Community Detection and Analysis on the DBLP v9 Dataset: Exploring Centrality, Network Metrics, and Predictive Modeling

Xiang Zheng
21307110169

Qin Ma
21307110024

Author n
Address line

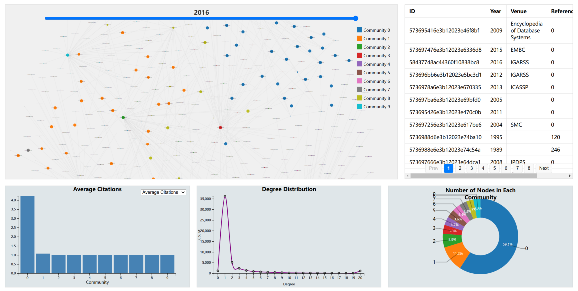
School of Data Science, Fudan University

Abstract

This study applies network analysis techniques to the DBLP dataset, a comprehensive resource for computer science publications, to explore collaboration patterns and network properties. Key methods include data preprocessing, community detection with Louvain and Label Propagation, centrality analysis using degree centrality and PageRank, and link prediction with the GLACE model. Results highlight the evolution of citation trends and identify influential nodes. An interactive visualization system, built with Python and D3.js, helps present the findings. The experimental results demonstrate the GLACE model's effectiveness in predicting future links, providing valuable insights into academic networks.

1 Introduction

Network analysis is a cornerstone methodology for understanding complex systems across diverse domains. The DBLP dataset, a comprehensive bibliographic resource for computer science, serves as a foundation for analyzing collaboration patterns and scholarly relationships. This study leverages advanced graph analysis techniques to investigate the structural and functional properties of the DBLP v9 dataset.



ID	Year	Venue	Referenc
573695416e3b12023e46f8bf	2009	Encyclopedia of Database Systems	0
573697476e3b12023e6336d8	2015	EMBC	0
58437748ac44360f10838bc8	2016	IGARSS	0
573696bb6e3b12023e5bc3d1	2012	IGARSS	0
5736978a6e3b12023e670335	2013	ICASSP	0
573697ba6e3b12023e69bfd0	2005		0
573695426e3b12023e470c0b	2011		0
573697256e3b12023e617be6	2004	SMC	0
5736988d6e3b12023e74ba10	1995		120
5736988e6e3b12023e74c54a	1989		246
573697666e3b12023e64dca1	2008	IPDPS	0

Figure 1: Snapshot of the visualization system

This report is structured as follows:

Preprocessing: Transforming the raw DBLP data into a structured network through cleaning, filtering, and attribute assignment.

Community Detection: Identifying cohesive subgroups using algorithms like Louvain and Label Propagation to uncover collaboration dynamics and research specializations.

Centrality Analysis: Analyzing metrics such as degree centrality and PageRank to identify influential nodes and assess network topology.

Link Prediction: Employing the GLACE model on the Cora-ML dataset to predict future connections based on structural features, providing insights into evolving citation trends.

Visualization: Presenting findings through an interactive system developed with **Python** and **D3.js**, highlighting key network characteristics and community structures.

This structured approach provides a comprehensive examination of the DBLP co-authorship and publication network, delivering insights into collaboration patterns, influential figures, and the evolution of academic networks.

2 Preprocessing

The preprocessing phase is a critical step in transforming the raw DBLP v9 dataset into a structured format suitable for detailed analysis. This stage encompasses data cleaning, network construction, feature engineering, and dataset filtering. The DBLP v9 dataset was specifically chosen for its balance between computational feasibility and data comprehensiveness, enabling robust analysis of academic collaboration and citation patterns within computer science.

2.1 Dataset Selection

The DBLP-Citation-network v9 dataset was selected due to its optimal balance between data richness and computational manageability, as summarized in Table 1. Other versions of the DBLP dataset, while valuable, either lacked the depth of information or exceeded practical computational limits for this project. DBLP v9 captures key trends up to July 3, 2017, offering a comprehensive yet tractable dataset with **3,680,007 papers** and **1,876,067 citation relationships**.

2.2 Data Cleaning and Integration

The raw DBLP dataset includes metadata such as titles, authors, venues, and citations. Data cleaning involved:

- **Grouping Records:** Papers were grouped using unique identifiers to ensure each record was correctly structured.
- **Resolving Missing Data:** Missing fields, such as titles, authors, or venues, were assigned default values to maintain consistency.

These steps ensured the dataset was standardized and ready for subsequent analysis.

2.3 Network Construction

The dataset was represented as two interconnected networks:

1. **Author Network:** Authors are represented as nodes, with weighted edges denoting co-authorship relationships. The weight of each edge corresponds to the frequency of collaborations between authors. This network contains a total of 3,680,007 nodes (authors) and 1,876,067 edges (co-authorship relationships).
2. **Paper Network:** Papers are represented as nodes, with directed edges indicating citation relationships. Each edge direction signifies the citing paper and the cited paper. This network also comprises 3,680,007 nodes (papers) and 1,876,067 edges (citation relationships).

This dual representation enables a comprehensive study of collaboration and citation dynamics.

2.4 Feature Engineering

Key features were engineered to enhance the dataset's analytical capabilities:

- **Co-authorship Features:** Unique author identifiers were assigned, and collaboration frequencies were calculated.
- **Citation Metrics:** In-degree (citations received) and out-degree (references made) were computed for each paper.
- **Venue Indexing:** Publication venues were standardized and indexed for uniform representation.

2.5 Dataset Filtering

Papers with no citations or references were flagged as "isolates" and excluded to improve computational efficiency. Thresholds were also applied to focus on significant collaborations and impactful papers.

2.6 Exploratory Analysis

Exploratory analyses were conducted using Python libraries such as `pandas` and `matplotlib`. Key findings are visualized in Figures 2–5:

- **Authors per Paper:** Most papers have few authors, with fewer multi-author publications.
- **Citation Distribution:** Citations are highly skewed, with a small number of papers receiving the majority of citations.
- **References per Paper:** Papers with more citations tend to reference more works.
- **Co-authors per Author:** A small group of authors collaborate extensively, while most have limited collaborations.

These analyses provide valuable insights into academic collaboration and citation patterns, establishing a solid foundation for deeper exploration of academic network structures.

3 Community Mining

Community mining in academic networks is essential for identifying cohesive subgroups that reflect underlying collaborative dynamics, research specializations, and the structural foundation of scholarly interactions. In this study, three widely used algorithms—Louvain, Label Propagation, and Multilevel—were employed to detect communities in both paper and author networks. These algorithms were chosen due to their computational efficiency, scalability, and ability to capture different aspects of community structure in large networks.

The effectiveness of community detection is assessed using modularity, a key metric that quantifies the strength of division between communities by comparing the actual number of edges within communities to the expected number of edges in

Dataset Version	Number of Papers	Number of Citation Relationships
Citation-network V1	629,814	>632,752
Citation-network V2	1,397,240	>3,021,489
DBLP-Citation-network V3	1,632,442	>2,327,450
DBLP-Citation-network V4	1,511,035	2,084,019
DBLP-Citation-network V5	1,572,277	2,084,019
DBLP-Citation-network V6	2,084,055	2,244,018
DBLP-Citation-network V7	2,244,021	4,354,534
DBLP-Citation-network V8	3,272,991	8,466,859
DBLP-Citation-network V9	**3,680,007**	**1,876,067**
DBLP-Citation-network V10	3,079,007	25,166,994
DBLP-Citation-network V11	4,107,340	36,624,464
DBLP-Citation-network V12	4,894,081	45,564,149
DBLP-Citation-network V13	5,354,309	48,227,950
DBLP-Citation-network V14	5,259,858	36,630,661

Table 1: Summary of DBLP Citation Network Versions

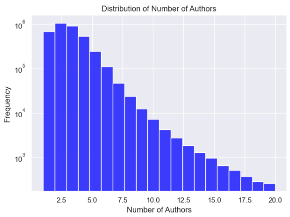


Figure 2: Number of Authors per Paper

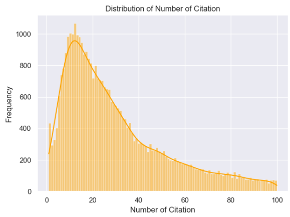


Figure 3: Citation Distribution of Papers

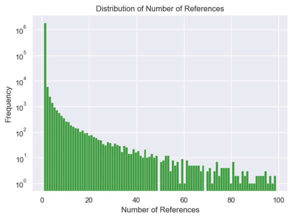


Figure 4: Reference Distribution of Papers

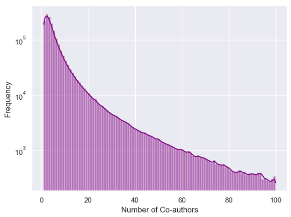


Figure 5: Number of Co-authors per Author

a random graph. A higher modularity value indicates a more significant and well-defined community structure, which is desirable in network analysis.

3.1 Overview of Community Mining

Community detection aims to partition the graph into subsets (communities) where the internal connectivity is higher than expected by random chance, while the external connectivity is relatively sparse. In academic networks, nodes typically represent entities such as papers or authors, and edges signify relationships like co-authorships or citations. The following outlines the network structures in the context of academic collaboration:

- **Paper Network:** In a paper network, nodes represent individual research papers, and edges represent citations. The goal is to identify thematic subfields or areas of active research, where papers share common references or citations, uncovering latent academic associations.
- **Author Network:** In an author network, nodes represent academic authors, and edges denote co-authorships, with edge weights indicating the strength of collaboration. Community detection in this context reveals research teams, collaborative clusters, and patterns of academic interaction.

The modularity measure, Q, is a crucial tool for evaluating community quality, and is given by the formula:

$$Q = \frac{1}{2m} \sum_{i,j} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j),$$

where m is the total number of edges, A_{ij} is the weight of the edge between nodes i and j, k_i and k_j are the degrees of nodes i and j, and $\delta(c_i, c_j)$ is the Kronecker delta, which is 1 if nodes i and j are in the same community and 0 otherwise.

3.2 Community Detection Algorithms

Several algorithms have been developed to identify communities in networks. Here, we discuss three prominent ones: Louvain, Label Propagation, and Multi-level, each offering unique advantages depending on the network characteristics.

3.2.1 Louvain Algorithm

The Louvain algorithm is a modularity optimization method that efficiently detects communities by iteratively optimizing modularity. It operates in two phases:

1. **Local optimization:** Each node is initially assigned to its own community. Nodes are then moved to the community of their neighboring node if it results in a higher modularity.
2. **Community aggregation:** After the first phase, communities are treated as super-nodes, and the process repeats on the new, aggregated graph.

The algorithm's efficiency makes it particularly suited for large networks, allowing for the identification of well-defined, cohesive communities that often correspond to real-world academic fields or research themes.

3.2.2 Label Propagation Algorithm

Label Propagation is a simple yet effective community detection algorithm. It works by assigning a unique label to each node initially and then iteratively updating the node's label to the most frequent label among its neighbors. The process continues until the labels converge. The update rule is expressed mathematically as:

$$l_i^{(t+1)} = \arg \max_{l_j \in N(i)} \sum_{k \in N(i), l_k = l_j} \frac{1}{d_k},$$

where $l_i^{(t+1)}$ is the new label for node i after iteration t, $N(i)$ is the set of neighbors of node i, and d_k is the degree of neighbor k.

Label Propagation is highly scalable and does not require predefining the number of communities, making it well-suited for large networks with an unknown structure.

3.2.3 Multi-level Algorithm

The Multi-level algorithm follows a hierarchical approach for community detection. The process involves three main steps:

1. **Coarsing:** The graph is iteratively reduced by merging nodes that are strongly connected, creating a smaller, coarser graph.
2. **Community Detection:** Community detection is applied to the coarser graph, typically using modularity maximization.
3. **Uncoarsing:** The community structure is refined by gradually returning to the original graph and adjusting the community assignments accordingly.

This method is particularly useful for large-scale networks and allows for community detection across multiple levels of granularity.

3.3 Community Proportions and Modularity

The proportion of nodes in community c relative to the total network is given by:

$$\text{Proportion of community } c = \frac{N_c}{N_{\text{total}}},$$

where N_c is the number of nodes in community c, and N_{total} is the total number of nodes in the network. This metric helps to assess the relative significance of different communities in the overall network.

Modularity Q serves as a key measure for evaluating the strength of community structure, with values greater than 0.7 indicating a significant division of the network into distinct communities. Table 2 presents the modularity values for the three algorithms employed in the paper network analysis.

Algorithm	Modularity
Louvain	0.9939
Label Propagation	0.9909
Multi-level	0.9938

Table 2: Modularity values for community detection in the paper network.

3.4 Community Detection Results

The results of community detection in the paper network revealed that the Louvain algorithm identified 29,898 communities, Label Propagation detected 30,549 communities, and Multi-level found 29,589 communities. Figure 6 illustrates the community partitioning of the paper network using the Louvain algorithm.

4 Centrality Measurement

Centrality measures are critical for identifying influential nodes within a network. In academic networks, these measures help pinpoint key authors,

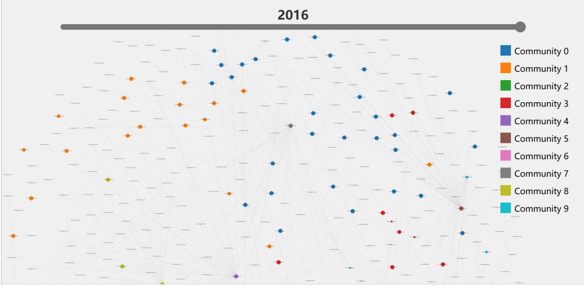


Figure 6: Community partitioning of the paper network using the Louvain algorithm.

influential papers, and collaborative hubs. In this study, we employ Degree Centrality and PageRank Centrality as two complementary metrics that capture local and global node influence, respectively.

4.1 Centrality Measures

4.1.1 Degree Centrality

Degree centrality counts the number of direct connections (edges) a node has, serving as an indicator of a node's connectivity within the network. In academic networks, a higher degree centrality often correlates with more collaborative activity or greater visibility. It is calculated as:

$$C_d(i) = \deg(i),$$

where $\deg(i)$ is the degree of node i. Nodes with high degree centrality are typically influential because they are well-connected to many other nodes.

4.1.2 PageRank Centrality

PageRank centrality assesses node influence by considering both the quantity and quality of its incoming edges. It accounts for the fact that links from highly connected or authoritative nodes are more significant than those from less connected ones. PageRank is calculated iteratively as follows:

$$PR(i) = \frac{1-d}{N} + d \sum_{j \in \mathcal{N}(i)} \frac{PR(j)}{|\mathcal{N}(j)|},$$

where d is the damping factor (set to 0.85 by default), N is the total number of nodes, $\mathcal{N}(i)$ is the set of neighbors of node i, and $|\mathcal{N}(j)|$ is the number of outgoing edges from node j.

PageRank centrality is particularly useful for identifying authoritative nodes, where a higher score indicates not just popularity but also the quality of influence, reflecting the academic importance of the paper or author.

4.2 Community Diameters

The diameter of a community, defined as the longest shortest path between any two nodes within the community, serves as an indicator of cohesion. A smaller diameter implies that nodes are more closely connected, suggesting a tighter-knit community. Conversely, a larger diameter may suggest the presence of subgroups or weak connections within the community, potentially highlighting fragmented or less cohesive research areas.

These centrality measures and structural analyses provide essential insights into the organization of academic networks, enabling the identification of key players and community structures that are critical for understanding collaborative patterns and research dynamics.

5 Link Prediction

Traditional link prediction methods for graph data mainly rely on structural features and similarity metrics of the graph. By calculating the similarity between nodes, utilizing path information, or applying statistical models, these methods predict potential links. They are simple and efficient, suitable for various types of graph data. Although they may face challenges in computational efficiency and accuracy when dealing with large and complex graphs, they lay the foundation for more advanced machine learning and deep learning-based approaches.

5.1 Similarity-Based Metrics

These methods predict potential links by calculating similarity scores between pairs of nodes. Common similarity metrics include:

5.1.1 Common Neighbors (CN)

Measures the number of shared neighbors between two nodes. The more common neighbors two nodes have, the higher the likelihood of a link forming between them.

$$\text{CN}(x, y) = |\Gamma(x) \cap \Gamma(y)|$$

where $\Gamma(x)$ denotes the set of neighbors of node x.

5.1.2 Jaccard Coefficient

Measures the ratio of the intersection to the union of the neighbor sets of two nodes, with values ranging from 0 to 1.

$$J(x, y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|}$$

5.2 Introduction of GLACE

With the advancement of deep learning technologies, neural network-based models have shown remarkable performance in link prediction tasks. This report introduces the GLACE (Gaussian Latent Attribute-based Contrastive Embedding) model and its application in link prediction. The GLACE model is a Gaussian-based graph embedding method designed for link prediction tasks. Unlike the LACE model, GLACE learns Gaussian distribution embeddings (mean μ and variance σ) for each node, enabling it to better capture the uncertainty and complex relationships between nodes. The model minimizes the symmetric Kullback-Leibler (KL) divergence between node pairs, aligning their distributions in the embedding space to enhance link prediction accuracy.

5.3 Model Architecture

The main components of the GLACE model include:

- **Input Processing**: Handles the sparse adjacency matrix by converting it into a format suitable for PyTorch sparse tensors.
- **Encoder**: A multi-layer fully connected neural network that extracts latent features of nodes.
- **Mean and Variance Embedding Layers**: Linear layers that generate the mean μ and log variance $\log \sigma$ for node embeddings.
- **Context Encoder**: Used when considering second-order proximity, it generates context embeddings for nodes.
- **Optimizer**: Utilizes the Adam optimizer for training the model parameters.

5.4 Gaussian Embeddings and KL Divergence

GLACE learns Gaussian distribution embeddings for each node, represented as (μ, σ). During link prediction, the symmetric KL divergence between node pairs is computed to measure their similarity. The specific steps are as follows:

1. For a node pair (u_i, u_j), retrieve their means μ_i, μ_j and variances σ_i, σ_j.
2. Calculate the KL divergence $KL(P||Q)$ and $KL(Q||P)$, where P and Q represent the Gaussian distributions of nodes u_i and u_j, respectively.
3. Average the two divergences to obtain the distance metric $KL\_distance$ between the node pair.

5.5 Model Training and Optimization

The training process of the GLACE model involves several key steps:

5.5.1 Loss Function

The model employs a log-sigmoid loss function, suitable for binary classification tasks in link prediction. It is defined as:

$$\text{Loss} = -\frac{1}{N} \sum_{i=1}^{N} \log \sigma(label_i \cdot energy_i)$$

where $label_i$ is the true label (1 for positive samples, -1 for negative samples), and $energy_i$ is the energy value computed by the model (negative KL divergence).

5.5.2 Optimization Process

The Adam optimizer is used to update the model parameters, with the learning rate specified by the experimental setup. The optimization goal is to minimize the loss function, thereby improving the model's performance in link prediction tasks.

5.5.3 Experimental Results

Dataset For our experiments, we utilized the **Cora_ML** dataset, a widely recognized benchmark in the field of link prediction and graph-based learning. The Cora_ML dataset consists of scientific publications classified into various topics, with citation links representing the relationships between these publications. Specifically, the dataset contains 2,708 nodes (publications), 5,429 edges (citations), and 1,433 features representing the presence of specific words in the documents. This dataset is well-suited for evaluating the performance of graph embedding models like GLACE in predicting missing or potential links within the citation network.

Results The GLACE model was trained and evaluated on the Cora_ML dataset over multiple batches. The key performance metrics recorded during the training process include loss, validation AUC (Area Under the Curve), and validation AP (Average Precision). Table 3 presents a summarized view of the results across different training batches, with intermediate batches omitted for brevity.

Table 3: GLACE Model Performance on Cora_ML Dataset

Batch	Loss	Val AUC	Val AP
49	0.303630	0.849437	0.828072
99	0.258379	0.891329	0.872213
149	0.266745	0.910746	0.896074
⋮	⋮	⋮	⋮
1749	0.172694	0.952421	0.949143
1799	0.182493	0.953662	0.948588
1849	0.207456	0.954621	0.950908

Analysis of Results The experimental results on the Cora_ML dataset demonstrate the effectiveness of the GLACE model in link prediction tasks. As observed from Table 3, several key trends emerge:

- **Loss Reduction**: The loss consistently decreases as training progresses, indicating that the model is effectively learning to minimize the discrepancy between predicted and actual links. For instance, the loss decreased from 0.303630 at batch 49 to 0.172694 at batch 1749.

- **Performance Metrics**: Both validation AUC and validation AP show an overall upward trend, reaching values above 0.95 towards the later batches. This signifies that the model's ability to distinguish between positive and negative links improves with training. For example, the validation AUC increased from 0.849437 at batch 49 to 0.954621 at batch 1849, and the validation AP similarly rose from 0.828072 to 0.950908.

- **Stability of Training**: The training process progresses smoothly without significant interruptions or delays, ensuring a steady training flow. The consistent decrease in loss and increase in performance metrics reflect the model's stable and effective learning dynamics.

Overall, the GLACE model exhibits robust performance in link prediction on the Cora_ML dataset, achieving high accuracy and precision. The incorporation of Gaussian embeddings allows the model to capture nuanced relationships between nodes, resulting in superior predictive capabilities compared to traditional methods.

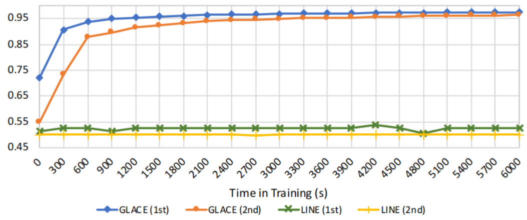


Figure 7: GLACE's link prediction performance

6 Visualization System

This study introduces a comprehensive visualization system designed to represent a paper citation network, where nodes correspond to academic papers and edges represent citations. The system is organized into five main sections: community network, bibliometric data, degree distribution, node distribution, and additional metrics. An overview of the system is shown in Figure 1.

Given the modularity of author communities (0.66) and paper communities (0.99), the focus is placed on paper-level data, which offers a clearer and more structured representation of the citation network. The higher modularity of paper communities facilitates more detailed analysis. For optimal visualization, data is derived from the top 10 paper communities and the 50 highest centrality papers, ensuring the most influential communities and papers are prominently displayed. This approach enhances both clarity and effectiveness.

6.1 Top Left: Community Network Visualization with Interactive Features

The top-left section visualizes the paper communities using an interactive network graph. Nodes represent individual papers, and edges indicate citations. Users can zoom, pan, and select nodes, which dynamically updates the corresponding sections (top-right and bottom-middle) with community-specific data. A year slider allows users to track the temporal evolution of the citation network, highlighting changes in structure over time.

6.2 Top Right: Bibliometric Data Table

The top-right section presents a dynamic bibliometric table containing key attributes such as paper ID, publication year, venue, references, and citation count. This table updates in response to node selections from the network graph, ensuring that users view relevant bibliometric data for the selected community. Pagination and dynamic updates allow efficient browsing of large datasets.

6.3 Bottom Left: Dynamic Bar Chart Visualization

The bottom-left section features a dynamic bar chart for metrics such as "Average Citations," "Average Centrality," and "Diameter." Users can select a metric from a dropdown menu, and the bar chart visually compares these metrics across communities. The chart includes smooth transitions, tooltips, and highlighted bars on hover, enhancing interactivity.

6.4 Bottom Middle: Degree Distribution Visualization

The bottom-middle section visualizes the degree distribution of communities within the network using a smooth line chart with spline curves. This chart depicts the frequency of nodes with each degree. Upon selecting a node in the network graph, the degree distribution for the corresponding community is displayed dynamically. Tooltips and adjustable dot sizes further improve visibility, and degree values exceeding a specified threshold are grouped for clarity.

6.5 Bottom Right: Node Distribution with Donut Chart

The bottom-right section uses a donut chart to show the distribution of nodes across communities. Each slice represents a community, with its size proportional to the number of nodes. Interactive features such as hover effects and percentage labels allow users to explore the relative sizes of the communities and gain insights into the network's composition.

6.6 System Overview and Interactivity

The visualization system integrates these components into a cohesive interface that enables detailed exploration of the citation network. Key interactive features include zooming, panning, node selection, and year slider adjustments. Selecting a node updates the bibliometric data and degree distribution sections with community-specific information. The year slider adds a temporal dimension, revealing the evolution of the network over time. Powered by the `D3.js` library, the system ensures responsive, high-quality visualizations and an engaging user experience.

7 Conclusion

This report presents a structured approach to analyzing academic collaboration networks using the DBLP-V9 dataset. By applying community detection, centrality analysis, and link prediction, we uncover key patterns in scholarly communication. The GLACE model shows promise in predicting future links, outperforming traditional methods. An interactive visualization system further aids in interpreting the results. This work offers valuable insights into academic networks, with potential applications in other domains for understanding complex systems.