CLAID: A Unified Social Network Analysis Framework for Community-Level Anomaly and Influencer Detection

Ramya S
Department of Computer Technology
Madras Institute of Technology
Anna University
2021503545@student.annauniv.edu

Rupesh A
Department of Computer Technology
Madras Institute of Technology
Anna University
2021503549@student.annauniv.edu

Akilan k
Department of Computer Technology
Madras Institute of Technology
Anna University
2021503719@student.annauniv.edu

***Abstract*—Social networks are complex structures formed by connections between individuals. The complex nature of social networks, characterized by complex connections among individuals, necessitates effective identification of communities for various applications.Network communities are subgroups formed by individuals with shared connections, not all participants in a social network are genuine. This necessitates anomaly detection, which aims to identify suspicious nodes or activities like fake accounts or individuals spreading misinformation within the network.Furthermore, within each community, some users are more influential than others. Identifying influential users helps us understand the dynamics within a community.However, existing deep learning-based methods for anomaly detection in social networks, are often reliant on graph auto encoders and exhibit limitations in fully leveraging network data richness, leading to sub-optimal performance.Many existing approaches suffer from scalability issues when dealing with large-scale networks, leading to high computational demands and sensitivity to parameter settings. Addressing these challenges, a novel framework CLAID is introduced which integrates community detection, anomaly detection, and influential user identification and potentially reducing computational complexity and improving scalability.**

***Index Terms*—Community Detection, Anomaly Detection,Louvain Modualrity,Influencer User,Isolation Forest.**

I. Introduction

SOCIAL network is a digital platform or service that allows individuals to connect and interact with each other online. These platforms facilitate communication, sharing of information, and forming of virtual communities based on common interests, relationships, or activities.

Social networks serve as channels for both personal and professional networking, enabling individuals to stay in touch with friends, family, colleagues, and acquaintances, as well as to meet new people with similar interests. It also represents complex webs of connections between individuals, forming complex structures that necessitate effective identification of communities for various applications. These communities, or subgroups within the network, are comprised of individuals who share connections [13]. However, not all participants within these networks are genuine [15], with some engaging in suspicious activities such as creating fake accounts or spreading misinformation. Detecting such anomalies becomes crucial in maintaining the integrity and functionality of the network. Moreover, within each community, there exists a hierarchy of influence, where certain users have more power and impact than others. Understanding the dynamics within these communities requires the identification of influential users [16].

While social networks are rich sources of connections and interactions, effectively identifying communities or subgroups within these networks remains a challenge. Existing methods may struggle to adapt to the evolving nature of social networks and may not fully capture the diversity of community structures [11]. Detecting anomalies is crucial for maintaining the integrity and functionality of the network. However, traditional anomaly detection methods may not be robust enough to effectively identify and mitigate these threats. [14]Within each community in a social network, there exists a hierarchy of influence, where certain users wield more power and impact than others. Understanding the dynamics within these communities requires the accurate identification of influential users.

Community detection based on Network Representation learning [1], [3] uses the node2vec with random walks to traverse the network and capture node relationships. It aims to learn node representations by maximizing the co-occurrence probability of neighbouring nodes.Non-Negative Matrix Factorization (NMF) based community detection [5], [11] in attributed networks is done by building a special network that combines structural connections and node attributes. To analyse network, it iteratively refines a similar matrix and a community membership assignment for each node. Granular Computing-Based Community Detection builds a special network [6] that incorporates both structural connections and node attributes. It calculates Object Community Factor (OCF) to considers how similar a node's attributes are to other nodes in the network, taking into account multiple attributes and their importance. The potential challenge that exists in the approach is the ineffectiveness in incorporating the diverse community structures present in real-world networks. Additionally, the accuracy of community labels assigned to vertices might be

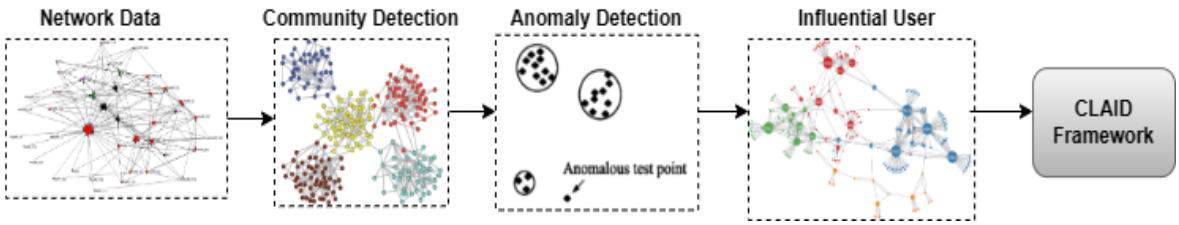

Fig. 1: CLAID framework

affected by noise or inconsistencies within the network data. Moreover, the scalability of these methods might become an issue when dealing with large-scale networks.

Influential users within the communities are individuals who hold significant power over others within their community or across the platform. Various dynamic social network analysis approaches [16], including heuristic, greedy, hybrid, incremental, sliding window, and interval methods, aim to detect influential nodes by considering temporal changes in network structure and interactions, leveraging heuristics, community detection, and incremental updates for improved accuracy in evolving networks. Change Point Detection [2] done by first capturing the structure of the network at different points in time (snapshots) and assigning a score to each snapshot that reflects both its local and global properties. Though it identifies the shifts in the data structures, it lacks in accuracy in detecting change points. Furthermore, the computational complexity of analysing multiple snapshots and detecting changes over time pose challenges, particularly for large-scale networks, leading to scalability issues.

Anomaly detection within community detection involves the identification of suspicious or unusual patterns, nodes, or behaviors within network communities.CoLA [4] addresses the challenge of identifying anomalies by looking at the relationships between nodes. It achieves this by introducing a contrastive learning approach that analyzes pairs of a target node and its local subgraph.Despite their effectiveness, CoLA's reliance on contrastive learning might face challenges in distinguishing anomalies accurately in complex network structures. Autoencoders(AE) [9] neural network techniques reconstruct data by assigning higher anomaly scores to data that the model has difficulty reconstructing. Similarly, the performance of Autoencoders in anomaly detection might be affected by the complexity and diversity of data, leading to suboptimal reconstruction results.A deep learning model named Hybrid-Order Graph Attention Network (HO-GAT) [8] captures intricate relationships between nodes and subgraphs within the network.But it has very limited focus on the anomaly nodes.Eigenvector Admixture based Sparse component Separation (EAS) approach [10] uses Independent Component Analysis (ICA) to extract anomalies while using sparsity constraints to eliminate noise and focus on the most significant anomalies.Though, these methods excels in detecting the anomalies it struggles to isolate those anomalies from the non-anomalous nodes which hinder there accuracy and performance impromacy.

In order to solve the above challenges, a novel framework CLAID is proposed.The CLAID framework integrates community, anomaly detection, and influential user identification in social networks, enhancing understanding and performance. Its scalability and efficiency streamline processes, reducing computational costs and time, especially for large-scale networks. CLAID employs the Louvain algorithm for efficient community detection and an OSINT-based algorithm to isolate anomaly nodes and betweeness centrality for influential user detection, ensuring better management of social networks. To summarize, the main contributions are as follows:

- We propose the CLAID framework that integrates community detection, anomaly detection, and influential user identification into a unified approach. Rather than conducting separate analyses for each task, CLAID combines these functionalities to enhance understanding and performance across all domains.
- The CLAID framework utilizes the Louvain algorithm for community detection, known for its efficiency in identifying community structures within networks while reducing computational complexity.
- CLAID incorporates an efficient isolation forest algorithm within the OSINT framework to isolate anomaly nodes within communities, facilitating the identification of fake profiles or misinformation spreaders.
- CLAID identifies influential users through betweenness centrality analysis, which measures the extent to which a node lies on the shortest paths between other nodes in the network. By pinpointing nodes with high betweenness centrality, CLAID accurately identifies influential users who serve as key bridges connecting disparate parts of the network.

The rest of this article is organized as follows. Section II reviews the related works. Section III presents the introduction of methodology and implementation details for each component of the proposed framework. In Section IV, the experimental evaluation and analysis are presented. Finally, Section V draws the conclusion.

II. Related Works

A social network's structure consists of users and relationships. Some relationships are dense, while others are sparse. A tightly connected part can be seen as a community, and overlapping communities have common nodes. For a given

social network, community detection can be seen as a process of clustering.

The KroMFac framework [17] for detecting overlapping communities in incomplete networks by characterizing influential nodes and using regularized nonnegative matrix factorization (NMF). It validates the approach using normalized mutual information (NMI) to demonstrate its effectiveness over baseline methods.But it lacks accuracy when influential nodes are not well-defined or when the network data is significantly incomplete.Community Detection in Partially Observable Social Networks [18]integrates group homophily and individual personality of topics to model network communities. It aims to explain the generation mechanisms of intracommunity and intercommunity links by analyzing real networks and identifying nodes with distinctive personality who are more active across communities. However, it may not capture all the community interactions and link generation, as it is based on a set of assumptions and mathematical models.Integrating Group Homophily and Individual Personality of Topics Can Better Model Network Communities [19], that integrates group homophily and individual personality of topics. It explains link generation in communities by identifying active nodes with distinctive personality. However, it may not capture all community and assumes active nodes have distinctive personalities.Graph Community Infomax (GCI) [21] for graph representation learning. It uses a graph convolutional autoencoder to learn node embeddings, applies k-means clustering to identify communities, and uses a discriminator module to maximize mutual information between community and node representations. However, it assumes nodes with distinctive personalities are more active, and doesn't consider community evolution,

Anomaly Detection from Diabetes Similarity Graphs using Community Detection [20] presents a methodology for detecting anomalous blood glucose levels in diabetes patients. It begins by transforming clinical and non-clinical features into binary sets to compute pairwise similarity scores. These scores are used to construct a diabetes similarity graph, where vertices represent individuals and edges are drawn based on a similarity threshold. The Louvain algorithm is then applied to identify natural groupings within the graph.Community-Aware Attributed Graph Anomaly Detection [21] integrates key components for effective anomaly detection on attributed graphs. It employs a tailored deep graph convolutional network (tGCN) to process both graph structure and node attributes. Community detection is facilitated through an autoencoder, capturing community-specific representations. An anomaly detection module reconstructs the graph's topology and nodal attributes from anomaly-aware node representations.Though, these methods excels in detecting the anomalies it struggles to isolate those anomalies from the non-anomalous nodes which hinder there accuracy and performance impromacy.

The Multiobjective Evolutionary Algorithm (MOEA) methodology [22] is based on the principles of natural selection to solve optimization problems. It is particularly useful in scenarios where two or more potentially conflicting objectives need to be optimized simultaneously, without reducing the objectives to a single one through a weighted sum. Techniques such as NSGA-II, SMS-EMOA, and MOEA/D are state-of-the-art in multiobjective optimization and represent key paradigms in evolutionary multiobjective algorithm design. These techniques have been used as alternatives to traditional clustering and single-objective evolutionary methods, allowing for the simultaneous optimization of multiple objectives.But the MOEAs can be computationally expensive, especially for problems with a large number of objectives or a high-dimensional search space.

TABLE I: Related Works Summary

Methods	Community Detection	Influential User	Anomaly Detection
Network Representation Learning [1],[3],[17]	✓	✓	
Non-Negative Matrix Factorization [5],[11]	✓		
Granular Computing [6]	✓		
Change Point Detection [2]	✓	✓	
Contrastive Self-Supervised Learning [4]	✓		✓
Auto-encoders [9]			✓
Motif-Augmented Network [8]	✓		✓
Hyperspectral Imaging [10]	✓	-	✓
Dynamic Approaches[16]	✓	✓	
graph representation learning[21]	✓		
CLAID(Community Level Anomaly and influential user Detection)	✓	✓	✓

To overcome this, CLAID (Community-Level Anomaly and Influencer Detection) framework is proposed. It overcomes by first identifying communities within the network. This allows for focused analysis on anomalies within each community, potentially simplifying their isolation. Additionally, CLAID leverages the network structure itself for detection, reducing reliance on methods like graph autoencoders and potentially leading to better anomaly identification. Finally, by focusing on communities, CLAID might require less computation for both community detection and anomaly detection, improving its scalability for massive social networks.

III. Proposed System

Section A deals with the proposed architecture and the flowchart depicting it. Section B deals with the proposed algorithm.

Our proposed framework CLAID addresses the challenge of spotting influential users and anomalies within social networks. The proposed architecture employs a machine learning with deep learning approach for this purpose.

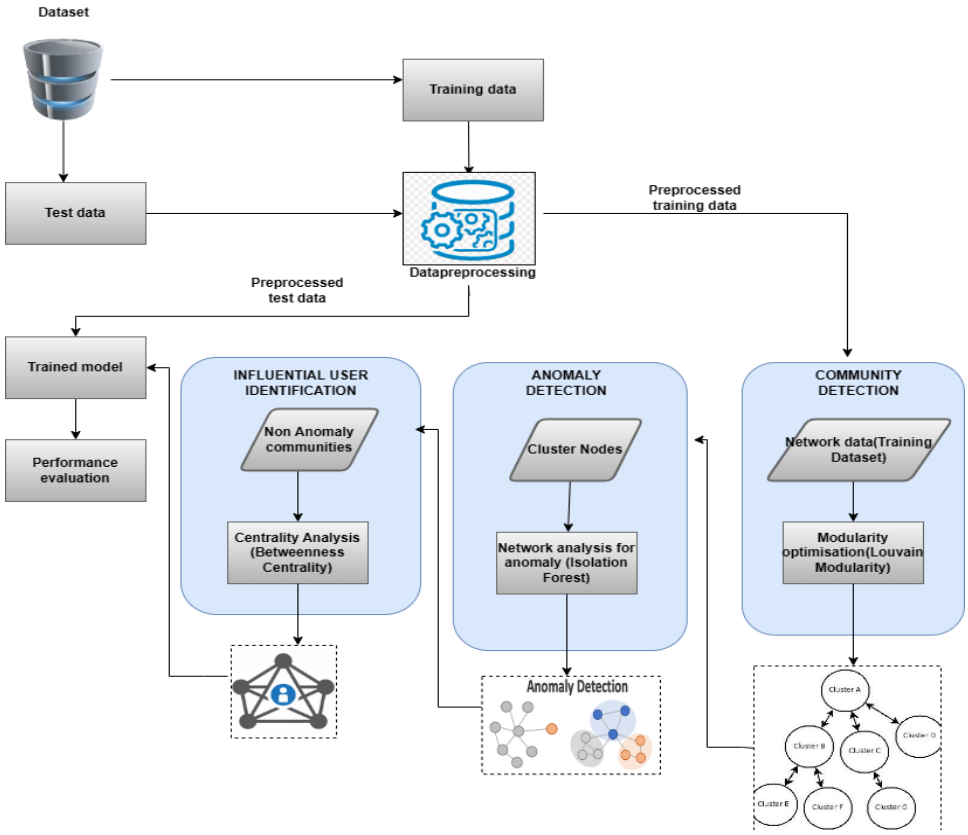

Fig. 2: Architecture diagram of CLAID Framework

First, data is collected from the social network and divided into training and testing sets. The training data undergoes preprocessing, a step that often involves normalization. Normalization ensures all features are on a similar scale, while feature selection helps identify the most relevant data points for the machine learning model. Once preprocessed, the training data is fed into the model, which trains it to recognize patterns of community formation, influential users and anomalous behaviours.

After training, the model is deployed on the testing data to evaluate its effectiveness in identifying influential users and anomalies within the social network. Additionally, the proposed architecture suggests a network analysis component, where communities within the network are identified. Modularity optimization, a technique is employed to increase the quality of these communities. Finally, the model's performance is assessed using metrics that measure how well it performed in identifying communities, influential users and anomalies. This comprehensive approach allows CLAID to not only detect influential users and anomalies but also understand the social network's community structure, providing valuable insights into user behaviour and content.

A. Community Detection

Community detection in social networks is a fundamental task of identifying the underlying structure of interconnected nodes. It involves identifying clusters of nodes that exhibit strong internal connections while having fewer connections with nodes outside the cluster. It represent subsets of nodes within the network that share common characteristics or interests.

The proposed Louvain Modularity algorithm addresses the challenge of community detection in social networks, where the network is represented as an adjacency matrix and the goal is to partition the nodes into distinct communities. The algorithm begins by initializing each node as its own community, setting up the groundwork for subsequent analysis. It iteratively evaluates the modularity of potential community structures and reassigns nodes to optimize the overall modularity score. It calculates the degree and expected degree of each node, identifies the best community for each node based on modularity gain, and updates the community assignments accordingly. This iterative process continues until no further improvement in modularity is observed. Finally, the algorithm returns a list of communities, where each element corresponds to the community ID of the respective node. By iteratively refining the community structure based on modularity optimization, the algorithm offers a systematic approach to uncovering meaningful community structures within social networks, facilitating deeper understanding in social networks and user interactions.

1) Node Connectivity Measures: Node degree measures the connectivity of a node within a network by counting the number of edges directly connected to it. It indicates the level of interaction or influence that the node has within its immediate neighborhood. Higher node degree values signify nodes with more extensive connections or interactions in the network topology.

$$k_i \leftarrow \sum_j A[i] \tag{1}$$

where k is the node degree of each Node in the adjacency matrix A.

The expected degree of a node in a network represents the average number of edges it is expected to be connected in the total number of edges in the network. It is computed by scaling the sum of connections in each row of the adjacency matrix by the total number of edges and normalizing by the total sum of connections in the adjacency matrix A. This measure serves as a baseline expectation for the level of connectivity that node i should exhibit within the network.

$$e_i \leftarrow \frac{m \times \sum_j A[i]}{\sum_{i,j} A} \tag{2}$$

where m is the total number of edges in the network and A[i] is the ith node in the adjacency matrix A.

Modularity gain quantifies the improvement in the quality of community structure by relocating a node from its existing community to a neighboring one within a network. The modularity gain is evaluated for each neighboring community C of node i, comparing it against the current best modularity gain. If the current modularity gain exceeds the current best gain, the algorithm updates the best gain and reassigns node to community C.

$$Q_i \leftarrow k_i - e_i \tag{3}$$

where k represents the node degree and e denotes the expected degree.

B. Anomaly Detection

Anomaly detection plays a critical role in identifying suspicious behavior within social networks, helping to maintain their integrity and functionality. CLAID employs an efficient isolation forest algorithm for anomaly detection, which is capable of identifying anomalous communities based on their deviation from normal behavior.

The anomaly detection process begins by inputting a set of communities into the isolation forest algorithm. These communities represent interconnected groups of individuals within the social network. The algorithm then proceeds to partition each community into anomalous and non-anomalous subsets, assigning anomaly scores to each community based on its deviation from the norm.The algorithm operates by recursively splitting the data into subsets using randomly chosen features and thresholds. This process continues until each subset contains only a single data point or reaches the specified maximum depth. An average path length is calculated for each data point within the forest of decision trees, representing the number of edges traversed to reach the data point's terminal node.

Algorithm 1 Compute Node and Edge Connection

procedure NODECONNECTION(C, i) ▷ Compute Node Connection
  $Sigma\_in\_C \leftarrow 0$
  for each node j in community C **do**
    if there is an edge between node i and node j **then**
      Increment $Sigma\_in\_C$ by 1
    end if
  end for
  return $Sigma\_in\_C$
end procedure
procedure EDGECONNECTION(G) ▷ Compute Edge Connection
  $Sigma\_tot \leftarrow 0$
  for each node i in graph G **do**
    Calculate the degree k_i of node i
    Add k_i to $Sigma\_tot$
  end for
  return $Sigma\_tot$
end procedure

Finally, the anomaly score for each community is calculated by averaging the path lengths across all decision trees in the forest. This score is normalized by dividing it by the maximum average path length expected for a data point in the dataset. Communities with higher anomaly scores are considered more anomalous, indicating a higher likelihood of containing suspicious activity.

1) Threshold calculation: The threshold for splitting a node in the decision tree is determined by randomly selecting a value within the range of the feature's minimum $\min(\cdot)$ and maximum $\max(\cdot)$ values in the subsample dataset $X_{\text{subsample}}$. Mathematically, it can be expressed as:

$$\begin{aligned}\text{threshold} = \text{random.uniform}(&X_{\text{subsample}}[:, f].\min(), \\ &X_{\text{subsample}}[:, f].\max())\end{aligned} \tag{4}$$

This random selection ensures a diverse split in the decision tree construction process, contributing to the robustness of the algorithm in handling various datasets and features.

2) Average path length calculation: The average path length average_path_length for a data point within a decision tree is computed by summing the number of edges traversed from the root node to the terminal node where the data point resides. It provides a measure of the data point's isolation within the tree structure. Mathematically, it can be expressed as:

$$\text{average\_path\_length}(x, \text{tree}) = \sum_{\text{node}(x,\text{tree})}^{1} 1 \tag{5}$$

Algorithm 2 Community Detection

procedure COMMUNITYDETECTION(A, m)
  Initialize communities: $C[i] \leftarrow i$, $\forall i$
  while modularity improvement occurs **do**
    for each node i **do**
      Calculate node degree.
      Calculate expected degree.
      Initialize best community: $best\_community \leftarrow C[i]$
      Initialize best modularity gain: $best\_modularity\_gain \leftarrow 0$
      for each neighboring community c of node i **do**
        Hypothetically move node i to community c
        Update degree and expected degree for node i as if it belonged to community c
        Calculate modularity contribution of node i in community c: $Q_i \leftarrow$ ModularityContribution $(i, c, A,$ updated degree, updated expected degree$)$
        if $Q_i > best\_modularity\_gain$ **then**
          Update best community: $best\_community \leftarrow c$
          Update best modularity gain: $best\_modularity\_gain \leftarrow Q_i$
        end if
      end for
      if $best\_modularity\_gain > 0$ **then**
        Update community ID of node i: $C[i] \leftarrow best\_community$
        Update degree and expected degree for node i in its new community
      end if
    end for
  end while
  Return C
end procedure

where node(x, tree) represents the path from the root node to the terminal node where the data point x resides.

3) Normalization of anomaly scores: Anomaly scores, representing the deviation of communities from normal behavior, are normalized by dividing them by the maximum average path length expected for a data point in the dataset. This normalization ensures that anomaly scores are comparable across different datasets and tree structures.Mathematically, it can be expressed as:

$$\text{Anomaly\_scores} = \frac{\text{anomaly\_scores}}{\text{avg\_path\_length\_max\_depth}} \tag{6}$$

where node(x, tree) represents the path from the root node to the terminal node where the data point x resides.

C. Influential User Detection

In social networks, identifying influential users within communities is crucial for understanding the dynamics and structure of the network. CLAID employs a robust algorithm for influential user detection, which aims to pinpoint individuals who wield significant influence within their respective communities.

The influential user detection process begins by inputting non-anomalous communities into the algorithm. These communities represent interconnected groups of individuals within the social network. The algorithm then iterates through each community, calculating the betweenness centrality measure for each node to determine its influence within the community.The algorithm computes the shortest paths between all pairs of nodes within the community graph, utilizing a breadth-first search approach. It then calculates various metrics, including dependency and sigma, to quantify the influence of each node. The betweenness centrality of each node is subsequently determined based on these metrics, representing its overall influence within the community.

Finally, the algorithm identifies the most influential user within each community based on their betweenness centrality scores and compiles a list of influential users for each community.

1) Dependency calculation: Dependency refers to the extent to which a node relies on another node to access the rest of the community. It is calculated iteratively for each node within the community graph using the following equation:

$$\text{dependency}[v] += \left(\frac{\text{sigma}[w]}{\text{sigma}[v]}\right) \times (1 + \text{dependency}[w]) \tag{7}$$

where:

- dependency$[v]$ is the dependency of node v,
- sigma$[v]$ is the number of shortest paths starting from node v,
- sigma$[w]$ is the number of shortest paths passing through node w,
- dependency$[w]$ is the dependency of node w.

2) Betweenness centrality calculation: Betweenness centrality measures the extent to which a node lies on the shortest paths between other nodes in the community. It is computed based on the accumulated dependency values of nodes within the community graph:

$$\text{betweenness\_centrality}[w] += \text{dependency}[w] \tag{8}$$

where:

- betweenness_centrality$[w]$ is the betweenness centrality of node w, and
- dependency$[w]$ is the dependency of node w.

REFERENCES

[1] M. Li, S. Lu, L. Zhang, Y. Zhang and B. Zhang, "A Community Detection Method for Social Network Based on Community Embedding," in IEEE Transactions on Computational Social Systems, vol. 8, no. 2, pp. 308-318, April 2021, doi: 10.1109/TCSS.2021.3050397.

[2] T. Zhu, P. Li, L. Yu, K. Chen and Y. Chen, "Change Point Detection in Dynamic Networks Based on Community Identification," in IEEE Transactions on Network Science and Engineering, vol. 7, no. 3, pp. 2067-2077, 1 July-Sept. 2020, doi: 10.1109/TNSE.2020.2973328.

[3] C. Tu et al., "A Unified Framework for Community Detection and Network Representation Learning," in IEEE Transactions on Knowledge and Data Engineering, vol. 31, no. 6, pp. 1051-1065, 1 June 2019, doi: 10.1109/TKDE.2018.2852958.

[4] Y. Liu, Z. Li, S. Pan, C. Gong, C. Zhou and G. Karypis, "Anomaly Detection on Attributed Networks via Contrastive Self-Supervised Learning," in IEEE Transactions on Neural Networks and Learning Systems, vol. 33, no. 6, pp. 2378-2392, June 2022, doi: 10.1109/TNNLS.2021.3068344.

[5] K. Berahmand, M. Mohammadi, F. Saberi-Movahed, Y. Li and Y. Xu, "Graph Regularized Nonnegative Matrix Factorization for Community Detection in Attributed Networks," in IEEE Transactions on Network Science and Engineering, vol. 10, no. 1, pp. 372-385, 1 Jan.-Feb. 2023, doi: 10.1109/TNSE.2022.3210233.

[6] E. D. Raj, G. Manogaran, G. Srivastava and Y. Wu, "Information Granulation-Based Community Detection for Social Networks," in IEEE Transactions on Computational Social Systems, vol. 8, no. 1, pp. 122-133, Feb. 2021, doi: 10.1109/TCSS.2019.2963247.

[7] Z. Li, X. Chen, J. Song and J. Gao, "Adaptive Label Propagation for Group Anomaly Detection in Large-Scale Networks," in IEEE Transactions on Knowledge and Data Engineering, vol. 35, no. 12, pp. 12053-12067, 1 Dec. 2023, doi: 10.1109/TKDE.2022.3176478.

[8] L. Huang et al., "Hybrid-Order Anomaly Detection on Attributed Networks," in IEEE Transactions on Knowledge and Data Engineering, vol. 35, no. 12, pp. 12249-12263, 1 Dec. 2023, doi: 10.1109/TKDE.2021.3117842.

[9] S. Zavrak and M. İskefiyeli, "Anomaly-Based Intrusion Detection from Network Flow Features Using Variational Autoencoder," IEEE Access, vol. 8, pp. 108348-108351, 2020. doi: 10.1109/ACCESS.2020.3001350.

[10] C. -I. Chang, "Target-to-Anomaly Conversion for Hyperspectral Anomaly Detection," in IEEE Transactions on Geoscience and Remote Sensing, vol. 60, pp. 1-28, 2022, Art no. 5540428, doi: 10.1109/TGRS.2022.3211696.S. Souravlas et al., "Probabilistic Community Detection in Social Networks," IEEE Access, vol. 11, pp. 18662-18672, 2023. doi: 10.1109/ACCESS.2023.3257021.

[11] C. He, X. Fei, Q. Cheng, H. Li, Z. Hu and Y. Tang, "A Survey of Community Detection in Complex Networks Using Nonnegative Matrix Factorization," in IEEE Transactions on Computational Social Systems, vol. 9, no. 2, pp. 440-457, April 2022, doi: 10.1109/TCSS.2021.3114419.

[12] W. Luo, D. Zhang, L. Ni and N. Lu, "Multiscale Local Community Detection in Social Networks," in IEEE Transactions on Knowledge and Data Engineering, vol. 33, no. 3, pp. 1102-1112, 1 March 2021, doi: 10.1109/TKDE.2019.2938173.

[13] X. Su et al., "A Comprehensive Survey on Community Detection With Deep Learning," in IEEE Transactions on Neural Networks and Learning Systems, doi: 10.1109/TNNLS.2021.3137396.

[14] X. Huang et al., "A survey of community detection methods in multilayer networks," in Data Mining and Knowledge Discovery, vol. 35, pp. 1–45, Jan. 2021, doi: 10.1007/s10618-020-00716-6.

[15] R. H. Elghanuni, M. A. M. Ali and M. B. Swidan, "An Overview of Anomaly Detection for Online Social Network," 2019 IEEE 10th Control and System Graduate Research Colloquium (ICSGRC), Shah Alam, Malaysia, 2019, pp. 172-177, doi: 10.1109/ICSGRC.2019.8837066.

[16] N. Hafiene, W. Karoui, L. B. Romdhane, "Influential nodes detection in dynamic social networks: A survey," Expert Systems with Applications, vol. 159, p. 113642, 2020, ISSN 0957-4174, doi: 10.1016/j.eswa.2020.113642.

[17] C. Zhang, Y. Zhang, and B. Wu, "A Parallel Community Detection Algorithm based on Incremental Clustering in Dynamic Network," in Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Beijing, China, 2018

[18] X. Luo, J. Wu, A. Beheshti, J. Yang, X. Zhang, Y. Wang, and S. Xue, "ComGA: Community-Aware Attributed Graph Anomaly Detection," in Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM),

[19] Y. Sebastian, J. T. Chew, X. T. Tiong, V. Raman, A. Y. Y. Fong, and P. H. H. Then, "Anomaly Detection from Diabetes Similarity Graphs using Community Detection and Bayesian Techniques," in Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)

[20] H. Sun, Y. Li, B. Lv, W. Yan, L. He, S. Qiao, and J. Huang, "Graph Community Infomax," in Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)

[21] Y. Wang, C. Yang, D. Jin, and J. Dang, "Integrating Group Homophily and Individual Personality of Topics Can Better Model Network Communities," in Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)

[22] C. Tran, W.-Y. Shin, and A. Spitz, "Community Detection in Partially Observable Social Networks," in Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining

[23] F. Folino and C. Pizzuti, "Multiobjective Evolutionary Community Detection for Dynamic Networks," in Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining