node features with the network topology, we add node features to the featureless graph nodes using centrality measurement. By applying the feature selection to the features of the nodes, the performance of the model has improved significantly, due to the elimination of data noise. Additionally, the inclusion of the number of communities in the bottleneck layer of the GAE structure allowed us to do away with clustering algorithms, which helped decrease the complexity time. deepening the model also improved the community detection. Because social media platforms are dynamic.

Key words:

Social Network, Community Detection, Graph Autoencoder, Feature Selection.

ISSN: 2312-8135 | Print ISSN: 1992-0652

info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

الخلاصة

مقدمة:

في هذا البحث ، نهدف إلى تحسين طرق اكتشاف المجتمع باستخدام Graph Autoencoder. يعد اكتشاف المجتمع مرحلة حاسمة لفهم الشبكات الاجتماعية وتكوينها.

طرق العمل:

نقترح إطار عمل اكتشاف المجتمع باستخدام نموذج Graph Autoencoder (CDGAE)، حيث قمنا بدمج ميزة العقد مع هيكل الشبكة كمدخل لطريقتنا. تستخدم CDGAE إستراتيجية قائمة على قياس المركزية للتعامل مع مجموعة البيانات الخالية من الميزات من خلال توفير ميزات اصطناعية لعقدها. تم تحسين أداء النموذج من خلال تطبيق تحديد الميزة على ميزات العقدة.

يتمثل الابتكار الأساسي لـ CDGAE في إضافة عدد المجتمعات التي تم حسابها باستخدام Bethe Hessian Matrix في طبقة عنق الزجاجة لبنية Graph Autoencoder (GAE) ، لاستخراج المجتمعات مباشرةً دون استخدام أي خوارزميات تجميع.

الاستنتاجات:

وفقًا للنتائج التجريبية ، تؤدي إضافة ميزات اصطناعية إلى عقد مجموعة البيانات إلى تحسين الأداء. بالإضافة إلى ذلك ، حصلنا على نتائج افضل بكثير في اكتشاف المجتمع باستخدام طريقة اختيار الميزة وبتعميق نموذج. أظهرت النتائج التجريبية أن نهجنا يتفوق على الخوارزميات الموجودة.

الكلمات المفتاحية:

الشبكة الاجتماعية ، اكتشاف المجتمع ، Graph Autoencoder ، اختيار الميزات

INTRODUCTION

Utilization of social networks has grown significantly in recent years. The term "social network" describes the use of web-based social media platforms like Facebook, Twitter, and WeChat to facilitate relationships with friends, family, or customers. Social network analysis (SNA) is currently one of the Data Science master's fields [1], [2]. Since these networks display some community structures, we must use community detection to reveal these structures in order to comprehend the behavior and organization of complex networks [3], [4]. Communities can be found in these networks, and it has been used for tasks like spammer detection and crisis response to infer relationships between individuals [5], [6].

In "community detection," complex network nodes are collected into groups that are heavily connected to one another and only loosely related to nodes in other communities [7].

The Graph Autoencoder (GAE) is a special type of GNNs that have recently been widely used in the field of machine learning, for their ability to deal with structured data. The capacity of GAE to learn unsupervised from the input data distinguishes it from other algorithms. The GAE system is made up of a number of interconnected parts that work in concert to gather data across nodes iteratively, capture the intricate dependencies of the underlying system, and approximate them in low dimensions [3], [8].

Research on community detection focuses on the following areas: First, the node features that, in addition to the graph's structure, can be used to identify similarities among the various

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

nodes and divide the graph into communities. Many real-world graphs lack node features due to privacy concerns or the onerousness process of collecting node features, hence a solution to dealing with featureless graphs must be found. Second, the issue of excessive dimensionality, which could cause learning algorithms to fail. Some node features may be misleading the results and not all of them may be relevant to the prediction. Finding a subset of the original features increases comprehensibility and improves the problem of high dimensions. Third, how to get results from clustering that are more precise for community discovery. In general, community networks are divided using the k-means clustering technique. Based on this technique, the findings of community detection are not very accurate. Therefore, a technique to produce high-resolution findings for community detection must be developed.

We have done many improvements to increase the precision of community detection in order to address the aforementioned problems. The following is a description of this paper's four primary contributions:

1. In order to boost the prediction capacity of GAE, node characteristics were added to featureless graphs using centrality measures.
2. To reduce data noise, two feature selection techniques were used, which improved the model's performance in community detection.
3. Counting the number of communities using the Bethe Hessian Matrix and utilizing it in the low dimensional representation of GAE bottleneck layer.
4. Additionally, the layers in GAE were increased (deepening the GAE model) so that GAE could improve its predictive capability.

is be use deep and graph autoencoder to reduce the The related works to our model dimensions of the input matrix, before collecting the communities. The modularity matrix was organized by Liang Yang et al. in 2016 [9] and used as an input to their deep nonlinear reconstruction (DNR) and (semi-DNR). Di Jin et al. in 2017 [10] used the normalized-cut and modularity matrix as input to their method, deep integration representation (DIR). Modularity and normalized-cut models were combined and used as inputs to deep autoencoder by Jinxin Cao et al. in 2018 [4]. The similarity matrix was used as the input for community detection with deep transitive autoencoder (CDDTA) method proposed by YingXie et al.'s in 2019 [11]. The adjacency matrix and feature matrix were inputs for Variational Graph Autoencoder for Community Detection (VGAECD) proposed by jun Jin et al. in 2020 [12]. The similarity matrix was used as the input by RongbinXu et al. in 2020 [13] for their Community Detection Method via Ensemble Clustering (CDMEC). After that, they extracted the reduced dimensions from the deep, graph autoencoder and used the Kmeans clustering algorithm to detect the community. While Chun Wang et al., in 2017 [14] combined the contents of the nodes with the network topology as input to their method, marginalized graph autoencoder (MGAE). The Laplacian matrix and the modularity matrix were inputs to Adaptive Autoencoder with Graph Regularization (AAGR) proposed by Cao J et al., in 2018 [15]. The community was then detected using the spectral clustering algorithm using the reduced dimensions obtained from the graph autoencoder.

While we combined the contents of the nodes with the network topology as input to our method, community detection framework using graph autoencoder (CDMEC), We then included the number of communities counted using the Bethe Hessian Matrix in the bottleneck layer of the GAE structure to directly extract communities.

Materials and Methods

We consider the connected and undirected graphs G= (V, E) of N=|V| nodes and the E edges connecting the two nodes in the graph, an adjacency matrix A∈R^(N×N) is the binary matrix such that Aij = 1 whenever (i,j)∈E , and Aij=0 otherwise, and the feature matrix X∈R^(N×F) that has the features of nodes. The ai is the ith row of the matrix A. The encoder layers, bottleneck layer, and decoder layers make up the three parts of the GAE model. The encoder and decoder components' hidden representations are calculated as follows:

Encoder layer: $$\acute{z}_i = \tau(W_i \bar{a}_i + b) \ldots(1)$$

Where źi is encoded ith column vector. τ is the rectified linear unit activation function [16], [17].

Decoder layer: $$\hat{\mathbf{a}}_i = \tau(W_2 \acute{z}_i + c) \ldots(2)$$

To optimize each layer's data, an activation function must be applied to it; one such activation function is ReLU, which is utilized for all hidden layers aside from the output layer, which uses the SoftMax activation function.

$$f(x) = ReLU = max(0, x) \ldots(3)$$

$$softmax(z)_i = \frac{exp(z_i)}{\sum_j exp(z_j)} \ldots(4)$$

The neurons' mapping range, when SoftMax is the activation function, is (0, 1). The output is said to be active when it is close to 1, while the output is said to be inactive when it is close to 0.

Learning During the forward pass, the model takes an input ai and computes its reformed output $\hat{\mathbf{a}}_i$. Through backpropagation, the optimizer parameter θ and loss function are learned. We estimate θ by minimizing by the optimizer (Adam optimizer) and the Cosine Similarity (sim) loss function during the backward pass. One of the most recent cutting-edge optimization algorithms utilized in deep learning is called Adam. The first time normalized by the second time gives the trend of the update.

Adam optimizer [18]: $$\theta_{n+1} = \theta_n - \frac{\alpha}{\sqrt{v_n + \epsilon}} \hat{m}_n \ldots(5)$$

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

Where α, $\hat{m}_n$and, v_n are the biased. Calculating the cosine similarity between labels and predictions using the cosine similarity loss function:

$$\text{sim}(\boldsymbol{x}, \boldsymbol{y}) = \frac{x \cdot y}{\|x\| \|y\|} \ldots(6)$$

where ||x|| is the Euclidean norm of vector x=(x1,x2,…,xp), defined as $\sqrt{x_1^2 + x_2^2 + \cdots + x_p^2}$. Conceptually, it is the length of the vector. Similarly, ||y|| is the Euclidean norm of vector y. This loss function's value ranges from -1 to 1. Higher similarity and greater dissimilarity are indicated by values closer to -1 and 1, respectively[19], [20].

- Number of Community

The spectral property of the Bethe Hessian matrix H, which is defined as follows, can be used to determine the number of communities:

$$H(r)=(r\text{^}2-1)\ I-rA+D..(7)$$

Where D= diag(di) is n × n diagonal matrix with degrees di on the diagonal, I is n × n identity matrix, and $r \in \mathbb{R}$ is a regularize value $|r| = r_c$ that define as following:

$$r_c = \sqrt{(\sum_{i=1}^{n} d_i)^{-1}(\sum_{i=1}^{n} d_i^2) - 1} \ldots(8)$$

The approach for determining the number of communities is by Finding the number of negative eigenvalues of matrix H(r) that represent the assortative features of the graph [21], [22].

- Centrality measures

Measures of centrality are a fundamental tool for understanding graphs. To determine the significance of each specific node in a graph, these algorithms employ graph theory [23], [24].

Local centrality measurements, iterative centrality measures, and global centrality measures are three different types of centrality measures that can be used [25], [26].

The 39-centrality measurement that listed in table (1) were calculated according to their equations mentioned on [27] for assigning as features for nodes.

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

Table (1): The Centrality Measurement

Local Centralities			
Degree Centrality	Local Entropy Measures	Distinctiveness Centrality	Semi-local Centrality
Mapping Entropy Measures	local clustering coefficient	Modified Local Centrality	ClusterRank
Neighborhood Connectivity	Volume Centrality	Average Neighbor Degree	Network Centrality
Clustering Coefficient	Node Density	Leverage Centrality	Square Clustering
Laplacian Centrality	Neighbor Based Centrality		
Global centrality			
Betweenness Centrality	Harmonic Centrality	Heatmap Centrality	Average Distance
Information Centrality	Eccentricity Centrality	Residual Closeness	Load Centrality
Bridgeness-Coefficient	Closeness Centrality	Radiality centrality	Wiener Index
Approximate Current Flow Betweenness	Harary Graph Centrality	Barycenter Centrality	Flow Betweenness Centrality
Iterative Centrality			
Eigenvalue Centrality	PageRank	Diffusion Centrality	Katz
Subgraph Centrality			

- **Feature Selection**

The obtained data typically has a lot of noise attached to it. The two main causes of noise in these data are the restricted technology that acquired the data and the source of the data itself [28], [29].

With the increasing the number of features, the dataset becomes larger. The process of limiting the amount of input variables to those thought to be most helpful to a model's ability to predict the target variable is known as feature selection. Reducing the number of inputs is useful to both reduce the computational cost of modeling and, in some cases, to improve the performance of the model [30].

There are information-based methods for unsupervised feature selection, including the Correlation-based.

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

The Correlation-based feature selection (CFS): This method is a filter approach where it assesses feature subsets just based on the information intrinsic properties, so it autonomous of the final classification model. its aim is to discover a feature subset with low feature-feature correlation, to avoid repetition, and high feature-class correlation to preserve or grow the predictive force [30], [31].

Results and Discussion

In this section, we describe our suggested system's architecture, in terms of the input data and the model's design and the experimental Results and Discussion.

1- Community Detection using Graph Autoencoder system (CDGAE)

- **Feature Initialization**

The explicit data available in datasets that take the form of an edge list, is used. Since the observed datasets lack any node features, artificial node features are assigned.

For the feature initialization, we use the centrality-based measurements, which are represented by the centrality measurements provided in table (1), For each node, these centrality measures are computed and assigned as its features in feature matrix X.

- **Feature Selection.**

We employ feature selection method since they can decrease model complexity, boost learning effectiveness, and increase predictive power by reducing noise.
correlation-based feature selection method eliminates redundancy features and finding a feature subset with low correlation to features CX.
In this method, we searching the subset with a high feature-class correlation by a specific percentage, then we delete it while keeping one feature on behalf of this subset.

- **Design CDGAE structure**

After the data has been collected and initialized, we discuss the layout of the proposed system that analyzes this data to achieve the suitable community detection.

The method includes calculating the number of communities NC via spectral method with the Bethe Hessian matrix, and then allocating the nodes to their communities depending on our model input, that is concatenate (A, X/CX) adjacency matrix A and feature matrix (X or CX) make up the augmented adjacency matrix $\bar{A} \in R^{\wedge(N\times(N+F))}$.

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

Our suggested CDGAE architectural model includes a series of non-linear transformations input $\bar{a}_i$ that are broken down into two parts: encoder $g(\bar{a}i): \mathbb{R}^N \rightarrow \mathbb{R}^{CN}$, and decoder $f(\mathbf{z}_i): \mathbb{R}^{CN} \rightarrow \mathbb{R}^N$. To see how the number of layers affects GAE performance, we stack one, three, or five layers of the encoder part to create a CN-dimensional low-level representation of the ith node $\mathbf{z}_i \in \mathbb{R}^D$, and then one, three, or five layers of the decoder part to produce an approximative reformed output $\hat{\mathbf{a}}_i \in \mathbb{R}^N$. This creates a two-, six-, or ten-layer GAE architecture, see figure (1).

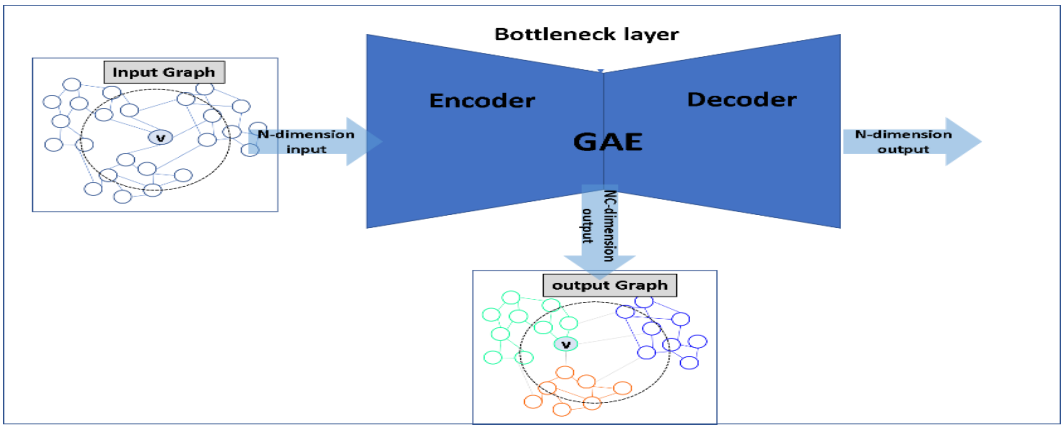

Figure (1): architecture of the CDGAE model

In our suggested model, the number of neurons for the input layer is equal to the input $\bar{A}$ (the number of graph nodes in addition to the number of features) which is equal to the number of outputs from the final decoder layer. In each layer between the bottleneck layer and the output layer, the neurons double, and the number of neurons for the bottleneck layer is equal to the number of communities NC. The neurons layers between the input layer and the bottleneck layer also diminish in size.

Using the CosineSimilarity loss function and the Adam optimizer function, the model is trained during these epochs, which results in a reduction in loss for the training set of data.

Following model training, the trained encoder is disconnected from the base model for employ it to predict the node communities.

- **System Evaluation**

Because we lack ground-truth graphs with well-known communities, we are creating LFR computer-generated graphs as a substitute for the best way to evaluate the efficacy of the community detection algorithm.

To provide us with a rough built-in community with the ground truth of the graph and to obtain an accurate evaluation, we prepared a variety of LFR graphs with properties comparable to those of our datasets.

The Normalized Mutual Information was employed (NMI). A measure called NMI is used to rate the graph division done by algorithms for discovering communities, it defined by:

$$NMI(Y,C)=(2\times I(Y;C))/(H(Y)+H(C)) \quad \ldots(9)$$

Where I denotes Mutual Information, H(Y) stands for Entropy of Labeled, and H(C) denotes community groups [32], [33].

2- Experimental Results and Discussion

We discuss the experiences in this section. Version 3.9.7 of the Python programming language was used to execute the CDGAE.

- **Experimental Results**

Three angles can be used to discuss the experimental results. The first has to do with the depth of the system, which is represented by the number of layers for CDGAE, the second with adding features to nodes, and the third with applying feature selection to nodes' features. following the application of our model on real-world graphs.

- **Datasets**

The eight datasets in table (2), which provide undirected and connected graphs without node features, are used to experience the performance of our model.

Three of these datasets are from Facebook pages (November 2017). These datasets represent Facebook page networks, of (Government, Politicians, and TV Shows) categories, where nodes represent the pages and edges are mutual likes between them[34]

Four of these datasets come from social networks of gamers who stream on Twitch (gathered in May 2018), where the nodes are the players themselves and the links are their friendships between them [35].

Football: a regular-season Fall 2000 network of American football contests between Division IA universities [36].

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

The CDGAE model was created and then trained for 300 epochs. The bottleneck layer is where the graph's crucial data is concentrated in this case.

Table 2: the dataset and number of nodes(N), edges(E), and communities (NC)

Dataset	N	E	NC
government	7057	89455	90
politician	5908	41729	73
TV show	3892	17262	80
Spain gamers	4648	59382	20
France gamers	6549	112666	24
Portugal gamers	1912	31299	13
Russia gamers	4385	37304	15
football	115	623	12

The encoder was extracted and utilized it to find the communities by using the argmax function to determine the node membership for any community.

- **Evaluation the community detection system.**

After detecting the communities nodes, we assess the system's performance by contrasting its community results with the communities represented by the LFR benchmark graphs on the test data.

For the purpose of assessing our system, we generated a collection of LFR benchmark graphs with properties resembling those of the dataset utilized in our experiments.

- **Experimental results of the evaluation metrics applied to the CDGAE models.**

We utilized the metrics NMI score to assess the effectiveness of our model.

Because the features of the nodes describe their orientations and inclinations, the performance of the community detection considerably improved with the addition of features to the nodes. By applying the feature selection method to the features of the nodes, the performance of the model has improved significantly, due to the elimination of data noise. With the exception of times when the performance of community detection dropped in the 5-layer model, adding more layers (deepening the model) improved it. This could be because some critical information is missing between the layers.

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

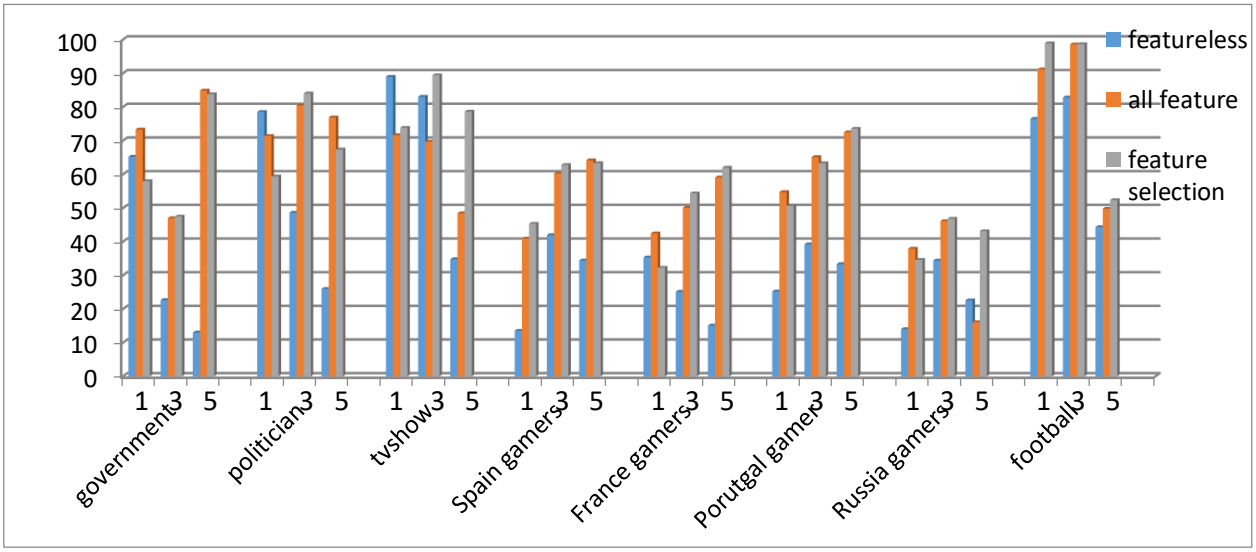

Figure (2): The charts of the CDGAE performance in term of NMI metric 8 real-world datasets.

Table (3) compares the performance of CDGAE and three additional community detection algorithms for community detection in terms of NMI.

Table (3): comparison CDGAE method and other existing methods in terms of NMI

Method	Facebook government	Facebook politician	Facebook TV show	Spain gamers	France gamers	Portugal gamers	Russia gamers	football
DNR [10]	-	-	-	-	-	-	-	92.7
CDDTA[11]	-	-	-	-	-	-	-	97
CDMEC[13]	-	-	-	-	-	-	-	98
CDGAE	84.9	84	89.4	64.2	62	73.5	46.8	98.9

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

Conflict of interests.

There are non-conflicts of interest.

References

[1] W. R. Scott and G. F. Davis, "Organizations and Organizing Rational, Natural, and Open System Perspectives," 2007, Accessed: Sep. 02, 2022. [Online]. Available: www.irpublicpolicy.ir

[2] P. Bedi and C. Sharma, "Community detection in social networks," *Wiley Interdiscip Rev Data Min Knowl Discov*, vol. 6, no. 3, pp. 115–135, 2016.

[3] J. Sun, W. Zheng, Q. Zhang, and Z. Xu, "Graph Neural Network Encoding for Community Detection in Attribute Networks," Jun. 2020, [Online]. Available: http://arxiv.org/abs/2006.03996

[4] J. Cao, D. Jin, L. Yang, and J. Dang, "Incorporating network structure with node contents for community detection on large networks using deep learning," *Neurocomputing*, vol. 297, pp. 71–81, Jul. 2018, doi: 10.1016/j.neucom.2018.01.065.

[5] B. S. Khan, · Muaz, and A. Niazi, "Network Community Detection: A Review and Visual Survey."

[6] D. Jin *et al.*, "A Survey of Community Detection Approaches: From Statistical Modeling to Deep Learning," Jan. 2021, [Online]. Available: http://arxiv.org/abs/2101.01669

[7] D. Singh and R. Garg, "Issues and Challenges in Community Detection Algorithms".

[8] V. Bhatia and R. Rani, "DFuzzy: a deep learning-based fuzzy clustering model for large graphs," *Knowl Inf Syst*, vol. 57, no. 1, pp. 159–181, Oct. 2018, doi: 10.1007/s10115-018-1156-3.

[9] L. Yang, X. Cao, D. He, C. Wang, X. Wang, and W. Zhang, "Modularity Based Community Detection with Deep Learning," 2016.

[10] D. Jin, M. Ge, Z. Li, W. Lu, D. He, and F. Fogelman-Soulie, "Using deep learning for community discovery in social networks," in *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, Jun. 2018, vol. 2017-November, pp. 160–167. doi: 10.1109/ICTAI.2017.00035.

[11] Y. Xie, X. Wang, D. Jiang, and R. Xu, "High-performance community detection in social networks using a deep transitive autoencoder," *Inf Sci (N Y)*, vol. 493, pp. 75–90, Aug. 2019, doi: 10.1016/j.ins.2019.04.018.

[12] J. J. Choong, X. Liu, and T. Murata, "Optimizing Variational Graph Autoencoder for Community Detection with Dual Optimization," *2020*, doi: 10.3390/e22020197.

[13] R. Xu, Y. Che, X. Wang, J. Hu, and Y. Xie, "Stacked autoencoder-based community detection method via an ensemble clustering framework," *Inf Sci (N Y)*, vol. 526, pp. 151–165, Jul. 2020, doi: 10.1016/j.ins.2020.03.090.

[14] C. Wang, S. Pan, G. Long, X. Zhu, and J. Jiang, "MGAE: Marginalized Graph Autoencoder for Graph Clustering", doi: 10.1145/3132847.3132967.

[15] J. Cao, D. Jin, and J. Dang, "Autoencoder based community detection with adaptive integration of network topology and node contents," in *Lecture Notes in Computer Science (including subseries*

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2018, vol. 11062 LNAI, pp. 184–196. doi: 10.1007/978-3-319-99247-1_16.

[16] R. Fei, J. Sha, Q. Xu, B. Hu, K. Wang, and S. Li, "A new deep sparse autoencoder for community detection in complex networks," *EURASIP J Wirel Commun Netw*, vol. 2020, no. 1, Dec. 2020, doi: 10.1186/s13638-020-01706-4.

[17] P. V. Tran, "Learning to Make Predictions on Graphs with Autoencoders," Feb. 2018, doi: 10.1109/DSAA.2018.00034.

[18] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," Dec. 2014, [Online]. Available: http://arxiv.org/abs/1412.6980

[19] J. Han, M. Kamber, and J. Pei, "Data Mining. Concepts and Techniques, 3rd Edition (The Morgan Kaufmann Series in Data Management Systems)," 2011.

[20] "tf.keras.losses.CosineSimilarity | TensorFlow Core v2.9.1." https://www.tensorflow.org/api_docs/python/tf/keras/losses/CosineSimilarity (accessed Aug. 17, 2022).

[21] C. M. Le and E. Levina, "Estimating the number of communities in networks by spectral methods," Jul. 2015, [Online]. Available: http://arxiv.org/abs/1507.00827

[22] A. Saade, F. Krzakala, and L. Zdeborová, "Spectral Clustering of Graphs with the Bethe Hessian," Jun. 2014, [Online]. Available: http://arxiv.org/abs/1406.1880

[23] Z. Wan, Y. Mahajan, B. W. Kang, T. J. Moore, and J. H. Cho, "A Survey on Centrality Metrics and Their Network Resilience Analysis," *IEEE Access*, vol. 9, pp. 104773–104819, 2021, doi: 10.1109/ACCESS.2021.3094196.

[24] A. Landherr, B. Friedl, and J. Heidemann, "A Critical Review of Centrality Measures in Social Networks," *Business & Information Systems Engineering*, vol. 2, no. 6, pp. 371–385, Dec. 2010, doi: 10.1007/s12599-010-0127-3.

[25] K. Das, S. Samanta, and M. Pal, "Study on centrality measures in social networks: a survey," *Social Network Analysis and Mining*, vol. 8, no. 1. Springer-Verlag Wien, Dec. 01, 2018. doi: 10.1007/s13278-018-0493-2.

[26] A. Saxena and S. Iyengar, "Centrality Measures in Complex Networks: A Survey," Nov. 2020, [Online]. Available: http://arxiv.org/abs/2011.07190

[27] et al. Jalili Mahdi, Salehzadeh-Yazdi Ali, "CentiServer." https://www.centiserver.org/centrality/list/

[28] J. G. Dy and C. E. Brodley, "Feature Selection for Unsupervised Learning," 2004.

[29] S. Solorio-Fernández, J. A. Carrasco-Ochoa, and J. F. Martínez-Trinidad, "A review of unsupervised feature selection methods," *Artif Intell Rev*, vol. 53, no. 2, pp. 907–948, Feb. 2020, doi: 10.1007/s10462-019-09682-y.

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com

[30] J. Tang, S. Alelyani, and H. Liu, "Feature Selection for Classification: A Review."

[31] M. A. Hall, "Correlation-based Feature Selection for Discrete and Numeric Class Machine Learning".

[32] X. Liu, H.-M. Cheng, and Z.-Y. Zhang, "Evaluation of Community Detection Methods," Jul. 2018, [Online]. Available: http://arxiv.org/abs/1807.01130

[33] "Normalized Mutual Information. A measure to evaluate network… | by Luís Rita | Medium." https://luisdrita.com/normalized-mutual-information-a10785ba4898 (accessed Aug. 25, 2022).

[34] "Datasets." https://github.com/benedekrozemberczki/datasets#github-stargazer-graphs

[35] "SNAP: Network datasets: Wikipedia Article Networks." https://snap.stanford.edu/data/twitch-social-networks.html (accessed Sep. 18, 2022).

[36] "Network data." http://www-personal.umich.edu/~mejn/netdata/ (accessed Sep. 15, 2022).

ISSN: 2312-8135 | Print ISSN: 1992-0652
info@journalofbabylon.com | jub@itnet.uobabylon.edu.iq | www.journalofbabylon.com