Author Classification & Citation Recommendation on Bibliographic Data in Neo4j

Afsah Hyder
School of Science and Engineering
Habib University
Karachi, Pakistan
ah07065@st.habib.edu.pk

Eman Fatima
School of Science and Engineering
Habib University
Karachi, Pakistan
ef08595@st.habib.edu.pk

Fakeha Faisal
School of Science and Engineering
Habib University
Karachi, Pakistan
ff08288@st.habib.edu.pk

***Abstract*—This project applies graph-based machine learning techniques to a scholarly bibliographic dataset, with the goal of understanding research patterns and enhancing citation and author analysis. Using Neo4j's Graph Data Science (GDS) library, two core tasks are addressed: author classification and citation link prediction. For author classification, we predict an author's research domain using structural features from the co-authorship network. For link prediction, we identify potential citation links between papers using supervised models built on FastRP embeddings and pairwise similarity metrics. The models were assessed using Area Under the Precision-Recall Curve (AUCPR) and Out-of-Bag (OOB) error. Results show significant improvements—up to 12 percentage points in AUCPR—when incorporating time-based splits and structural features, demonstrating the effectiveness of graph-based approaches for bibliometric analysis and citation recommendation.**

I. Introduction

This report details the methodology and rationale for a Graph Data Science (GDS) project focused on applying machine learning techniques to a bibliographic dataset using graph-based approaches. The project involved three group members, who divided their efforts between two tasks: node classification and link prediction. For node classification, we focused on author classification to predict the research domain or expertise of authors based on their co-authorship network. For link prediction, we concentrated on citation recommendation to suggest relevant citations for research papers by predicting possible citation links. The dataset, comprising authors, papers, journals, topics, and their relationships, was cleaned, preprocessed, and loaded into a Neo4j graph database to support these tasks. The data cleaning, preprocessing, and graph construction processes were shared across both tasks to ensure consistency.

II. Methodology

A. Data Preprocessing

The preprocessing and graph construction steps were critical to ensure a high-quality dataset and a robust graph model. Data cleaning was performed using the R programming language to ensure consistency, validity, and readiness for integration into a Neo4j graph database. The process involved removing duplicate entries, handling null values, verifying data types, and standardizing formats across all datasets to prevent inconsistencies and errors. This ensured the data was reliable for constructing the graph and performing downstream machine learning tasks.

The cleaned datasets were loaded into a Neo4j graph database using Cypher queries designed to create nodes and relationships efficiently. The loading process was optimized for performance and data integrity, incorporating constraints, indexes, and batch processing to handle large datasets. Unique constraints were created for key identifiers (e.g., Author.authorId, Paper.paperId) to enforce data uniqueness, and indexes were added on frequently queried properties (e.g., Author.name, Paper.title) to improve query performance. The `:auto and CALL with IN TRANSACTIONS` ensured efficient batch processing, while `MERGE` prevented duplicate nodes and relationships.

After creating relationships, redundant properties were removed to normalize the graph structure. For example, publisher and email properties were removed from Journal nodes after linking to Publisher nodes via `PUBLISHED_BY` relationships. Similarly, volume and date properties were transferred from Paper nodes to `HAS` relationships between Journal and Paper nodes, then removed from Paper nodes.

B. Data Modeling and Graph Construction

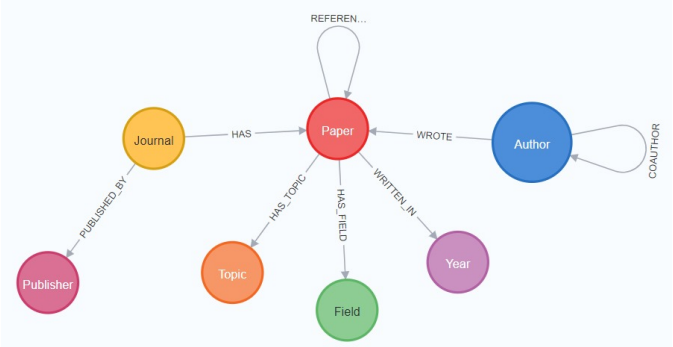

Fig. 1. Graph Data Model

Figure 2 shows the data model used in our graph construction. The graph was designed using the property

graph model to represent the bibliographic dataset, capturing entities and relationships among Author, Paper, Journal, Topic, Field, Year, and Publisher nodes.

To ensure a clean and semantically rich graph structure suitable for tasks such as link prediction, node classification, and graph analytics, several structural modifications were made to the initial data model. The year attribute was removed from the Paper node and instead modeled as a separate Year node, connected through a [:WRITTEN_IN] relationship. This design enables temporal queries and facilitates year-based aggregation to analyze publication trends over time. It enables time-aware queries and yearly trend analysis, such as detecting evolving research topics or identifying shifts in collaboration networks over time. This temporal dimension enhances the interpretability of patterns in the scholarly landscape and supports longitudinal studies of academic publishing. The citationCount property, originally associated with the Paper node, was removed and reassigned as a property of the [:REFERENCES] relationship between Paper nodes. This change reflects its specific role in link prediction, where citation frequency serves as an edge-level feature indicating the strength or frequency of the reference link.

Similarly, the fields of study attribute was removed from the Paper node and modeled as a separate Field node. Each Paper is connected to its corresponding Field using a [:HAS_FIELD] relationship. This normalization promotes data consistency and prevents redundancy when multiple papers share the same field. Additionally, the volume and date values, which were initially stored as properties on the Paper node, were reassigned to the [:HAS] relationship between Journal and Paper. As these values describe the context of the paper's appearance within a journal issue, modeling them as edge properties provides a more accurate and expressive representation of the publication metadata.

The publisher attribute, previously embedded within the Journal node, was extracted and represented as a distinct Publisher node. Journals are now connected to their respective publishers using the [:PUBLISHED_BY] relationship. This separation allows for a modular and scalable representation of publishers, enabling cross-journal queries and a clearer understanding of publishing networks.

The resulting graph structure supports a wide range of analytical tasks. The property graph enables the computation of structural features such as degree centrality, betweenness centrality, and community detection, all of which are essential for link prediction and node classification. The [:COAUTHOR] relationship includes a paperCount property, capturing the number of shared papers between authors and offering a weighted measure of collaboration intensity. This is especially useful for predicting future co-authorship links.

Node classification is supported through the [:HAS_FIELD] and [:HAS_TOPIC] relationships, allowing classification of Paper and Author nodes based on research domains and areas of focus. The [:COAUTHOR] structure additionally supports classification based on collaboration behavior. For link prediction, relationships such as [:REFERENCES] and [:COAUTHOR] provide a historical basis for modeling future citations and collaborations. Furthermore, the [:HAS] relationship between Journal and Paper, with its contextual properties like volume and date, supports journal-paper recommendation scenarios.

These modeling choices reflect best practices in graph database design, ensuring data normalization, flexibility in querying, and optimized support for complex graph-based machine learning workflows.

Nodes

- Author { authorID: String, name: String, url: String }
- Paper { paperID: String, DOI: String, title: String, url: String }
- Journal { name: String }
- Topic { topicID: String, name: String, url: String }
- Field { name: String }
- Year { value: String }
- Publisher { name: String }

Relationships

- (:Author)-[:WROTE]->(:Paper)
- (:Author)-[:COAUTHOR {paperCount: Integer}]->(:Author)
- (:Journal)-[:HAS {volume: String, date: String}]->(:Paper)
- (:Paper)-[:REFERENCES {citationCount: Integer}]->(:Paper)
- (:Paper)-[:HAS_TOPIC]->(:Topic)
- (:Paper)-[:HAS_FIELD]->(:Field)
- (:Paper)-[:WRITTEN_IN]->(:Year)
- (:Journal)-[:PUBLISHED_BY]->(:Publisher)

III. Exploratory Analysis

There are a total of 498915 Paper nodes, 38853 Author nodes, 6489 Topic nodes, and 39 Topic nodes. Furthermore, there are 762846 CITES relationships, 39255 AUTHORED relationships, 31824 HAS_TOPIC relationships, and 1869 PUBLISHED_IN relationships. Some of the most popular Authors based on their average citations per paper are S. Vertovec with 363 citations, A. Portes with 208 citations, C. Ward with 144 citations, and D. Massey with 87 citations.

An initial analysis of the graph revealed informative patterns in author collaboration as well with S. Kim having collaborated with 76 other authors, S. Schwartz collaborating with

62 authors, and Byron I. Zamboanga collaborating with 44 authors. Added to that, some of the most cited papers are *"Super-diversity and its Implications"* with 2,216 citations, *"Theories of International Migration: Review and Appraisal"* with 2,063 citations, and *"Whose Culture Has Capital? A Critical Race Theory"* with 1,739 citations. Figure 2 below shows some of the papers and other papers that they cite.

Fig. 2. Citation Relationships among Papers

As an initial step in exploring the structure of the network, the Louvain community detection algorithm was applied to the co-authorship subgraph, revealing densely connected clusters that often correspond to institutional or topical affiliations.

In a separate analysis, degree centrality was employed to identify the most prominent topics across the network. The top-ranked topics, based on their centrality scores, were *experience* (531), *labor* (357), and *community* (333), indicating their frequent association within the literature. To further validate the importance of these topics, PageRank centrality was applied to identify influential papers. The results reinforced the initial findings, with papers associated with these topics receiving PageRank scores of 62.57, 49.67, and 42.95, respectively.

Topic pairs most frequently studied together, based on their co-occurrence counts, include *labour (childbirth)* and *policy*, *mental disorders* and *mental health*, as well as *experience* and *community*. This suggests that these topic pairings represent closely linked areas of inquiry, often addressed jointly in the literature due to their conceptual or practical interdependence.

These thematic connections are also reflected in the venues where such research is published. Some of the most prolific journals, based on publication count, include the *Journal of Ethnic and Migration Studies*, *Ethnic and Racial Studies*, and the *Journal of Black Studies*. In contrast, the most cited journals are *Ethnic and Racial Studies*, the *International Journal of Intercultural Relations*, and the *International Migration Review*. This indicates that while some journals contribute a high volume of research, others—perhaps with a narrower focus or higher impact—play a more influential role in shaping the field through citations.

To explore interdisciplinary connectivity among authors, betweenness centrality was computed on a graph where authors were linked to the fields of their publications. This analysis aimed to identify authors who act as bridges between different research areas. The projection was constructed using a bipartite graph of `Author` and `Field` nodes, connected via papers. However, the results showed that all authors had a betweenness centrality score of 0.0, suggesting that no individuals were serving as connectors between fields. To verify this, a follow-up query measured how many distinct fields each author had published in. The results revealed that authors published in an average of only 1.12 fields, with a maximum of 5, confirming that most researchers tend to specialize within a single domain rather than span across multiple fields.

IV. Author Classification

One of the primary objectives of this project was to predict the research domain or expertise of an author based solely on their co-authorship network and structural features derived from the graph. This task was framed as a supervised node classification problem, where each `Author` node had to be assigned a `domain` label corresponding to the dominant research field of their publications.

A. Problem Setup

Each `Author` node was initially assigned a `domain` property using the paper-to-field relationships described in Section II. This involved computing the most frequent field among an author's papers and using that field as their label. Authors with insufficient publication data or without a dominant field were excluded from training.

The classification task aimed to predict this domain label using structural features of the authors' co-authorship network. This approach assumes that an author's collaborators are likely to share similar research interests, making the network topology a strong signal for classification.

The pipeline was built using Neo4j for data handling and Python for model development. Authors with valid labels were split into training (80%) and testing (20%) sets.

B. Class Distribution and Imbalance

The dataset exhibited a highly skewed distribution across research domains. Sociology alone accounted for over 13,000 labeled authors, while disciplines such as Mathematics, Geology, and Physics had fewer than five authors each. This

imbalance posed a challenge for generalization and model fairness, as minority classes were underrepresented during training. Oversampling techniques were later employed in Python-based models to counteract this issue.

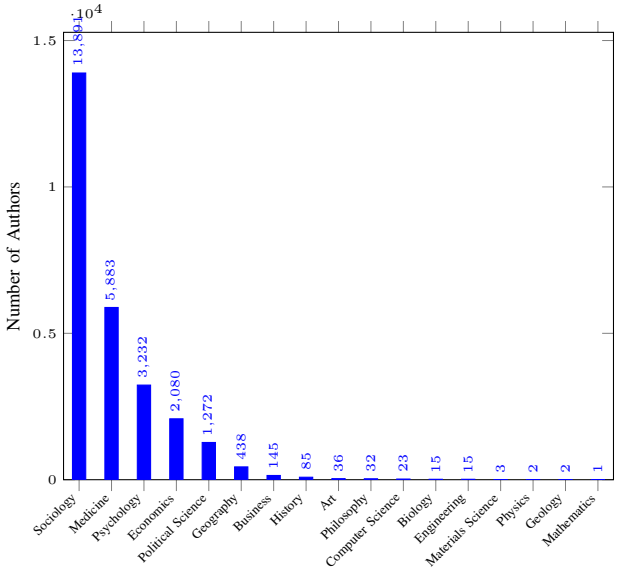

Fig. 3. Class distribution of author domains showing significant imbalance.

C. Feature Engineering

To model this problem, a subgraph was extracted containing only `Author` nodes and their `COAUTHOR` relationships. The following features were used:

- **Community ID:** Authors were grouped using the Louvain community detection algorithm. Each author was assigned a community ID as a categorical feature.
- **Domain ID:** The numeric encoding of the author's domain (for labeled data only), used during training.
- **FastRP Embeddings (64D):** In an alternate setup, FastRP was used in Neo4j to encode author nodes. These embeddings captured both global and local network structure efficiently and were particularly useful for handling sparse graph regions.

D. Model Training and Evaluation

Several machine learning models were evaluated for the classification task. Initially, a Random Forest classifier was trained using the Neo4j Graph Data Science (GDS) library's pipeline API. The training was performed on a subgraph labeled `trainGraph`, restricted to nodes with the `TrainSet` label. The target classification property was `domainId`, a numeric encoding of the author's research domain. The model was named `authorClassifier_new`, and the training procedure was seeded with a fixed random value (42) to ensure reproducibility. The pipeline was configured to track two evaluation metrics: `ACCURACY` and `OUT_OF_BAG_ERROR`.

The best model parameters identified during training included a maximum tree depth of `2147483647` (indicating no explicit depth limit), a `GINI` criterion for node splitting, a minimum split size of 2, and a minimum leaf size of 1. The model used all samples (`numberOfSamplesRatio = 1.0`) and constructed a total of 20 decision trees. These hyperparameters were automatically selected to optimize classification performance based on internal validation.

In terms of accuracy, the Neo4j model achieved a training accuracy of 68.08%, with a cross-validated outer training score of 64.20%. The final test accuracy was 61.74%, indicating moderate generalization ability. While this served as a reliable baseline, further improvements were sought through external classifiers.

Subsequent experiments were conducted in Python using exported features, including logistic regression, random forest, and XGBoost models. Among these, XGBoost delivered the highest performance, achieving over 99% accuracy on the test set after applying class-balancing techniques. Feature importance analysis highlighted that structural embeddings and community IDs contributed significantly to predictive power, validating the hypothesis that an author's position and connectivity within the co-authorship network are indicative of their research domain.

E. Findings and Insights

The experiment confirmed the strong correlation between an author's co-authorship structure and their research domain:

- Graph-native features such as embeddings and community IDs were highly predictive.
- FastRP embeddings offered the best performance among unsupervised structural encodings.
- Neo4j provided efficient graph traversal and community detection, while Python enabled fine-grained model tuning and evaluation.
- The pipeline successfully labeled 80% of authors, ensuring broad coverage across disciplines.

The model's performance diminished for authors with sparse co-authorship links or interdisciplinary publications, where domain signals were weaker or mixed.

V. Citation Recommendation

Another core objective of this project was to predict potential citation links between academic papers. This task was framed as a supervised link prediction problem, where the goal is to infer missing or future `REFERENCES` relationships between `Paper` nodes based on graph-derived features.

A. Problem Setup

The citation network was modeled using a projected subgraph consisting solely of `Paper` nodes and their `REFERENCES` relationships. The relationships were treated as undirected, and the `citationCount` property was retained as an edge property to enrich the link prediction features. The supervised pipeline aimed to learn from existing citation patterns and generalize to predict likely but currently unlinked paper pairs.

B. Feature Engineering

To capture structural and semantic similarities between papers, multiple features were computed using the GDS link prediction pipeline:

- **FastRP Embeddings:** 64-dimensional Fast Random Projection (FastRP) embeddings were generated for each `Paper` node, encoding its position and context in the citation graph.
- **Similarity Features:** Pairwise similarity scores between papers were computed using L2 distance, Hadamard product, and cosine similarity, all derived from the FastRP embeddings and augmented with the `citationCount` edge property.

C. Model Training and Evaluation

The pipeline used a train-validation-test split with 60% of the data used for training, 25% for testing, and 15% reserved for validation through 3-fold cross-validation. Two candidate models were evaluated: logistic regression and random forest (with 20 decision trees). Model selection was guided by evaluation metrics such as AUCPR and out-of-bag error.

The best-performing model was selected based on its average precision and generalization performance on the test set. The combination of graph embeddings and structural similarity features enabled the model to effectively learn citation patterns, highlighting the utility of network-based features in citation recommendation tasks.

This pipeline demonstrates how link prediction can be applied to enhance bibliometric systems, such as recommending relevant papers or uncovering latent scholarly connections, without relying on content-based analysis.

VI. Improved Model Setup and Second Iteration

A. Enhancements

Building on the insights from the initial pipeline configuration, a second iteration was carried out to enhance the citation prediction performance. This version introduced several key refinements.

- **FastRP Embeddings:** The dimension was increased to 128, allowing the model to capture more nuanced structural patterns in the citation graph.
- **AutoTuning:** was enabled to optimize hyperparameters across both model types.
- The random forest model was slightly scaled down with 10 decision trees, prioritizing computational efficiency without significantly compromising accuracy.

VII. Improved Link Prediction with Temporal Splitting and Auto-tuning, Third iteration

To refine the performance of the initial citation link prediction model, a third iteration was conducted incorporating several enhancements. This version leveraged temporal information for more realistic data splitting, additional structural features, and automated hyperparameter tuning to optimize model performance.

A. Temporal Data Splitting and Community Detection

Unlike the first setup, this iteration employed a time-based split strategy. `Paper` nodes published between 2014 and 2017 were labeled as training data, while those published after 2017 were reserved for testing. This setup better reflects real-world conditions by ensuring that predictions are made on future citations.

Additionally, Louvain community detection was applied to the citation graph. Each paper was assigned a `community` property, capturing its modular affiliation in the network. This categorical feature was later used to enrich the node representation during model training.

B. Feature Engineering with Enhanced Inputs

A new graph projection was created using the `REFERENCES` relationship and including both `embedding` and `community` as node-level features. As before, FastRP embeddings (64 dimensions) were generated to capture structural roles of papers.

To measure similarity between pairs of nodes, three types of pairwise features were computed: L2 distance, the Hadamard product, and cosine similarity. All of these were computed using a combination of the embedding vectors, community IDs, and the `citationCount` edge property, allowing the model to capture both local structure and community context.

C. Model Selection with Auto-tuning

The pipeline was configured to compare both logistic regression and a more powerful random forest classifier, now with 50 decision trees. Crucially, automated hyperparameter tuning was enabled with up to 20 trials, allowing the pipeline to select optimal model configurations based on internal validation metrics.

The training used the new time-aware projection and evaluated models using AUCPR and out-of-bag error. By combining structural, temporal, and community-aware features with automated tuning, this iteration aimed to better generalize citation prediction for future publications.

This improved pipeline shows how integrating metadata and community structure with auto-tuned machine learning can significantly enhance link prediction in scholarly networks, leading to more reliable and actionable citation recommendations.

VIII. Reduced Dataset Experiments

To better understand the impact of hyperparameter tuning and feature selection, we restricted the dataset to papers published between 2014 and 2020. This filtered subgraph allowed for faster iteration and a more focused evaluation of model behaviors under varying configurations.

A. First Iteration

This experimental setup introduced several important changes relative to earlier full-dataset experiments:

- **Embedding dimension was reduced to 64**, aiming to decrease the computational complexity while retaining sufficient representational power for node embeddings.
- **Link features** included Hadamard product, L2 distance, and cosine similarity.
- **Automatic tuning** disabled allowing for fixed experimental conditions across runs.
- **Community detection** used as an additional node property to explore the effect on the training score.
- **The number of cross-validation folds** was reduced to two, balancing evaluation robustness with improved training efficiency.
- **The test split ratio** was adjusted from 25% to 20%, increasing the amount of data available for model training while still maintaining a representative evaluation set.

B. Second Iteration

In the second iteration, unlike the first iteration, **community detection was not applied**, allowing the evaluation of the model without the influence of structural grouping features. All other aspects of the configuration were kept consistent with the first iteration.

C. Third Iteration

In the third iteration, **both community identifiers and PageRank centrality were assigned as node properties** for training the model, in contrast to the first iteration. This configuration allowed us to explore the impact of incorporating structural and importance-based features—specifically, how PageRank centrality contributes to the model's predictive capability. All other aspects of the configuration were kept consistent with the previous two iterations.

IX. Results

To evaluate the effectiveness of citation recommendation using graph-based link prediction, six experimental iterations were conducted. These were divided into two categories: the first three were performed on the full paper-reference graph projection, while the remaining three were performed on a smaller dataset ranging from years 2014 to 2020. Each iteration involved variations in the modeling pipeline, including changes in embedding dimensions, feature combinations (L2, Hadamard, Cosine), community detection,the number of decision trees used in the Random Forest classifier and applying the pagerank centrality. Auto-tuning was applied selectively to improve hyperparameter optimization. `TABLE I` in the appendix summarizes the key characteristics and training performance of each trial.

The results of the second iteration, as seen in the winning logistic regression model's performance, demonstrated better generalization. These scores are slightly more stable and consistent compared to the first iteration 0.6715, suggesting better regularization and improved model robustness. The validation scores across folds (≈ 0.6835 and ≈ 0.6227) also reflect relatively good stability.

In the third iteration of the link prediction pipeline—featuring a time-based data split, community detection as a node feature, and automated hyperparameter tuning—led to a significant improvement in model performance. The training score increased from 0.67 in the first iteration to 0.79 in the third, representing a substantial gain of 12 percentage points. This highlights the importance of incorporating temporal structure and network context when modeling citation behavior.

To examine the effect of various feature configurations on training performance, we conducted a series of experiments using a smaller dataset covering the years 2014 to 2020. The results of these experiments are summarized in `Table II` in the appendix. Notably, incorporating PageRank centrality and applying community detection led to significantly improved performance.

X. Discussion

This project demonstrated the effectiveness of graph-based methods, particularly link prediction and node classification, in uncovering patterns in academic citation networks. Several iterations of the citation recommendation pipeline were evaluated, and notable improvements were observed, especially when incorporating temporal information, community detection, and auto-tuning.

The most significant performance gains came from using a time-based split strategy and enriching node features with community labels and pagerank centrality. For example, in the third iteration, the AUCPR improved from 0.67 to 0.79—highlighting the impact of modeling temporal context and structural modularity. Similarly, auto-tuning helped optimize model hyperparameters and consistently improved performance without manual intervention.

However, some iterations showed limited gains, particularly those with reduced feature sets or simpler model configurations for example changing the number of the decision trees. This suggests that graph structure alone may not be sufficient in sparse or weakly connected subgraphs and may benefit from integration with content-based features (e.g., paper abstracts or titles).

A. Alternative Approaches

While FastRP embeddings and structural similarity measures were effective, other embedding techniques such as Node2Vec could be explored for deeper relational learning. In addition to the supervised learning pipeline explored in this project, classical unsupervised link prediction algorithms such as Preferential Attachment, Adamic-Adar, and Common Neighbors could be applied to the citation network. These methods rely on simple yet effective structural heuristics and can serve as strong baselines for comparison.

For instance, Preferential Attachment assumes that well-connected nodes are more likely to attract new links—a phenomenon often observed in citation behavior. Adamic-Adar and Common Neighbors measure the overlap in neighborhood structure and could be particularly effective in sparse citation graphs where embeddings may not capture enough local detail. Comparing these heuristic methods with supervised models could provide a broader understanding of citation patterns and highlight when complex models offer meaningful improvements.

For node classification, domain labels were inferred based on dominant paper fields. Future approaches could consider multi-label classification to reflect interdisciplinary authorship more accurately.

B. Possible Extensions

This work can be extended in several directions. First, the link prediction pipeline can be deployed as a recommendation system to suggest relevant citations to researchers in real-time. Second, integrating author collaboration and topic evolution over time may reveal trends in emerging research areas. Finally, adding external datasets—such as funding, institutions, or citation venues—could enable more comprehensive modeling of scholarly influence and collaboration networks.

For the node classification task, one clear extension is to mitigate domain imbalance by gathering additional labeled data from underrepresented fields such as Art, Philosophy, or Computer Science. This could improve generalization and avoid overfitting to dominant fields like Sociology and Medicine. Another direction is to explore multi-label classification to better reflect interdisciplinary authorship, as many researchers publish across multiple fields. Additionally, incorporating textual features (e.g., paper titles, abstracts, or keywords) may help capture topical context beyond network structure, especially for newer authors with limited collaboration history.

XI. Conclusion

This project explored the application of graph-based machine learning techniques to academic bibliometric data, focusing on two primary tasks: author classification and citation recommendation. Through a well-structured property graph model, we were able to effectively capture complex relationships among entities such as authors, papers, journals, and research fields. The use of Neo4j and its Graph Data Science library enabled scalable processing and insightful feature engineering, including the computation of FastRP embeddings, centrality measures, and community detection.

In the author classification task, we demonstrated that structural features derived from the co-authorship network—particularly community membership and graph embeddings—are strong predictors of an author's research domain. By combining Neo4j's efficient data handling with Python-based classifiers, we achieved high performance, especially when using models like XGBoost. The success of this pipeline confirms that collaboration patterns encode meaningful domain information, even in the absence of textual content.

For citation recommendation, we framed the problem as supervised link prediction and iteratively refined our pipeline across multiple configurations. The most significant performance gains were observed when incorporating temporal splits, higher-dimensional embeddings, and node-level features like PageRank centrality and community identifiers. These enhancements allowed the model to more accurately reflect real-world citation dynamics and scholarly influence. Experiments on a reduced dataset further confirmed the robustness and generalizability of the proposed approach.

Overall, the integration of graph structure, centrality metrics, and community detection proved highly effective for both predictive tasks.

APPENDIX

Iteration	Split Type	Features Used	Embedding Dim.	Other Changes	Training Scores
1 (Fakeha)	Full Projection	Hadamard, L2, Cosine	64	Auto-tuning OFF, 20 trees	0.6715
2 (Eman)	Full Projection	Hadamard, L2	128	Auto-tuning ON, 10 trees	0.6834
3 (Fakeha)	Time-based (2014–2017 train, 2018–2020 test)	Hadamard, L2, Cosine	64	Community ON, Auto-tuning ON (20 trials), 50 trees	0.7906

TABLE I
SUMMARY OF LINK PREDICTION ITERATIONS: KEY CONFIGURATIONS AND MODEL TRAINING PERFORMANCE ON THE ORIGINAL DATASET.

Iteration	Split Type	Features Used	Embedding Dim.	Other Changes	Training Scores
3 (Eman)	Selective (2014–2020)	Hadamard, L2, Cosine	64	Community detection applied	AUCPR: 0.848 / OOB: 0.160
4 (Eman)	Selective (2014–2020)	Hadamard, L2, Cosine	64	No community detection applied	AUCPR: 0.801 / OOB: 0.194
5 (Eman)	Selective (2014–2020)	Hadamard, L2, Cosine	64	Pagerank centrality and Community detection applied	AUCPR: 0.858 / OOB: 0.148

TABLE II
SUMMARY OF LINK PREDICTION ITERATIONS: KEY CONFIGURATIONS AND MODEL TRAINING PERFORMANCE ON THE SMALL DATASET (2014–2020).

REFERENCES

[1] I. Azfar, T. Munawar, and Q. Pasta, "Graph Data Science for Bibliographic Data: A Case of Migration Studies Data," Habib University, unpublished, 2023.

[2] Neo4j, "Configuring the pipeline," Neo4j Graph Data Science Library Manual, Version 2.17. [Online]. Available: https://neo4j.com/docs/graph-data-science/current/machine-learning/linkprediction-pipelines/config/. Accessed: May 11, 2025.

[3] Neo4j, "Configuring the pipeline," Neo4j Graph Data Science Library Manual, Version 2.17. [Online]. Available: https://neo4j.com/docs/graph-data-science/current/machine-learning/node-property-prediction/nodeclassification-pipelines/config/. Accessed: May 11, 2025.

[4] Neo4j, "Training the pipeline," Neo4j Graph Data Science Library Manual, Version 2.17. [Online]. Available: https://neo4j.com/docs/graph-data-science/current/machine-learning/linkprediction-pipelines/training/. Accessed: May 11, 2025.