Community Detection in Social Networks: A Deep Learning Approach Using Autoencoders

*Priyanka Gupta[1], Mamta Arora[2], Hardeo Kumar Thakur[3]
[1]Research Scholar, Department of CSE, Manav Rachna University, Faridabad, India
[2]Department of CSE, Manav Rachna University, Faridabad, India
[3]Bennett University, Greater Noida, India
E-mail: drpriyankagupta.research@gmail.com, imamta.arora@gmail.com, hardeokumar@gmail.com
*Corresponding author

Keywords: community detection, social network, clustering algorithm, girvan–newman, autoencoder, deep learning, artificial intelligence

Received: June 14, 2024

This research aims to propose a more sophisticated clustering and community detection technique in complex social networks through the use of neural networks; autoencoder, in particular. In the past, methods for network analysis and community detection used several graph algorithms, but with the emergence of deep learning, autoencoders are used for learning node features. These node representations are learnt using a neural network-based autoencoder and then clustering algorithms like k-means, agglomerative and spectral clustering are performed. These algorithms are then improved by incorporating with the Louvain algorithm for community detection. The proposed method, named the Spectral Louvain Algorithm, offers several advantages: it saves the stage of feature extraction, is suitable for Call Detail Record (CDR) and Social Network Analysis (SNA), does not require model retraining for different scale networks, and can work in mesh scale networks. It has better accuracy and performance than former approaches, even getting 100% of NMI and ARS, 78% of modularity in Karate Club and 89% of NMI, 93% of ARS and 86% of modularity in Dolphin data set with the least conductance. This method is particularly useful in discovering new relations and structures in a system and it is very efficient.

Povzetek: Članek uvaja metodo Spektral Louvain za odkrivanje skupnosti v družbenih omrežjih, ki združuje avtomatske kodirnike, spektralno združevanje in Louvain algoritem. Metoda dosega visoko modularnost in prilagodljivost velikosti omrežij.

1 Introduction

It is regarded one of the most important tasks of the network analysis studies – the identification of communities in complex networks[1] which concerns the identification of the groups of nodes densely connected internally, while the links between groups are either sparse or lack at all. Detecting communities [1] can certainly help in understanding the functional organization in different fields including social networks (SN) [2], Biological networks (BioNet), communication networks (CN) [3], [4] etc.

Classical methods of community identification include the use of graph concepts [5] where information flows such as modularity optimization [6], spectrum analysis of the graph[7], and hierarchical clustering [8]are used.

These methods usually need a set of features to be designed manually and can be less effective for large-scale networks with complex patterns. Furthermore, these methods perform poorly in practice, especially when the data is noisy or high-dimensional and are not very scalable and very fragile [9].

Deep learning is a relatively novel technique that tackles feature extraction from raw data in different fields[10]. Auto-encoder [9] is a class of neural networks that can learn a compact and meaningful representation of the data input, based on the input requirement that the network be able to reconstruct the output data as closely as possible [9]. Auto-encoders are beneficial for community detection problems, especially when the internal structure of networks is intricate and could be multi-dimensional, Due to this capability of extracting features from data automatically [9].

This paper is communicating the three significant ways, Firstly, autoencoder algorithms [9] are used to reconstruct communities and provide the node embedding. In this research, algorithms are implemented on two datasets, karate club datasets [11] and Dolphin's datasets [12]. These datasets produce actual communities employing the adjacency matrix which defines the graph data required to discover the communities [13]. Secondly, various clustering methos are applied to learn the node embeddings. Thirdly, these clustering methods are combined with the Louvain community detection algorithm to improve the modularity [14]. The subsequent sections of this paper are structured as follows to outline the process: section ii contains the related work and associated research with potential limitations. Section iii contains the proposed methodology and contributions to explain the novelty of the approach, Section iv contains the research methodologies and techniques employed.

Section v contains benchmark datasets and community validation for preprocessing of benchmark data set Zachary's karate club [11] and dolphins [12] . section vi of our work includes an analytical approach. Section vii covers the preliminaries and proposed algorithms; section viii covers the results and outcomes in which few Evaluation Methods for Assessing Algorithm Performance are discussed. ix contains the outcome and concludes our findings. section ix contains the future scope. section xii contains the nomenclature table.

2 Related works

For the past decades, the algorithms concerned with the classification of communities have evolved greatly, and the recent focus has been on the application of deep learning techniques in enhancing traditional methods [10]. In addition to the well-known techniques, such as modularity optimization and spectral clustering, deep learning-based methods, especially with the use of autoencoders, have become an interesting topic for research in the field of community detection in complex networks [15].

Autoencoder derived methodologies use neural network-based architecture to extract the relevant features and learn the representations of the Network data [9]. This has the following advantages: feature extraction is done automatically and the training with the end-to-end model. These approaches try to identify complex densities and forms in networks, which are different from traditional algorithms.

The earlier methods that are still widely used are modularity optimization by [16] and spectral clustering by von Lux burg [17] while the other methods include self-organizing hierarchal clustering by Jain and Dubes [18] Moreover, the Louvain method developed by Blondel [19] has recently attracted much interest because of its ability to efficiently maximize modularity and identify communities in huge networks.

Few recent studies have also been carried out in this area by authors D. Jin et al., [20] who offer a review of several community detection approaches with emphasis on how deep learning can be incorporated. One of the limitations pointed out here is the lack of standardization in evaluation metrics and benchmark datasets, which prevents the easy comparability of various community detection algorithms [20]. Thomas Kipf & Max Welling [21] learned that the graph convolutional networks that incorporate node features and graph structure outperform other methods in semi supervised learning problems. According to Yang et al. community detectors make use of graph neural networks [22] doing so captures various relationships between the nodes. Bhattacharya et al. [23] offered the most closely related method Embedded graph convolutional networks improve community detection by learning node representations and utilizing structural data. In their paper Xing Su et al.[24] provides a comprehensive review of different types of deep learning in relation to community detection while elucidating the existing progress and issues in this field. Zhang et al. [22] the author presents contrastive graph autoencoders for community detection to enhance the clustering efficiency by using the contrastive learning approach. Apicella et al. [25] discussed the adaptive filters in graph convolution networks and revealed how the community detection enhances the clustering performance due to contrastive learning. X. Li et al.[26] work relates to temporal graph convolutional networks for detection of communities in dynamic graph solving for evolving structures.

From the preceding discussion, it is apparent that deep learning and autoencoder-based techniques in combination with clustering methods along with the help of community detection algorithms offer new prospects for improving the effectiveness of community detection system. The rationale is to enhance understanding of the structures using conventional approaches along with utilization of neural networks and, consequently, discover new patterns in different fields of science.

3 Proposed methodology and contributions

3.1 Flow of work

This research introduces a novel community detection method that leverages feature discovery using autoencoders, network structure using spectral clustering and community detection using the Louvain algorithm. The process starts with feature learning for node data, using autoencoder and converting high-dimensional data to low-dimensional data where typically, a massive amount of effort is required in feature engineering. The autoencoders are involved in capturing structural relationships within the network that may be hidden. To localize these node embeddings, a process called spectral clustering is used to detect nonlinear separations between communities and therefore provide an optimal partition of nodes which is determined by eigenvectors of Laplacian matrix of the graph. Lastly the Louvain algorithm is applied to enhance the modularity for the formation of proper communities of nodes. This hybrid methodology, termed Spectral Louvain, improves both accuracy and scalability of community detection, making it ideal for real-world complex network analysis.

3.2 Novelty of the study

The novelty of proposed approach consists in its hybrid integration of autoencoders, spectral clustering, and the Louvain algorithm which has not been extensively explored in the existing literature. Although each of these techniques has been employed in previous research, the combination presented here —particularly in the parallel process—is more effective to enhance the feature learning capability, the global structure perception, and modularity optimization step. Moreover, the disclosed method can learn and improve without retraining if the network size changes, which is the main advantage in terms of scalability over the prior art.

3.3 Key Ffndings

The proposed Spectral Louvain method shows its higher efficiency comparing to other community detection methods in terms of accuracy and scalability. Experimentation carried out on widely used benchmark datasets Zachary's Karate Club, Dolphin social networks substantiate the effectiveness of the proposed solution in delivering better results across multiple metrics. Specifically, the method achieved perfect scores in both Normalized Mutual Information (NMI) and Adjusted Rand Score (ARS) on the Karate Club dataset[27], while yielding high values in NMI, ARS, and modularity on the Dolphin dataset. These results highlight clear advancements in the performance profiles of different community detection tasks in terms of modularity and the smallest conductance. In the light of such a discussion, the Spectral Louvain approach emerges as one of the most effective and feasible methods to identify the communities in large scale networks.

4 Research methodologies and techniques employed

In the present methodology, A novel community detection approach leverages auto-encoders for feature learning followed with the clustering and community detection method is used to increase the performance of traditional methodology.

4.1 Auto-Encoder feature learning

Auto-encoder structure is used to learn node representations from the data in the network [9]. The autoencoder encodes the input data and tries to reconstruct it in the next layer while learning a compressed representation in the middle layer. The transformation of original network into an embedding space is more conducive for clustering [25]. The autoencoder with low noise is usually defined as:

$$\mathrm{z} = f_{encoder}(x) \tag{1}$$

$$(\hat{\mathrm{x}}) = f_{decoder}(z) \tag{2}$$

loss function of a simple autoencoder is defined as:

$$\mathrm{L}(\mathrm{x}, \hat{\mathrm{x}}) = ||\mathrm{x} - \hat{\mathrm{x}}||^2 \tag{3}$$

where x signifies the input and $\hat{x}$ is the reconstruction produced by the decoder.

4.2 Clustering techniques

Various clustering techniques are applied to learned node embeddings to partition nodes into communities [23] :

- **K-means clustering:** *S*plits nodes into K groups by similarity within a specified embedding space [28]. The purpose of the K-means algorithm is to reduce the total of the difference between each point and the centroids. given by:

$$\mathrm{Q} = \min_{\{m_k\}_{k=1}^{K}} \sum_{i=1}^{n} \min_{k} ||X_i - m_k||^2 \tag{4}$$

where X_i represents the input data and m_k is the centroid of the kth cluster [29].

- **Spectral clustering***:* Clusters the nodes by assigning eigenvectors [30] of the Laplacian of the graph G = (V, E). The Laplacian of the graph is defined as [22]:

$$\mathrm{L} = D - A \tag{5}$$

D is the diagonal matrix of the node degrees and A is with the adjacency matrix of the graph G. The eigenvectors, therefore, correspond to the small eigenvalues of L, are for clustering tasks in low-dimensional space [17].

- **Agglomerative clustering**: The process of combining more nodes into fewer clusters according to the specified distance measure [31]. The method applied in finding the distances between two clusters C_A, C_B based on the average linkage metric is defined as [32]:

$$\mathrm{d}(C_A, C_B) = \frac{1}{|C_A|,|C_B|} \sum_{i \in C_A} \sum_{j \in C_B} d(\mathrm{i}, \mathrm{j}) \tag{6}$$

d (i, j) is the distance between the points i and j of cluster C_A and C_B [32].

4.3 Community detection methods

Among the community detection methods, the most popular ones are the Girvan Newman[3] and Louvain methods [14] .

- **Girvan-Newman:** Proceeds by repeatedly scanning for edges with high betweenness centrality to identify the community structure [33].
- **Louvain algorithm:** Recurrently relinquishes nodes to enhance the modularity of the network since communities tend to connect to other communities within the same community [1] [14].

4.4 Combining community detection with clustering

Integration of community detection with clustering approaches such as K-means, Agglomerative Clustering, and Spectral Clustering adds significant value, as it not only considers the network structure but also optimizes the Modularity score. This approach is compared with communities generated by the autoencoder algorithm, and the combination of these clustering and community detection algorithms offers a more accurate and detailed detection of community structures in both large-scale networks [19], [34] and complex Network [35].

• **Louvain K-means:** The procedure used here is a merger of the Louvain method and K-means clustering to find community.

• **Louvain agglomerative:** Automate the Louvain method using agglomerative clustering for enhanced detection of clusters.

• **Spectral louvain:** Also, this approach incorporates the use of the Louvain method in conjunction with spectral clustering to improve community detection.

4.5 Evaluation and validation

The proposed method is then tested and assessed based on existing real-life data sets with reference solutions to calculate the communities found [2]. Evaluating Quality, the proposed method is tested and evaluated based on a comparison with existing communities with the same data set Karate Club and Dolphin to calculate the found communities. Evaluation of the quality [2] of detected communities: based on modularity [6] and normalized mutual information (NMI) [27], the silhouette score [36].

The approach formulated here has some advantages over other methods: It combines the auto-encoder feature learning with the combination of K-means, Spectral Clustering, Agglomerative Clustering, Girvan-Newman, and Louvain methods to improve the potential of the method in detecting complicated graphs and structures in network data. This integration helps to achieve better and more stable community detection by combining the advantages of deep learning feature extraction and graph-based methods. In addition, the approach is scalable, it can control a large set of networks.

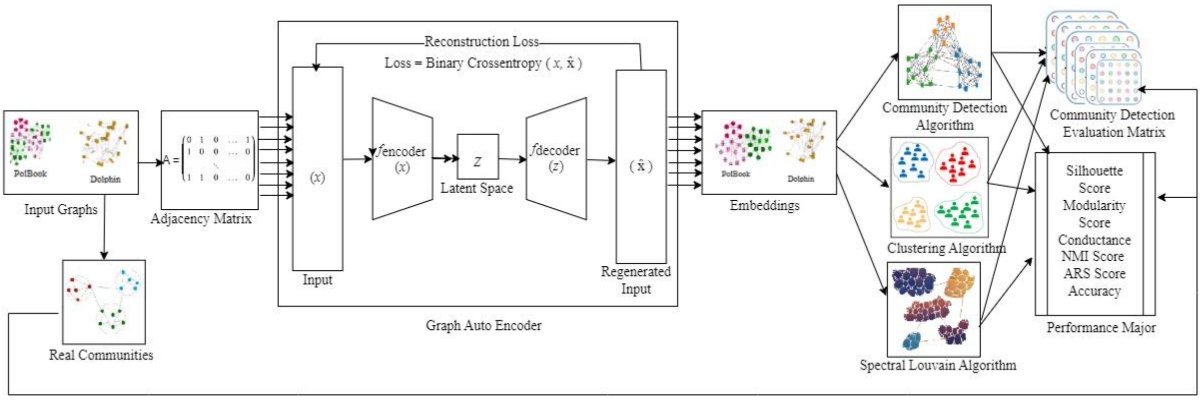

Figure 1: workflow for community detection by using the deep learning AI

5 Benchmark datasets and community validation

Benchmark datasets are used in many research projects to assess the effectiveness of the algorithms being used. In this study, two prominent benchmark datasets were employed: Zachary's Karate Club [11] and the Dolphins [12].

Zachary's karate club:

- **Description:** This data set consists of the social relations of 34 members of a karate club [37],[11] which later leads to the division of the club into two groups.
- **Relevance:** It provides a clear ground truth suitable for comparing the efficacy of community detection algorithms.

- **Dolphins:**
  - **Description:** For this case, the dataset composed of 62 bottlenose dolphins exhibits the social network found through association indices [12].
  - **Relevance:** It shows a denser and more sophisticated network taken from real life, which is useful to test community detection algorithms.

With a given set of datasets and a deep learning approach, the increased benchmarking, validation, and reproducibility with the combination of clustering and the community detection algorithm presented here reveals the effectiveness of the methodology in terms of community detection in a variety of networks [38], [39], [40].

6 Analytical approach

The analytical approach in this research combines modern methods for the identification of communities in complex networks [35]. First, the autoencoder algorithms are used to learn compact and semantically relevant node representations from the given input matrices. These representations are then used to provide an efficient way to cluster the nodes into communities using several clustering algorithms [9]. They are K-means, Spectral clustering, Agglomerative clustering, Girvan Newman and Louvain methods. Thus, combining the capabilities of deep learning for feature extraction and the graph-based clustering approach for the extraction of structural and complex pattern features, it will be possible to improve the existing approach and reveal high-level patterns in network data. Based on the quality of the detected communities [2] the model parameters such as modularity, normalized mutual information (NMI) [27,32], and Silhouette scores are analyzed [36], and the model is tested for scalability on datasets of different sizes.

6.1 Community detection algorithm and evaluation workflow

The analytical approach in this research combines modern methods for the identification of communities in complex networks [1], [35] . First, the autoencoder algorithms are used to learn compact and semantically relevant node representations from the given input matrices. These representations are then utilized to cluster the nodes into communities using various clustering algorithms. [30]. They are K-means [29], Spectral clustering [17],

Agglomerative clustering [31], Girvan Newman [16] and Louvain method [14]. Thus, combining the capabilities of deep learning for feature extraction and the graph-based clustering approach for the extraction of structural and complex pattern features, it will be possible to improve the existing approach and reveal high-level patterns in network data. Based on the quality of the detected communities the model parameters such as modularity [16], normalized mutual information (NMI) [27], adjusted Rand score (ARS) [35], and silhouette score are analyzed to assess the quality of the detected communities, and the proposed model is also tested for scalability on the two benchmark datasets of different size karate club and dolphin. The given figure 1 depicts the process workflow of the proposed community detection model for checking the effectiveness of comprehensive network structures.

7 Preliminaries

Consider a network graph G, comprising V_e nodes and E_v edges, G (V_e, Ev). The adjacency matrix [5] A of G, where Aij ∈ {0, 1, signifies the presence of an edge between, nodes i and j [5].

7.1 Algorithm overview autoencoder

The proposed method utilizes Graph Autoencoder (GAE) to learn latent representations of nodes from the network's adjacency matrix. This helps the auto encoder to learn the embeddings [7]. Represented by the algorithm 1-5.

Algorithm 1, The Graph Autoencoder (GAE) Communities is used as an algorithm that can learn the latent features for graphs. It establishes a model with the help of input parameters latent_dim and adj_matrix and sets the autoencoder encoder. The encoder is a sequential neural network with dense layers to map the graph's adjacency matrix to a latent space. The decoder then maps the embeddings back to the original space to reconstruct the input in the matrix form of the adjacency matrix using the learned embeddings.

Algorithm 1: Graph Autoencoder Community

Input: - latent_dim, adj_matrix
Output: -Initialized GraphAutoencoder model
1: **procedure** init(LatentDim, adj_matrix)
2: **Initialize** superclass
3: **Set** self.LatentDim to LatentDim
4: self. encoder ← Sequential([
5: Dense (Latnt_dim, activation ==" LeakyReLU"),
6: Dense (2 * LatntDim, activation= "LeakyReLU"),
7: Dense (4 * LatntDim, activation = "LeakyReLU"),
8: Fully connected (8 * LatntDim, activation LeakyReLU)])
9: self.decoder ← Dense(adj_matrix. shape[0], activation='sigmoid')
10: **end procedure**

Algorithm 2, The Graph Autoencoder is a model that takes input data, applies a forward pass of an unspecified algorithm to produce a latent representation, and then passes it through the decoder to reconstruct the original input.

Algorithm 2: Graph Autoencoder Call Function

Input: inputs
Output: Decoded output
1: **procedure** call(inputs)
2: Encoded ← selfEncoder(inputs)
3: Decoded ← selfDecoder(encoded)
4: return Decoded
5: **End procedure**

Algorithm 3, The loss function computes the difference of true and the predicted adjacency matrix in order to measure how well the model is performing by changing the parameters of autoencoder.

Algorithm 3: Loss Function

Input: - YTrue, YPred
Output: - Loss value
1: **procedure** loss_function (YTrue, YPred)
2: return mean (binary_crossentropy (YTrue, YPred))
3: **end procedure**

Algorithm 4, concerns reading data from a GML file to transform the graph into a format used by the Graph Auto-encoder model, an adjacency matrix. The algorithm primarily processes the GML file by reading the graph and converting it into a dense adjacency matrix, which is then used as the input graph and its corresponding adjacency matrix.

Algorithm 4: To read GML data

```
Input: - path
Output: - Graph G, Adjacency Matrix
procedure read_gml_data(path)
    G ← read_gml(path, label='id')
    adjacency ← AdjacencyMatrix(G). todense()
    return G, adjacency
end procedure
```

Algorithm 5, Predict the communities by using Girvan Newman Predictions and Louvain communities.

Algorithm 5: Generate Community Prediction

```
Input: - Graph, graph_from_similarity_embeddings, seed
Output: - predictions, louvain_predictions
procedure Girvan_Newman_Predictions (Graph,
    graph_from_similarity_embeddings, seed)
                > #set random seed for reproducibility of the results
    communities ← girvan_newman (Graph)
    node_groups ← []
    for i in range 4 do
        for c in next(communities) do
            if i = 2 then
                node_groups. append(c)
            end if
        end for
    end for
louvain_communities←louvain_communities (graph_
    from_similarity_embeddings)
    Predictions ← [0] * lLen (GraphNodes)
    louvain_predictions = [0] * lLen (GraphNodes)
    for i in the range lLen (NodeGroups) do
        for (j, k) in zip(node_groups[i],
                louvain_communities[i]) do
            if '0' is present in Graph. nodes then
                predictions[j] ← i
            else
                predictions[j-1] ← i
            end if
            louvain_predictions[k] ← i
        end for
    end for
    return predictions, louvain_predictions
end procedure
```

The learned embeddings are then clustered using Clustering [5], agglomerative clustering [31], and various techniques, including K-means [28], Spectral after that the Final communities are generated using the community detection algorithms [38] Girvan Newman [33] and Louvain methods [14].

This integration of autoencoder-based feature learning and clustering algorithms and community detection [39] algorithms enhance the ability to detect communities in complex networks [34]. Among all these approaches we find the best approach that with the combination of spectral and Louvain represented by algorithms 6.

Algorithm 6: Spectral Louvain

```
Input: - Graph G, number of clusters k, resolution parameter
gamma,
 random seed
Output: - Final partition
procedure spectral_louvain (G, k, gamma, seed)
        # Set random seed for reproducibility of the results
>## Step 1: Calculate the Laplacian matrix of the graph L
L ← laplacian_matrix(G). todense ()

>## Step 2: Let the eigenvectors be v=(v1, v2,……..vn) of L
 such that Lv=kv where k is an eigenvalue of L
obtain the eigenvalues (EigVal) and the corresponding
eigenvectors (EigVect): EigVal, EigVect (v)← eigh(L)
   indices ← argsort (EigVal)[: k]
   U← EigVect [: indices]

># Step 3: clustering of the rows of U is obtained through k-means
 clustering.
   kmeans ← KMeans(n_clusters=k)
   initial_partition ← kmeans. fit_predict(U)

># Step 4: use the Louvain Algorithm to the initial partition
partition ← best_partition (G, partition = initial_partition, resolution
 = gamma)

># Perform step 3 and step 4 until the process converges.
    While true do
        old_partition ← partition

># Step 5. is to conduct k-means clustering on the   rows of U.
        kmeans ← KMeans(n_clusters=k)
        initial_partition ← kmeans. fit_predict(U)

># Step 6: The Louvain Algorithm for the initial formation of a
 partition is equal to best_partition
 Louvain_Algorithm = best_partition (G, partition = initial_partition
, resolution=gamma)

        if partition == old_partition then
            break
        end if
    end while

># Output the final partition
return partition
end procedure
```

The proposed method of combining autoencoder-based deep learning of features with multiple clustering methods improves and optimizes the process of community detection in networks data. Integration in this structure guarantees well distributed and coherent neighborhoods in Louvain Partition Algorithm. The algorithm will also keep on running and the divide will keep adjusting and will only stop whenever it reaches a state that signifies a good and stable community structure. Finally, the function returns the final partition as a dictionary in which each node is paired with its community index; thus, presenting an improved and efficient community detection.

7.2 Time complexity

- **Graph autoencoder training:** The time complexity [5] is O (V_e. d^2.L). Where V_e signifies the total number of vertices in the graph, d signifies the number of hidden dimensions, and L symbolizes the number of layers in the autoencoder.
- **K-means clustering:** The worst-case complexity of time is [28] O (V_e. k. t) where k implies number of clusters and t signifies the number of iterations [28].
- **Spectral clustering:** The worst-case time complexity is calculated as O ($V_e{}^3$). due to eigen value and vector computations [30].
- **Louvain method:** The time complexity is O (E_v log V_e).

7.1 Space complexity

- **Graph autoencoder:** The space complexity[5] is O (V_e. d+d2.L).
- **Clustering algorithms:** The space complexity for K-means, Spectral Clustering, is O ($V_e{}^2$).
- **Community detection algorithms:** The space complexity for Louvain [5], O (V_e + $E_{v)}$ where 'V_e' is the number of nodes and 'E_v' is the number of edges [5].
- **Combining the complexities:** The Graph Autoencoder (GAE) training and the clustering and community detection algorithms, the overall complexities are:

  Time Complexity: O ($V_e{}^3$)
  Space Complexity: O ($V_e{}^2$)

8 Results and discussion

The research also shows that integrating feature learning through autoencoders alongside multiple clustering procedures enhances community detection remarkably. When it comes to assessing benchmark data sets, such as Karate Club and Dolphins, the method proves proficient in perceiving complex patterns/structures within the network data.

8.1 Evaluation methods for assessing algorithm performance

- **Modularity score:** Modularity measures the strength of connections within groups [6], Higher modularity value displays stronger community structure on the divisions detected, in the networks.

$$Q = \frac{1}{2E_v}\sum_{i,j}^{k_{i,j}}\left(A_{i.j} - \frac{k_i k_j}{2E_v}\right)\delta\left(c_i, c_j\right) \qquad (7)$$

where:

• $A_{i.j}$ is the actuality of the adjacency_matrix [5] of the graph,

• Here, the value k_i and k_j are the degrees of nodes.

• E_v is defined as the total of all edges of the graph .

• The nodes are arranged in several communities or groups and the particular nodes i and j belong to the communities of nodes c_i and c_j.

• $\delta\left(c_i, c_j\right)$ Kronecker [6] delta function [16] meaning it equals 1 if the indices c_i and c_j are equal and 0 if they are different [5].

- **Silhouette score:** It describes the extent of the resemblance of an object within the cluster to the other clusters[36] . It is defined as:

$$\mathrm{d} = b(m) - a(m) \qquad (8)$$

$$S(m) = \frac{d}{\max\left(a(m), b(m)\right)} \qquad (9)$$

where:

The intra-cluster distance a(m) measures the mean distance of i^{th} data point from the other points belonging to its cluster [36].The inter-cluster distance b(m) average of the shortest distance from the i^{th} point to all the points belonging to the different cluster [36]. d signifies the difference between the two distances.It must be noted that the overall silhouette coefficient is the average of s(m) for all instances [41]. Consequently, the silhouette score nearly equal to 1, testify that all the data points belong to the appropriate cluster.

- **Normalized mutual information (NMI) score:**

NMI is employed in comparison of two clustering results [27]. As:

$$NMI(M, N) = \frac{2.I(M;N)}{\mathrm{H(M)+H(N)}} \qquad (10)$$

Where

• The last is the mutual information of two clustering results M and N, which is defined as I(M;N) [27].

• Information entropies of M and N are represented by H(M) And H(N). An NMI value closer to 1 suggests that the predicted clusters are nearly identical to the actual clusters of the data.

- **Conductance:**

It is one of the measures of evaluation of the quality of the clusters or communities that are formed. For a subset of nodes S, it is defined as:

$$(S) = \frac{|\{(U,V) \in E : u \in S, v \in S|}{\min\left(\mathrm{vol(S)}, \mathrm{Vol}\left(\frac{V}{S}\right)\right)} \qquad (11)$$

where:

• |{(u,v)∈E: The desired quantity is defined as [42] |{(u,v)∈E: u∈S,v∈/S}|[3,6], the fraction of edges between the subset S and the rest of the graph.

• vol(S) [42] sum of the degree of the nodes in S.

• vol(V\S) [42] sum of elements or degree of the nodes are not in the set S.

- **Adjusted rand score (ARS):**

The Adjusted Rand Score defines the similarity of two clusterings, by taking into account all the possible pairs of samples and counting the number of pairs included in the same or different clusters of the clustering prediction and the true clustering [27]. The ARS is adjusted for the chance grouping of elements:

$$ARS = \frac{RI - Expected\ RI}{\max(RI) - Expected\ RI} \tag{12}$$

where:

RI is the Rand Index. Expected RI is the expected value of the Rand index [43] which namely shows an expected of the current structure of a set of clusters.

8.2 Comparison table

The Comparison Table of Methods for community detection focuses on the key features, benefits and the limitations of each method based on factors including the accuracy, scalability and the computational complexity of the type of clustering and the applicability to large scale graphs. It is represented by table 1.

Table 1: Comparison table of methods

S.No	Method	Approach	Evaluation Metrics (Accuracy, NMI, ARS, Precision, etc.)	Scalability	Modularity Optimization	Key Benefits	Strengths	Limitations
1	**Autoencoder + K-Means Clustering** [44]	get low-dimensional embeddings and feature learning using Autoencoder further applies K-Means for clustering	Accuracy, Macro Precision, ARS, Modularity	High	Low	Useful dimensionality reduction accompanied by uncomplicated clustering at the same time	Uncomplicated, topological, high-speed clustering in the spatially smoother dimensions.	Assumes spherical clusters, struggles with complex data
2	**Autoencoder + Agglomerative Clustering** [45]	An autoencoder is used to learn embeddings and hierarchical agglomerative clustering is performed.	accuracy, NMI, ARS. Macro-precision, Macro-recall, Macro- F1-score.	Moderate	Moderate	However, the proposed model is capable of capturing power-law distributions that represent hierarchical structures in the latent representation space.	Discovers the community relationships, which are particularly useful for hierarchical ones.	Poor scalability in large datasets, computationally expensive and sensitive to linkage criteria.
3	**Autoencoder + Louvain Algorithm** [46]	The modularity of Louvain is used to optimize the Autoencoder embeddings.	Modularity, NMI, ARS, Macro precision	High	High	Implements a modularity optimization of Louvain that improves with learned representations.	The algorithm discovery is highly scalable to large networks also Optimizes modularity,	Even when there are good embeddings Louvain may fail to capture small communities because of its greediness, Limited in very dense graphs
S.No	**Method**	**Approach**	**Evaluation Metrics (Accuracy, NMI, ARS, Precision, etc.)**	**Scalability**	**Modularity Optimization**	**Key Benefits**	**Strengths**	**Limitations**

4	**Autoencoder + Louvain + Agglomerative** [10]	Integrates embeddings of autoencoder about what has been learnt into agglomerative clustering then employ Louvain to detect the dense communities	NMI, Modularity, ARS	Moderate	High	The described method is based on hierarchical and modularity-based clustering of data and is aimed at creating a multi-scale network.	It seems to balance the global and the local structures.	Low efficiency and high demand on computation.
5	**Autoencoder + Spectral Clustering** [17]	While autoencoder is applied to learn the feature representations of nodes, spectral clustering is used to generate latent space and partition communities.	NMI, ARS, Modularity, Macro Precision, Macro Recall.	High	Moderate	More efficiently partitions latent space by the help of spectral characteristics.	Captures global structure, good for non-linear separations	Very sensitive to the total of vertices in the graph, preventively expensive on large datasets
6	**Proposed Algorithm Autoencoder + Spectral Louvain** [47]	Embeddings from the autoencoder are segmented via spectral clustering, followed by Louvain for modularity.	Modularity, NMI, ARS, Macro Precision, Conductance, Macro Recall, Accuracy	High	High	Combines the approach of the global graph structure using the spectral approach with the optimized modularity approach.	Efficient in unsupervised learning, detection of global clusters with strong modularity optimization, scalable	Spectral clustering has a problem of high computational complexity if the graph is large.
7	**Autoencoder + Louvain + K-Means** [48]	autoencoder used for embeddings, Louvain for modularity optimization, , and K-Means for improving the clustering results.	Accuracy, Modularity, Macro Precision, Average Rand Score.	Moderate	Moderate	Optimal trade-off between the optimization of modularity and the clustering.	present a fast and interpretable community detection.	Struggles with complex network structures, K-Means is known for its difficulties in identifying non-convex clusters; still, Louvain may fail to identify smaller communities.
8	**Graph Convolutional Networks (GCN)-based methods** [25], [49]	Finds node embeddings for a graph that allows for accounting for both features and relationships using graph convolutional networks.	Accuracy, NMI, ARS, Macro Precision, Macro F1-score,	Moderate	Moderate	Both nodes features and pairwise connection in graph structures are employed for enhanced clustering.	Effective with labelled data, node features, exceptional in graph-structured data and node classification.	Requires labelled data, retraining for new tasks, Onerous to compute particularly when deep models or large graph scenarios are in real-life applications.
9	**GNN-based methods** [50]	General framework for Graph Neural Networks Architecture to incorporate information of the closer neighbours.	Accuracy, Macro-Precision, Macro-Recall, ARS, Macro- F1-score NMI,	Moderate	High	Versatile containing both node and link information but suffer.	Can work with unsupervised, semi-supervised, or supervised tasks, learning from graph structures, An all-around tool and robust for a variety of graph computations.	High computational cost solution and over-smoothing issue in the deep architecture. struggle when applied to large-scale graphs consume high memory
S. No	**Method**	**Approach**	**Evaluation Metrics (Accuracy, NMI, ARS, Precision, etc.)**	**Scalability**	**Modularity Optimization**	**Key Benefits**	**Strengths**	**Limitations**

10	**Graph Attention Networks (GATs)** [51]	Utilizes attention mechanism in assigning importance of neighbours in node embedding learning.	NMI, ARS, Macro Precision, Macro F1-score	Moderate	High	Enhances the node and the edge feature representation using attention.	Introducing an effective method for node representation learning while having a precise control over the relevance of the neighbours. Captures importance of different edges	High computational complexity, A computationally intensive approach and its memory demand increases with the size of the graph.
11	**Deep Walk** [52]	(Random Walk + Deep Learning) Acquires node representations by randomly walking on the graph with the Skip-Gram approach.	NMI, ARS, Modularity, Macro Precision, Macro Recall, Accuracy	High	Moderate	General-purpose, generative node embedding that restores community information.	Adaptable to local and global graphs and easy to implement.	Sensitive to hyperparameters and sometimes fails to identify delicate community structure. Limited with dynamic graphs,
12	**Girvan-Newman** [33]	Edge Betweenness Centrality, Successively, removes edges with the highest betweenness centrality for the purpose of detecting communities.	Modularity, NMI, Macro Precision, Macro Recall,	Low	High	Highly suitable when clusters are very distinct from each other.	Highly suitable when clusters are very distinct from each other.	Highly computationally complex and not feasible for very large graphs.
13	**Node2Vec** [53]	(Biased Random Walk + Embeddings) Improves Deep Walk by incorporating biased random walks that allows enhanced exploration of the node neighbourhoods.	Accuracy, NMI, ARS, Macro Precision, Macro Recall,	High	Moderate	The exploration and exploitation approach enhances the richness of the embeddings.	Works well for identifying perfect clusters, when the data is clean.	Struggles with multi-scale community structures, Sensitive to parameter tuning. Not suitable for multi-scale community structures,
14	**Label Propagation** [54]	Propagation-based Algorithm, it sequentially transmits labels through a graph in accordance with the Neighbour information.	Accuracy, NMI, ARS, Macro Precision, Macro Recall,	High	Moderate	High efficiency for large graphs and simple to deploy at a large scale.	Highly scalable and simple.	May result in oversimplified communities because labels spread out. Unreliable with overlapping communities,
15	**Info map**[55]	Flow-based Clustering, finds communities using information theory where minimum description length of random walks are identified.	Modularity, NMI, ARS and Macro Precision	High	High	maintain both speed and accuracy for large graphs.	Suitable for large graph problem.	Preserves graph structure but may fail to identify the small communities.

8.3 Evaluation metrics

In the case of all performance measures, better performance always possesses the higher value of a measure; however, the importance of some particular measures, such as precision, recall, or overall accuracy, may be higher in one situation than in another depending on the context in which it is used or required. In this research, these metrics are employed in the autoencoder based community detection and then clustering to determine the effectiveness of the algorithm in detecting the communities in the networks. The performances of the community detection algorithms are assessed with a set of classification measures out of which some are discussed for proper evaluation of the results. The presented evaluation offers quantitative evidence of each algorithm’s utility, stability, and transferability across various datasets, which makes the conclusions falsifiable and universal [56].

8.3.1 Precision

- **Micro precision**: Subtracts the false positive count from the true positive count in all the discovered communities with the help of autoencoder and clustering method occurred at the global level for measures precision.

- **Macro precision:** Sums up the precision of every detected community without any preference for a particular community, offers an equal score or a clear picture of all the communities' performance.

- **Weighted precision:** Takes the arithmetic mean of each detected communities' precision and the number of true cases in each community is used to weigh each community

average so that larger communities have more weight than the smaller ones.

8.3.2 Recall

- **Macro recall:** Calculate the recall for each of the detected communities, and then average this measure considering all of them as similar to each other.

- **Micro recall:** Measures recall globally by summing the number of true-positives and false-negatives[57] regarding all the communities that have been distinguished by the autoencoder and the clustering algorithm.

- **Weighted recall:** Computes the average of recall of each detected community to balance their contribution to the final metric, keeping in mind bigger communities should contribute more.

8.3.3 F1-Score

- **Micro F1-Score:** The mean of micro precision and micro recall, that gives the overall estimate of the autoencoder and clustering algorithm [57].

Macro F1-Score: The combination of both macro precision and macro recall provides a balanced assessment of the algorithm's effectiveness for each discovered community [57].

Weighted F1-Score: Sums up all the F1-scores of each detected community and then divides by the number of true instances in each community to provide an estimated F1-score: [57] of the algorithm in the general framework of community detection.

8.4 Tables and figures

The following table 2 to table 5 represents the Classification Report of several clustering algorithms [56] such as K-Means [28], Spectral Clustering, agglomerative clustering [8], community detection algorithm Girvan-Newman's [33] and Louvain Algorithm [14], and their combinations of Louvain K-means, Louvain Agglomerative, Spectral Louvain All these algorithms are applied on two different data sets Zachary karate club [11] and Dolphin datasets [12].

Table 2: Evaluation metrics result - (Zachary Karate club dataset)

S.No.	Algorithms	NMI	ARS	Modularity	Conductance	Silhouette Score
1	KMeans Clustering	0.57	0.67	0.6	0.61	0.23
2	Spectral Clustering	0.68	0.77	0.65	0.59	0.23
3	Agglomerative Clustering	0.73	0.77	0.65	0.59	0.21
4	Girvan Newman	0.84	0.88	0.71	0.57	0.22
5	Louvain Algorithm	0.56	0.41	0.2	0.85	0.1
6	Louvain KMeans	0.68	0.77	0.65	0.59	0.23
7	Louvain Agglomerative	0.84	0.88	0.66	0.58	0.23
8	Spectral Louvain	1	1	0.78	0.50	0.23

Table 3: Evaluation metrics result - (Dolphin dataset)

S.No.	Algorithms	NMI	ARS	Modularity	Conductance	Silhouette Score
1	KMeans Clustering	0.33	0.19	0.62	0.6	0.2
2	Spectral Clustering	0.87	0.90	0.8	0.53	0.16
3	Agglomerative Clustering	0.21	0	0.58	0.62	0.23
4	Girvan Newman	0.88	0.88	0.81	0.52	0.15
5	Louvain Algorithm	0.55	0.4	0.62	0.71	0.08
6	Louvain KMeans	0.32	0.16	0.61	0.61	0.1
7	Louvain Agglomerative	0.32	0.16	0.61	0.61	0.1
8	Spectral Louvain	0.89	0.93	0.86	0.50	0.23

Table 4: Classification evaluation metric result (Zachary Karate Club Dataset)

S.No.	Algorithms	Accuracy	Macro Precision	Macro Recall	Macro F1 Score	Weighted Precision	Weighted Recall	Weighted F1 Score
1	KMeans	0.91	0.91	0.91	0.91	0.91	0.91	0.91
2	Spectral Clustering	0.94	0.94	0.94	0.94	0.94	0.94	0.94
3	Agglomerative Clustering	0.94	0.94	0.94	0.94	0.95	0.94	0.94
4	Girvan Newman	0.97	0.97	0.97	0.97	0.97	0.97	0.97
5	Louvain	0.24	0.22	0.1	0.14	0.53	0.24	0.33
6	Louvain KMeans	0.06	0.06	0.06	0.06	0.06	0.06	0.06
7	Louvain Agglomerative	0.97	0.97	0.97	0.97	0.97	0.97	0.97
8	Spectral Louvain	1	1	1	1	1	1	1

Table 5: Classification evaluation metric result (Dolphin Dataset)

S. No	Algorithms	Accuracy	Macro Precision	Macro Recall	Macro F1 Score	Weighted Precision	Weighted Recall	Weighted F1 Score
1	KMeans Clustering	0.27	0.22	0.21	0.22	0.3	0.27	0.28
2	Spectral Clustering	0.02	0.01	0.02	0.02	0.01	0.02	0.01
3	Agglomerative Clustering	0.6	0.73	0.7	0.59	0.82	0.6	0.58
4	Girvan Newman	0	0	0	0	0	0	0
5	Louvain	0.02	0.01	0	0.01	0.03	0.02	0.02
6	Louvain KMeans	0.71	0.77	0.78	0.71	0.84	0.71	0.71
7	Louvain Agglomerative	0.71	0.77	0.78	0.71	0.84	0.71	0.71
8	Spectral Louvain	0.94	0.92	0.95	0.93	0.95	0.94	0.94

Details of the comparative studies that can be made using the assessment metrics for classification and community detection proposed here regarding each of the corresponding Figures (2 - 13) are stated in the caption for each of such figures. From Table 2 it can be noticed that the Louvain Agglomerative has a better performance in terms of NMI (0. 84), ARS (0. 88), and Modularity (0. 66) and has proved to be better than alternatives like K-means and Spectral Clustering. The Spectral Louvain method also yields high results with the optimum Modularity score of 0. 78 coupled with the minimum Conductance score of 0. 50 which proves that there is a better way to recognize the communities of interconnection and thereby reducing inter-community links. Table 3 showed that up to this point, the Spectral Louvain algorithm outperforms all methods by registering the highest value for NMI 0.89, ARS 0. 93 and Modularity 0. 86. This suggests a strong capability in terms of reconstructing community structure within the context of the Dolphin dataset. The low Conductance score (0.52), This shows great performance of the proposed Spectral Louvain method and validate the algorithm on the basis of the conductivity measure that demonstrates its ability, to draw sharp and distinct line between communities. Other methods such as Girvan Newman and Agglomerative Clustering are also quite efficient and accurate, however they do not have the broad characteristics of the Spectral Louvain approach, most importantly in terms of NMI and ARS indicators that determine clustering accuracy as well as stability. The results in Tables 4 and 5 compare the performance of various communities' detection algorithms across two datasets: the Zachary Karate Club and the Dolphin. Several evaluation parameters have been displayed in the tables such as Accuracy, Macro Precision, Macro Recall, Macro F1 Score, Weighted Precision, Weighted Recall, and Weighted F1 Score. Out of all the methods, Louvain K-means and Spectral Louvain performed significantly better in most of the cases, especially in the weighted metrics and Spectral Louvain touches the near perfect score of around (0.94 to 0.95) in case of Dolphin dataset, Spectral Clustering and Louvain Agglomerative are two methods, successful in optimizing different sample data sets. Based on the results obtained, Spectral Louvain with overall performance of 1. 0 i.e. 100% on all the measures evaluated has been identified as the best fit for Zachary Karate Club data set. Other algorithms like Agglomerative Clustering and Louvain Agglomerative also have good results, however Louvain alone performs poorly with metrics such as Macro F1 Score and Weighted F1 Score. K Means and GN are still quite sound but not as precise as Spectral Louvain. In given Figures 2-13,

Each algorithm is represented by its corresponding serial number as follows: Being an integer, method can be 1 for K-means Clustering, 2 for Spectral Clustering, 3 for Agglomerative Clustering, 4 for Girvan Newman, 5 for Louvain, 6 for Louvain K-means, 7 for Louvain Agglomerative, and 8 for Spectral Louvain. Thus, based on the above Figures 2-13 and from the results given in table 2-6, it can be concluded that Algorithm 8 (Spectral Louvain) performs better than other algorithms with reasonable accuracy on most of the metrics, especially for the Karate Club dataset. Yet, even in the case of the Dolphin dataset, which also yields better results when the proposed methodology is applied compared to other algorithms.

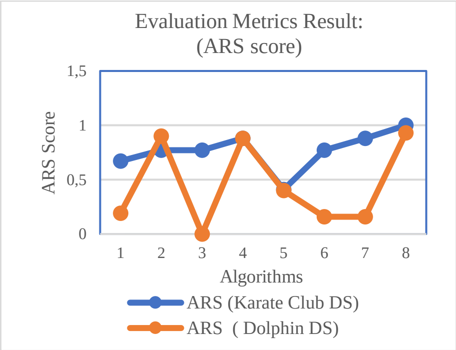

Figure 2: Evaluation Metrics Result (ARS Score)

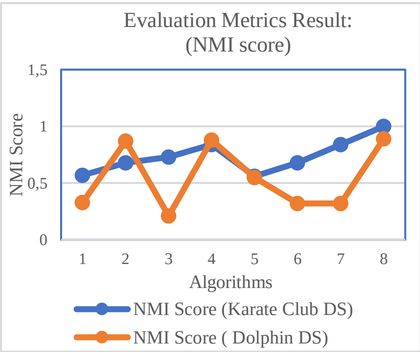

Figure 3: Evaluation Metrics Result (NMI Score)

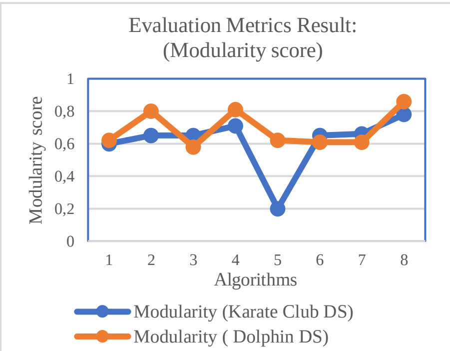

Figure 4: Evaluation Metrics Result (Modularity Score)

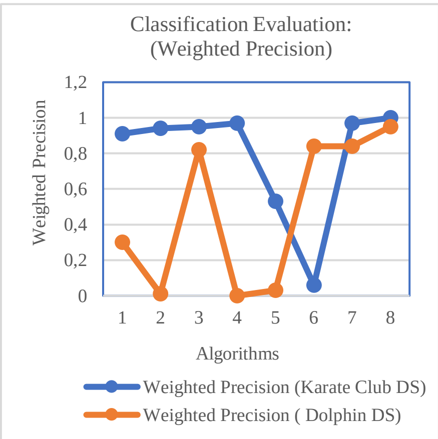

Figure 5: Evaluation Metrics Result (Conductance Score)

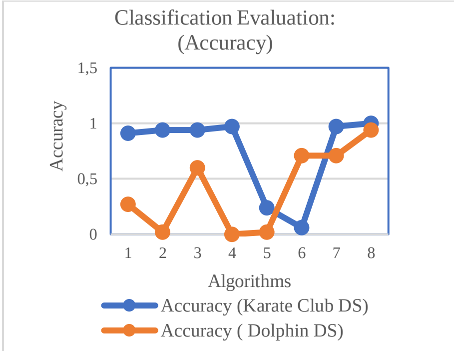

Figure 6: Classification Evaluation Metrics Result (Accuracy)

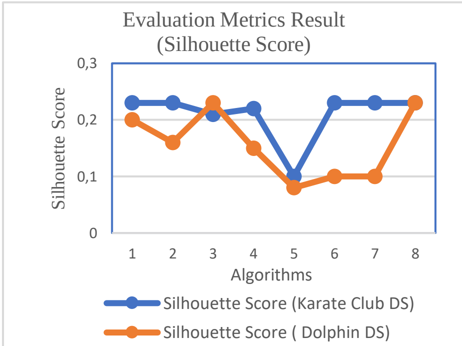

Figure 7: Evaluation Metrics Result (Silhouette Score)

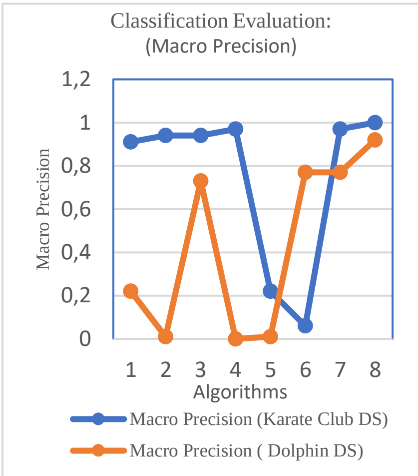

Figure 8: Classification Evaluation Metrics Result (Macro Precision)

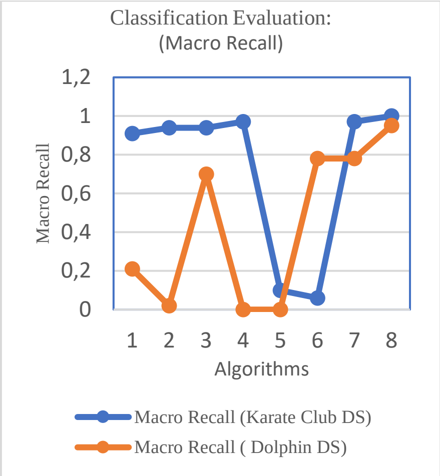

Figure 9: Classification Evaluation Metrics Result (Macro Recall)

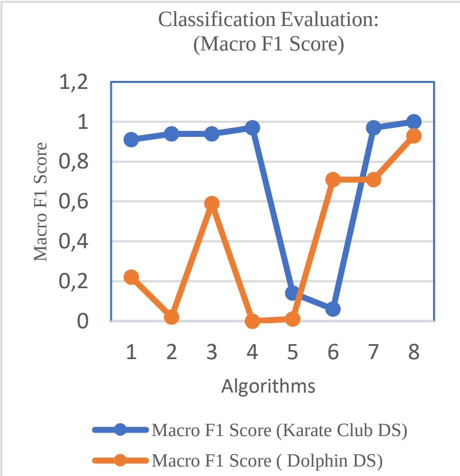

Figure 10: Classification Evaluation Metrics Result (Macro F1 Score)

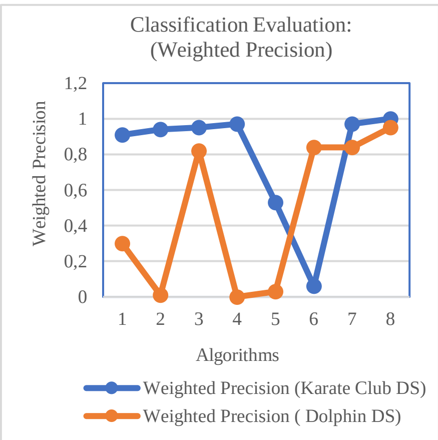

Figure 11: Classification Evaluation Metrics Result (Weighted Precision)

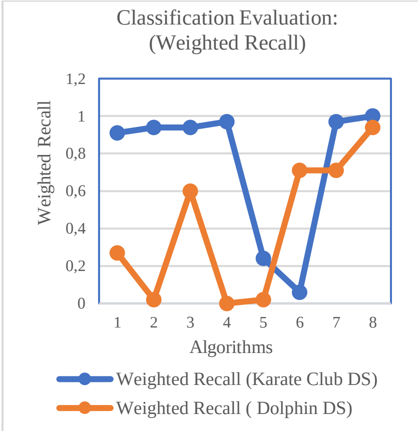

Figure 12: Classification Evaluation Metrics Result (Weighted Recall)

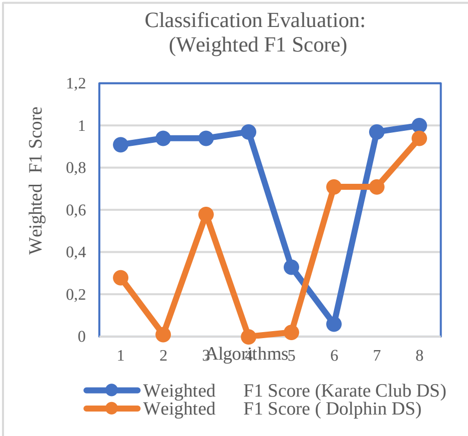

Figure 13: Classification Evaluation Metrics Result (Weighted F1 Score)

On the same kind of network, there is another algorithm called Algorithm 4 (Girvan Newman) and Algorithm 7 (Louvain Agglomerative) which is also good for the Karate Club data set, but not effective at all in case of the Dolphin data set. For instance, Algorithm 4 (Girvan Newman), it outperforms itself on the Karate club dataset, whereas it yields zero remarks on the Dolphin dataset on all sorts of measures. Moreover, Graph 5 (Louvain) algorithm and algorithm 6 (Louvain K-Means) algorithm portray significant poor performance on the Dolphin dataset.

Thus, the Algorithms Spectral Louvain based on the proposed methodology is the best overall but as with Dolphin dataset its superiority is less than in the Karate Club dataset but still better than other algorithms. Agglomerative also have good results, however Louvain alone performs poorly with metrics such as Macro F1 Score and Weighted F1 Score. K Means and GN are still quite sound but not as precise as Spectral Louvain.

9 Conclusion

This paper presents the autoencoder based Spectral Louvain Algorithm for community detection and clustering in complex networks with the performance evaluation on the Zachary karate club and Dolphin networks. The general proposal of this paper utilized autoencoders to learn nodal representations for nodes, and then used k-means clustering, agglomerative and spectral and the Louvain algorithm for the detection of communities. The results offered a few advantages: Individual feature extraction is not required in this model, does not require re-training of the network for different sizes of the network and is valuable in large networks.

The empirical results confirm that the Spectral Louvain outperforms the previous methods as it reached 100% of NMI and ARS in Zachary karate club and 78% modularity whereas in Dolphin it had 89% NMI, 93% of ARS and 86% modularity having lowest conductance. Also, during the classification assessment, it possesses a higher accuracy of 94% in the Dolphin dataset than those methods as K-means and Girvan Newman with 27% and 0% respectively. At Zachary Karate Club, Spectral Louvain got a score of 100 percent on all the evaluation criteria and hence validated its ability of detecting communities and ranking precision in various settings.

Therefore, Spectral Louvain can be stated as a successful and accurate method of community detection in large networks, free from criticisms that were made concerning earlier approaches and convenience in terms of practical implementation. This research will serve as valuable input in the development of the autoencoder in community detection making deeper learning pointed towards complex social network analysis.

10 Future work

Future research could focus on several areas to further advance the field of community detection. One area is scalability, developing methods to efficiently handle larger networks with billions of nodes and edges. Another important area is dynamic networks, extending algorithms, an important area is dynamic networks, extending algorithms to capture evolving communities in dynamic networks over time. Applying these algorithms to diverse real-world networks, such as social media, biological networks, and financial systems, will also help validate their effectiveness in practical scenarios. Additionally, refining the existing algorithms to reduce computational complexity and improve speed without compromising accuracy is another promising area for future work.

The experiments demonstrated the effectiveness of combining autoencoder-based feature learning with traditional clustering techniques for community detection. The results highlighted the potential of hybrid methods, particularly Spectral Louvain, in accurately identifying communities within complex networks.

11 Nomenclature table

Here is the expanded Nomenclature Table including additional terminology used

Abbreviation/Term	Definition
AI	Artificial Intelligence
Autoencoder	A type of artificial neural network used to learn efficient representations (for coding) of data
GCN	Graph Convolutional Networks
GNN	Graph Neural Networks
NMI	Normalized Mutual Information
ARS	Adjusted Rand Score
RI	Rand Index
SNA	Social Network Analysis
CDR	Call Detail Record
F1	F1-Score (The avg./mean of Precision and Recall)
K-means	A method of vector quantization used for clustering analysis in data mining
Spectral Clustering	A clustering algorithm which relies on the eigenvalues of a graph Laplacian matrix
Louvain Algorithm	A modularity-based community detection technique.
Modularity	A way of summarizing how well a network fits together into communities.
Precision	True positives divided by total positive predictions
Recall	True positives divided by total actual positives
Micro-Precision	Precision calculated across all instances regardless of community size
Macro-Precision	Average precision of each community, treating all equally
Weighted-Precision	Precision weighted by the size of each community
Silhouette Score	Determines the quality of mapping of an object into its cluster.
Conductance	Assesses the quality of clusters by addressing inter-cluster links.
LatntDim	Latent Space, in which raw data is located and subsequently mapped into a lower-dimensional space by an autoencoder.

Embedding	Ongoing embedding of graph nodes in a continuous space.
Adjacency Matrix	A matrix used to represent a graph, with rows and columns indicating nodes and entries indicating edges
Eigenvalue	A scalar indicating the variance captured in principal component analysis or spectral clustering
Node Embedding	A method of mapping nodes to a continuous vector space for machine learning tasks
Graph Laplacian	A matrix used in graph theory that reflects the structure of a graph

Ethical approval

To the best of the author's knowledge, this article does not incorporate any research involving humans or animals.

Funding

I would like to declare that this research did not benefit from any special funding from the agencies of the public, commercial, or non-profit sectors.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Our sincere appreciation goes to the mentors and colleagues who have selflessly given their time and expertise in the development of this research. Their advice and motivation have played an important role in the production of this paper.

References

[1] M. E. J. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, no. 2, pp. 167–256, 2003, https://doi.org/10.1137/S003614450342480

[2] S. Arora and B. Barak, "Computational Complexity: A Modern Approach," 2006. [Online]. http://www.cs.princeton.edu/theory/complexity/

[3] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proc Natl Acad Sci U S A*, vol. 99, no. 12, pp. 7821–7826, Jun. 2002, https://doi.org/10.1073/pnas.122653799

[4] **S. Fortunato,** "Community detection in graphs," *Physics Reports,* vol. 486, no. 3–5, pp. 75–174, 2010. https://doi.org/10.1016/j.physrep.2009.11.002

[5] A. Guha and G. Student, "Organization Network Analysis & Community Detection of Graphs; Organization Network Analysis & Community Detection of Graphs." [Online]. https://www.ijert.org/

[6] T. B. Dwi Putra Aditama and A. SN, "Determining Community Structure and Modularity in Social Network using Genetic Algorithm," *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 14, no. 3, 2020, https://jurnal.ugm.ac.id/ijccs/article/view/57834

[7] M. Alfaqeeh and D. B. Skillicorn, "Community detection in social networks by spectral embedding of typed graphs," *Soc Netw Anal Min*, vol. 14, no. 1, pp. 1–26, Dec. 2024, https://link.springer.com/article/10.1007/s13278-023-01172-y

[8] R. J. Gil-García, J. M. Badía-Contelles, and A. Pons-Porrata, "A General Framework for Agglomerative Hierarchical Clustering Algorithms.", https://ieeexplore.ieee.org/document/1699269

[9] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science (1979)*, vol. 313, no. 5786, pp. 504–507, Jul. 2006, https://www.science.org/doi/10.1126/science.1127647

[10] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature 2015 521:7553*, vol. 521, no. 7553, pp. 436–444, May 2015, doi:10.1038/nature14539. https://www.nature.com/articles/nature14539

[11] "KarateClubDataset — DGL 1.1.3 documentation." Accessed: Dec. 05, 2023. https://docs.dgl.ai/en/1.1.x/generated/dgl.data.KarateClubDataset.html

[12] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, "The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations," *Behav Ecol Sociobiol*, vol. 54, no. 4, pp. 396–405, Sep. 2003 https://link.springer.com/article/10.1007/s00265-003-0651-y

[13] M. E. J. Newman, "Spectral methods for network community detection and graph partitioning," Jul. 2013, : https://doi.org/10.1103/PhysRevE.88.042822

[14] P. De Meo, E. Ferrara, G. Fiumara, and A. Provetti, "Generalized Louvain Method for Community Detection in Large Networks," Aug. 2011, https://doi.org/10.1109/ISDA.2011.6121636

[15] J. Zhang, Z. Ma, Q. Sun, and J. Yan, "Research Review on Algorithms of Community Detection in Complex Networks," *J Phys Conf Ser*, vol. 1069, no. 1, p. 012124, Aug. 2018, doi: . https://doi.org/10.1088/1742-6596/1069/1/012124

[16] M. E. J. Newman, "Modularity and community structure in networks," *Proc Natl Acad Sci U S A*, vol. 103, no. 23, pp. 8577–8582, Jun. 2006, doi: https://doi.org/10.1073/pnas.0601602103

[17] U. Von Luxburg, “A tutorial on spectral clustering,” *Stat Comput*, vol. 17, no. 4, pp. 395–416, Dec. 2007, doi: https://doi.org/10.1007/s11222-007-9033-z

[18] “Jain, A. K., & Dubes, R. C. (1988). Algorithms for Clustering Data. Upper Saddle River, NJ Prentice-Hall, Inc. - References - Scientific Research Publishing.” Accessed: Aug. 17, 2024. https://www.scirp.org/reference/referencespapers?referenceid=2060572

[19] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, Oct. 2008, https://doi.org/10.1088/1742-5468/2008/10/P10008

[20] D. Jin et al., “Adaptive Local Modularity Learning for Efficient Multilayer Graph Clustering,” IEEE Transactions on Signal Processing, Volume 72, Issue,20244446 pages, ISSN:1053-587X, April. 2024, https://dl.acm.org/doi/10.1109/TSP.2024.3385654

[21] T. N. Kipf and M. Welling, “Semi-Supervised Classification with Graph Convolutional Networks,” Sep. 2016, [Online]. https://arxiv.org/abs/1609.02907

[22] H. Zhang, J. Yang, B. Zhang, Y. Tang, W. Du, and B. Wen, “Enhancing generalized spectral clustering with embedding Laplacian graph regularization,” *CAAI Trans Intell Technol*, Mar. 2024, https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/cit2.12308

[23] R. Bhattacharya, N. K. Nagwani, and S. Tripathi, “A community detection model using node embedding approach and graph convolutional network with clustering technique,” *Decision Analytics Journal*, vol. 9, p. 100362, Dec. 2023, https://www.sciencedirect.com/science/article/pii/S2772662223002023

[24] X. Su *et al.*, “A Comprehensive Survey on Community Detection with Deep Learning,” *IEEE Trans Neural Netw Learn Syst*, vol. 35, no. 4, pp. 4682–4702, May 2021, https://doi.org/10.1109/TNNLS.2021.3137396

[25] A. Apicella, F. Isgrò, A. Pollastro, and R. Prevete, “Adaptive filters in Graph Convolutional Neural Networks,” *Pattern Recognit*, vol. 144, p. 109867, Dec. 2023, https://www.sciencedirect.com/science/article/pii/S0031320323005654?via%3Dihub

[26] X. Li, X. Zhen, X. Qi, H. Han, L. Zhang, and Z. Han, “Dynamic community detection based on graph convolutional networks and contrastive learning,” *CSF*, vol. 176, p. 114157, Nov.2023 https://doi.org/10.1016/j.chaos.2023.114157

[27] A. Amelio and C. Pizzuti, “Is normalized mutual information a fair measure for comparing community detection methods?,” in *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2015*, https://doi.org/10.1145/2808797.2809344

[28] Y. Li and H. Wu, “A Clustering Method Based on K-Means Algorithm,” *Phys Procedia*, vol. 25, pp. 1104–1109, 2012, https://www.sciencedirect.com/science/article/pii/S1875389212006220

[29] K. P. Sinaga and M. S. Yang, “Unsupervised K-means clustering algorithm,” *IEEE Access*, vol. 8, 2020, https://ieeexplore.ieee.org/document/9072123

[30] P. Macgregor, “Fast and Simple Spectral Clustering in Theory and Practice,” Oct. 2023, [Online]. http://arxiv.org/abs/2310.10939

[31] H. Gao, J. Jiang, L. She, and Y. Fu, “A new agglomerative hierarchical clustering algorithm implementation based on the map reduce framework,” *International Journal of Digital Content Technology and its Applications*, vol. 4, no. 3, 2010, https://doi.org/10.4156/jdcta.vol4.issue3.9

[32] T. Li, A. Rezaeipanah, and E. S. M. Tag El Din, “An ensemble agglomerative hierarchical clustering algorithm based on clusters clustering technique and the novel similarity measurement,” *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, pp. 3828–3842, Jun. 2022, https://doi.org/10.1016/j.jksuci.2022.04.010

[33] M. E. J. Newman and M. Girvan, “Finding and evaluating community structure in networks,” *Phys Rev E*, vol. 69, no. 2, p. 026113, Feb. 2004, https://doi.org/10.1103/PhysRevE.69.026113

[34] B. S. Rees and K. B. Gallagher, “Detecting overlapping communities in complex networks using swarm intelligence for multi-threaded label propagation,” in *Studies in Computational Intelligence*, Springer Verlag, 2013, pp. 111–119. doi: https://doi.org/10.1007/978-3-642-30287-9_12

[35] H. Jin, W. Yu, and S. J. Li, “A clustering algorithm for determining community structure in complex networks,” *Physica A: Statistical Mechanics and its Applications*, vol. 492, 2018, https://ideas.repec.org/a/eee/phsmap/v492y2018icp980-993.html

[36] P. J. Rousseeuw, “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,” *J Comput Appl Math*, vol. 20, no. C, pp. 53–65, 1987, https://doi.org/10.1016/0377-0427(87)90125-7

[37] “karate | Social Networks | Network Data Repository.” Accessed: Dec. 05, 2023. [Online]. Available: https://networkrepository.com/soc-karate.php

[38] Z. Huang, W. Xu, and X. Zhuo, “Community-CL: An Enhanced Community Detection Algorithm Based on Contrastive Learning,” *Entropy 2023, Vol. 25, Page 864*, vol. 25, no. 6, p. 864, May 2023, https://doi.org/10.3390/e25060864

[39] S. Fortunato, "Community detection in graphs," Jun. 2009, doi: https://doi.org/10.1016/j.physrep.2009.11.002
[40] P. Bedi and C. Sharma, "Community detection in social networks," *Wiley Interdiscip Rev Data Min Knowl Discov*, vol. 6, no. 3, pp. 115–135, May 2016, https://doi.org/10.1002/widm.1178
[41] T. Ameen, W. Rashid, and A. Ahmad, "Estimation of vehicular delay in presence of illegally crossing pedestrians and determination of LOS using cluster analysis at midblock sections of urban roads," *Innovative Infrastructure Solutions*, vol. 8, no. 1, 2023, https://doi.org/10.1007/s41062-022-00994-7
[42] P. Basuchowdhuri, S. Sikdar, V. Nagarajan, K. Mishra, S. Gupta, and S. Majumder, "Fast detection of community structures using graph traversal in social networks," *Knowl Inf Syst*, vol. 59, no. 1, pp. 1–31, Apr. 2019, https://doi.org/10.1007/s10115-018-1209-7
[43] L. Hubert and P. Arabie, "Comparing partitions," *J Classif*, vol. 2, no. 1, pp. 193–218, Dec. 1985, https://doi.org/10.1007/bf01908075
[44] K. Berahmand, F. Daneshfar, E. S. Salehi, Y. Li, and Y. Xu, "Autoencoders and their applications in machine learning: a survey," *Artificial Intelligence Review 2024 57:2*, vol. 57, no. 2, pp. 1–52, Feb. 2024, doi: https://doi.org/10.1007/s10462-023-10662-6
[45] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, https://doi.org/10.1038/nature14539
[46] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, Oct. 2008, doi:10.1088/1742-5468/2008/10/P10008
[47] X. Yang, C. Deng, F. Zheng, J. Yan, and W. Liu, "Deep Spectral Clustering using Dual Autoencoder Network". https://arxiv.org/abs/1904.13113
[48] Z. Chen, J. Bruna, and L. Li, "Supervised Community Detection with Line Graph Neural Networks," *7th International Conference on Learning Representations, ICLR 2019*, May 2017, Accessed: Oct. 10, 2024. https://arxiv.org/abs/1705.08415
[49] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, Sep. 2016, Accessed: Aug. 14, 2024. https://arxiv.org/abs/1609.02907
[50] J. Zhou *et al.*, "Graph neural networks: A review of methods and applications," 2021, https://doi.org/10.1016/j.aiopen.2021.01.001
[51] P. Veličković, A. Casanova, P. Liò, G. Cucurull, A. Romero, and Y. Bengio, "Graph Attention Networks," *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, Oct. 2017, https://doi.org/10.1007/978-3-031-01587-8_7
[52] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 701–710, 2014, https://doi.org/10.1145/2623330.2623732
[53] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 13-17-August-2016, pp. 855–864, Aug. 2016, https://doi.org/10.1145/2939672.2939754
[54] B. Laassem, A. Idarrou, L. Boujlaleb, and M. Iggane, "Label propagation algorithm for community detection based on Coulomb's law," *Physica A: Statistical Mechanics and its Applications*, vol. 593, p. 126881, May 2022, https://doi.org/10.1016/j.physa.2022.126881
[55] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proc Natl Acad Sci U S A*, vol. 105, no. 4, pp. 1118–1123, Jul. 2007, https://doi.org/10.1073/pnas.0706851105
[56] S. Emmons, S. Kobourov, M. Gallant, and K. Börner, "Analysis of network clustering algorithms and cluster quality metrics at scale," *PLoS One*, vol. 11, no. 7, Jul. 2016, https://doi.org/10.1371/journal.pone.0159161
[57] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf Process Manag*, vol. 45, no. 4, pp. 427–437, Jul. 2009, https://doi.org/10.1016/j.ipm.2009.03.002