BRAUDE
College of Engineering, Karmiel

Software Engineering Department

Capstone Project Phase A – 61998

Modeling Citation Trustworthiness in Evolving Scholarly Networks

26-1-R-9

Fatma Dabbah	**213675549**	**fatma.dabbah@e.braude.ac.il**
Manal Nama	**325393999**	**manal.nama@e.braude.ac.il**

Supervisors:

Dr. Dvora Toledano-Kitai

Prof. Zeev Volkovich

GitHub Link: https://github.com/Manal2308/Final-Project.git

Table of Contents

Abstract **3**
1. Introduction **4**
2. Problem Formulation 5
3. Theoretical Background and Preliminaries **5**
3.1 Citation Networks 5
3.2 Embedding 6
3.3 Graph Embedding Methods 6
3.3.1 Node2Vec 6
3.3.2 GraphSAGE 7
3.4 Link Prediction 7
4. Project Requirements **8**
4.1 Functional Requirements 8
4.2 Non-Functional Requirements 8
4.3 Requirements Gathering Methodology 9
5. Model Description **9**
5.1 Structural Stability and Reconstruction 10
5.1.1 Iterative Perturbation 10
5.1.2 Embedding Generation 11
5.1.3 Link Prediction Task 11
5.1.4 Reconstruction Evaluation 11
5.2 Temporal Stability Analysis 12
5.2.1 Snapshot Construction 12
5.2.2 Applying the Structural Stability method per Snapshot 13
5.2.3 Temporal Aggregation 13
5.2.4 Temporal Stability Scoring and Classification 13
5.2.5 Ranking and Analysis 14
6. Implementation Environment and Computational Tools **14**
6.1 Development Environment 14
6.2 Programming Language 14
6.3 Key Libraries and Frameworks 14
7. Evaluation and Testing Plan **15**
7.1 Evaluation Plan 15
7.1.1 Artificial Edge Injection 15
7.1.2 Reconstruction Process 16
7.1.3 Expected Outcome 16
7.1.4 Quantitative Evaluation Metrics 16
7.2 Testing Plan 17
7.2.1 Unit Tests 17
7.2.2 Integration Tests 17
7.2.3 End-to-End System Tests 17

7.2.4 Load Testing 18

8. Expected Challenges 18

8.1 Dynamic Network Evolution 18

8.2 Parameter Sensitivity 18

8.3 Scalability 18

8.4 Lack of "Ground Truth Labels” 18

8.5 Dataset Suitability and Temporal Metadata Availability 19

9. AI Tools and prompts Used 19

9.1 Chat Gpt 19

9.2 google scholar 19

9.3 DeepL 19

References 20

Abstract

This project proposes a framework for assessing citation trustworthiness in large-scale, dynamic scholarly networks. Using citation graphs as a case study, the framework aims to distinguish citations with strong structural support from those with low support or potential manipulative behavior, while capturing meaningful patterns versus random perturbations and tracking how citation structures evolve as the network grows.

The approach combines controlled graph perturbations with embedding-based link prediction.

Citation edges are randomly removed and then reconstructed using node representations to assess their structural stability. Two embedding methods, Node2Vec and GraphSAGE, are compared using complementary static structural and dynamic temporal analyses. In the dynamic method, the structural reconstruction process will be applied iteratively to cumulative snapshots, with stability scores aggregated to capture the network's evolution, including the addition of new papers and citations. Model performance will be then validated through repeated runs with controlled artificial edge injection, and the separation between genuine and injected citation patterns is quantified using AUC-ROC, Detection Rate, and Distribution Distance metrics.

Keywords: Dynamic Citation Networks, Graph Embeddings, Network Perturbations, Temporal Stability, Anomaly Detection.

1. Introduction

The reliability of scientific literature is a fundamental prerequisite for the advancement of research and the development of human knowledge. Citations serve as a central currency in the academic ecosystem: they acknowledge prior work, situate research within its scholarly context, and form the basis for quantitative metrics used to assess the influence of researchers, journals, and academic institutions [7]. However, growing evidence suggests that this system is increasingly compromised by citation manipulation [5,11], whereby references are included not out of scholarly necessity, but to artificially inflate impact metrics, thereby undermining objectivity and integrity in scientific discourse [4].

Citation Manipulation manifests in several sophisticated forms, including citation cartels [5], excessive self-citation, and *forced citation*, in which editors or reviewers pressure authors to cite specific works [6,11]. Empirical studies indicate that a significant portion of references in scientific articles may be unnecessary, underscoring both the scale of the problem and the difficulty of detecting it using traditional approaches [4,6]. Existing methods, largely based on manual review or simple statistical analysis, do not provide an effective response to modern citation networks, which are characterized by enormous size, structural sparsity, and topological complexity [1,2].

The need for advanced algorithmic solutions has intensified in light of the recent developments in academic fraud. Whereas citation manipulation was once done manually and locally, it has increasingly become industrialized, with the emergence of so-called “citation mills” that generate large volumes of artificial publications and citations for commercial purposes [4,5]. By exploiting loopholes in academic indexing platforms, these entities introduce substantial noise into the global citation networks, rendering conventional verification methods insufficient [1,2]. Therefore, there is a growing need for machine learning and graph-cased methods that can identify artificial structural patterns and distinguish them from genuine scholarly activity.

Against this backdrop, recent research has increasingly adopted network-based perspectives relying on the assumption that citations are not isolated entities but part of a broader topological structure.

This project follows this line of work and builds on the *citation stability hypothesis*, which suggests that reliable citations are embedded within dense networks of indirect connections and thus form a stable core of the citation graph [1,2]. In contrast, manipulative citations tend to lack such structural support and are therefore more sensitive to perturbations in the network.

2. Problem Formulation

Given a large-scale, evolving citation network, the objective is to assess the reliability of individual citation links in the absence of explicit ground-truth labels. The task is to distinguish citations that reflect well-established scholarly relationships from those that are potentially manipulative or structurally weak, by formulating citation reliability as a structural and temporal inference problem, rather than relying on local or counting-based indicators such as citation frequency or journal impact metrics.

Formally, given a citation graph that evolves over time, each citation edge is assigned a stability-based reliability score reflecting its structural support and temporal consistency within the network. Citations that exhibit consistent reconstructability across snapshots are considered reliable, while edges showing instability or anomalous temporal behavior are flagged as suspicious. This formulation enables the detection of abnormal citation patterns without relying on manual inspection or predefined labels.

The remainder of this document is organized to guide the reader through the project's conceptual foundations, methodology, and implementation. **Section 2** covers the theoretical background, including key concepts in citation networks and graph embedding. **Section 3** outlines the project requirements. **Section 4** describes the model and its methods for structural and temporal analysis. **Section 5** outlines the implementation environment. **Section 6** details the evaluation and testing approach. **Section 7** discusses expected challenges and mitigation strategies. Finally, **Section 8** lists the AI tools and prompts utilized during the research process.

3. Theoretical Background and Preliminaries

3.1 Citation Networks

A citation network is a formal mathematical model that represents the complex relationships of “who cites whom” within a collection of publications, such as academic articles. Unlike a flat bibliographic list, a citation network is represented as a graphical structure that allows for in-depth analysis of knowledge flow, research influence, the formation of subfields, and the identification of unusual patterns and unnatural behavior in the academic system [7].

In this network, each node represents a single publication, and an edge $A \rightarrow B$ indicates that article A cited article B. Formally, the network is defined as a directed graph $G = (V, E)$,where V represents the collection of publications and E the set of citation links, where and edge $(u, v) \in E$ exists if and only if publication u cited publication v.

The presence of a citation does not necessarily indicate quality, scientific contribution, or authentic research contribution; Articles may cite each other for a variety of reasons, such as criticism, strategic reasons, or due to phenomena like self-citation, forced citation, or citation cartels [5,11]. Citation network analysis often combines statistical tools, structural analysis, chronological examination, and sometimes textual context analysis to identify unusual patterns [1].

3.2 Embedding

Embedding is a key technique in machine learning that allows discrete and diverse objects, such as text, images, audio, or graph nodes, into a continuous, relatively low-dimensional vector space R^d [8,9].

While many machine learning algorithms are limited to accepting only numerical input, raw data (such as words or network connections) does not come in this format. Embedding bridges this gap by converting objects into information-rich vectors. The location of each point in the vector space is not random but captures meaningful semantic or structural relationships: objects that are similar in terms of content or structural role will be located in geometric proximity to each other.

In citation networks, embedding allows nodes to be represented numerically while preserving structural or semantic similarity: nodes with similar roles or context in the network appear close in the embedding space. Unlike manual feature engineering, these vectors are learned automatically using methods such as neural networks or random walks [8,9]. Similarity between nodes is typically measured via cosine similarity or Euclidean distance.

3.3 Graph Embedding Methods

This project uses two complementary graph embedding methods:

3.3.1 Node2Vec

Node2Vec [8], is a transductive method based on biased random walks skip-gram modeling. The method converts the graph into sequences of nodes, similar to sentences in text, and learns the vector representation using a skip-gram model. Hyperparameters p (return) and q (in–out) control the balance between local and global exploration. Node2Vec is mainly suitable for static networks.

3.3.2 GraphSAGE

GraphSAGE [9], is an inductive embedding algorithm for learning on graphs. Unlike Node2Vec, which learns a separate vector representation for each existing node, GraphSAGE learns a general function that generates embeddings for both existing and new nodes, based on the node's properties and the local structure of its neighbors in the graph. The algorithm works by sampling a fixed number of neighbors for each node and aggregating the information from them (such as average, LSTM, or pooling), which is combined with the representation of the node itself to create a rich embedding that describes both its characteristics and its structural context in the network. This process can be performed in multiple layers, with each layer expanding the node's field of view to more distant neighbors.

This approach makes GraphSAGE scalable and particularly suitable for dynamic graphs.

These methods allow modeling citation stability by capturing both local neighborhood structure and broader network topology. This approach builds on prior work in anomaly detection and link prediction in dynamic networks, leveraging structural patterns to identify unstable or unusual connections [1,2,10].

3.4 Link Prediction

Link prediction is a central task in network analysis, aimed at estimating the likelihood of a connection between two nodes that are not directly linked in the network. In citation networks, it predicts whether a genuine citation relationship exists between two papers, even if the corresponding edge is missing or temporarily removed [10]. This approach is based on the assumption that the network’s topological structure, together with indirect connections, contains sufficient information to infer latent links. High similarity between node embeddings indicates a high likelihood of a connection [8,9].

In this project, link prediction will be employed to evaluate citation stability: reliably supported citations can be consistently reconstructed, whereas manipulative or structurally weak citations tend to be less predictable and exhibit lower reconstruction stability [2]. Metrics such as AUC-ROC, Detection Rate, and Distribution Distance can be used to quantify the separation between genuine and artificial citation patterns, providing a principled measure of the model’s discrimination capability.

4. Project Requirements

This section presents the main requirements of the project, divided into **functional** and **non-functional** requirements.

4.1 Functional Requirements

- The system shall load citation datasets.
- The system shall construct a citation network from the input data.
- The system shall generate node embeddings using Node2Vec and GraphSAGE.
- The system shall perform controlled perturbations by removing citation edges from the network.
- The system shall reconstruct removed citation edges using a link prediction task.
- The system shall compute a structural stability score for each citation independently in each temporal snapshot.
- The system shall aggregate stability scores across all temporal snapshots.
- The system shall rank citation edges based on the aggregated stability scores.
- The system shall inject artificial edges into the network to simulate manipulative behavior.
- The system shall compare stability results obtained using Node2Vec and GraphSAGE embeddings.
- The system shall distinguish between structurally supported citation and injected artificial noise, based on stability scores.

4.2 Non-Functional Requirements

- The system shall ensure reproducibility by generating Initial node feature vectors using predefined topological properties (e.g., node degree or centrality measures) or via random initialization when content-based features are unavailable.
- Node embeddings shall be generated in a continuous vector space with fixed dimensionality.
- Perturbation shall be performed by randomly removing a predefined fraction of citation edges in each iteration.
- Citation edge reconstruction shall be based on embedding similarity within a link prediction framework.
- Stability scores shall be aggregated across multiple perturbation iterations and temporal snapshots to improve robustness.
- The comparison between Node2Vec and GraphSAGE shall use consistent evaluation metrics and experimental settings.

- Embedding models shall be trained using consistent hyperparameters across all temporal snapshots.
- Artificial edges shall be injected in a controlled and statistically consistent way.
- System Performance shall be measured quantitatively using AUC-ROC, detection rate, and distribution-based metrics.
- Multiple experimental runs shall be performed to reduce sensitivity to random perturbations and ensure reproducibility.
- The system shall be designed to support scalability to large-scale and dynamic citation networks.

4.3 Requirements Gathering Methodology

The project requirements were derived using a dual approach combining a systematic literature review with expert guidance. The primary functional scope was defined based on the methodologies proposed in prior studies on citation network perturbation and stability analysis [1,2].

These research-based insights were further validated and refined through consultations with our supervisors, who possess expertise in citation network analysis and experience with related studies. We anticipate that this combined approach ensures that both the functional and non-functional requirements will accurately reflect state-of-the-art practices in citation anomaly detection, while remaining aligned with the specific goals of the project.

5. Model Description

Consistent with the citation stability hypothesis [1,2], the proposed solution evaluates citation reliability by examining the robustness of links under controlled perturbations of the citation network. Specifically, a subset of edges is randomly removed, after which the system attempts to reconstruct them using a link prediction task [10]. The underlying assumption, supported by recent studies on network stability [2,3], is that citations belonging to the stable structural core of the network will be consistently reconstructed across multiple perturbations, whereas structurally weak or manipulative citations will exhibit lower reconstruction stability.

To this end, the project will apply graph embedding techniques to represent nodes in a continuous vector space and compare two main approaches: Node2Vec [8], a transductive method primarily suited to static graphs, and GraphSAGE [9], an inductive method capable of generalizing to newly added nodes in dynamic networks. This comparison allows an assessment of how different embedding paradigms contribute to modeling citation reliability in evolving academic graphs.

After training the model and generating embeddings for each node, the system proceeds to the stage of reconstructing the edges that were removed during the perturbation phase, which is formulated as a link prediction task. For each pair of nodes whose indirect connection has been deleted, a similarity measure between their corresponding vector representations is calculated in order to assess how well the connection is supported by the learned network structure [8]. Edges that receive high similarity scores are interpreted as being well supported by underlying network patterns and are therefore considered more reliable, while low similarity scores indicate weaker structural support and may suggest anomalous behavior [2,10].

To strengthen the reliability of the inference and reduce dependence on a single random removal, the perturbation and reconstruction process is performed across multiple runs and at different removal rates [1,2]. This procedure allows each edge to be assigned a stability index based on the consistency of its reconstruction across experiments [2]. Edges that are consistently reconstructed under different perturbations are classified as stable and reliable, while edges that exhibit frequent reconstruction failure or high variability are flagged as potentially suspicious. Finally, edges are ranked according to their level of reliability, enabling the identification of abnormal connections and the examination of potential citation manipulation within the network [5,11].

The model is implemented using two complementary methods, enabling the analysis of both the static structure of the citation network and its temporal evolution.

5.1 Structural Stability and Reconstruction

This phase focuses on examining the "structural necessity" of each citation at a given point in time. The approach relies on the hypothesis that legitimate citations within the undirected citation graph are supported by a wide network of direct connections; therefore, the topological information remaining in the network should enable their reconstruction even in the absence of the direct link [1,2]. In contrast, manipulated citations, which are introduced artificially, do not conform to the natural structural patterns of the network and therefore tend to demonstrate inconsistency under perturbations [1,2].
The identification process is performed in four main steps:

5.1.1 Iterative Perturbation

The system performs N iterations. In each iteration, a fixed fraction (Fr) of the edges in the original graph G is randomly removed [1]. The resulting graph, denoted G_temp, serves as the training graph, while the removed edges are defined as the test set to be reconstructed [1].

5.1.2 Embedding Generation

The model maps the nodes in the partial graph G_temp to a continuous vector space of low dimension [1]. The study compares two graph embedding approaches:

- Node2Vec: A method based on biased random walks to capture the topological neighborhood of each node [8]. This method is suitable for analyzing the static structure of the network [1].
- GraphSAGE: A method that generates node representations using neighborhood sampling and aggregation [9]. This capability allows the model to learn transferable structural patterns and generalize them to evolving networks [10].

5.1.3 Link Prediction Task

At this stage, the model evaluates its ability to reconstruct the removed edges based on the remaining structural information [10]. For each edge (u, v) in the test set, the following procedure is executed:

- Similarity Calculation: The system calculates a similarity measure between the embedding vectors of nodes u and v.
- Thresholding: The similarity score is compared against a predefined threshold (Tr) [9]. If the score exceeds the threshold ($Sim(u, v) > Tr$), the link is classified as "existing"; otherwise, it is classified as "non-existing" [10]. This thresholding step can serve as a form of statistical hypothesis test, where low similarity implies rejecting the hypothesis of a strong structural connection [1,2].
- Counter Update: Following the classification, the system records the outcome of the current iteration. If the link is successfully reconstructed, a cumulative success counter associated with that edge is incremented. This mechanism tracks the reconstruction performance of each edge across all N perturbations, providing the empirical basis for the subsequent stability assessment.

5.1.4 Reconstruction Evaluation

At the end conclusion of all N iterations, a *stability score* is calculated for each edge, defined as the ratio between the number of successful reconstructions and the total number of perturbation runs in which the edge was evaluated [1].

Edges displaying a consistently low reconstruction score, typically located in the lower percentiles of the score distribution, are classified as "suspicious", as they lack the structural support characterizing genuine citations relationships [1,2].

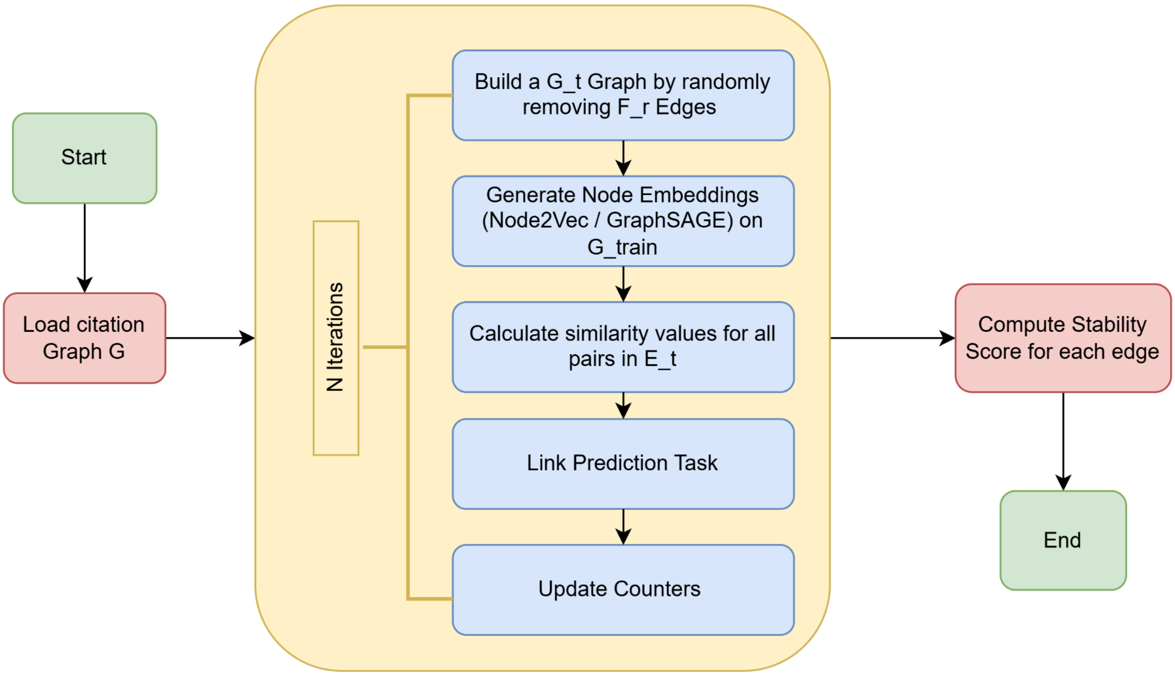
Start
Load citation Graph G
N Iterations
Build a G_t Graph by randomly removing F_r Edges
Generate Node Embeddings (Node2Vec / GraphSAGE) on G_train
Calculate similarity values for all pairs in E_t
Link Prediction Task
Update Counters
Compute Stability Score for each edge
End

Figure 1. Flowchart of the Structural Stability Model.

5.2 Temporal Stability Analysis

While the structural stability approach assesses the network as a static entity, citation networks are inherently dynamic systems that evolve over time. New papers are published, and new citation links are formed, continuously altering the topological structure. To capture this evolution, the second model introduces a temporal dimension to the analysis.

This method expands on the structural stability framework described in the first layer by explicitly incorporating the temporal dynamics of the citation network. While the structural approach examines the necessity of citation edges at a given point in time, the temporal approach analyzes the consistency of edge reconstruction across a sequence of time periods in order to identify anomalous citations within an evolving developmental context [1,2].

The identification process is performed in five main steps:

5.2.1 Snapshot Construction

In the first stage, the citation network is divided into cumulative snapshots. Each snapshot represents the state of the citation network at a specific time period and includes:

- Nodes: Academic articles.
- Indirected edges: Citation links between articles that exist up to the corresponding time window.

This construction provides a chronological representation of the network's evolution, while preserving the underlying citation structure [1,2].

5.2.2 Applying the Structural Stability method per Snapshot

For each snapshot, the structural stability method is applied independently to the corresponding subgraph. Specifically, for every snapshot, the four-stage procedure defined in Section 5.1 is executed:

- Iterative perturbation of citation edges.
- Embedding generation using Node2Vec and GraphSAGE.
- Link prediction for reconstructing removed citations.
- Computation of a structural stability score for each citation within the snapshot.

This process yields a snapshot-level stability score for every citation edge, for each embedding method.

5.2.3 Temporal Aggregation

After applying the structural stability method to all snapshots, the stability scores obtained for each citation across time windows are aggregated. These scores capture the temporal behavior of each citation and reflect the degree of consistency of its reconstruction under evolving topological structures [1,2].

5.2.4 Temporal Stability Scoring and Classification

Based on the aggregated results, a temporal stability score is calculated for each citation, representing the degree of consistency of citation recovery across multiple time windows:

- Stable citations: Citations that are consistently recovered across multiple snapshots are classified as stable and reliable.
- Suspicious citations: Citations that exhibit repeated reconstruction failures or high volatility across snapshots are classified as suspicious.
- Delayed-impact citations: Citations that show low or no recovery in early snapshots but become consistently recovered following a clear awakening phase (e.g., *Sleeping Beauty* or delayed-Impact behavior) are classified as *delayed-impact* rather than suspicious by default. This classification reflects legitimate late recognition, unless accompanied by additional anomalous indicators, such as unstable recovery patterns, concentration of citations from a limited group of sources, or abnormal behavior relative to the overall network structure.

5.2.5 Ranking and Analysis

Finally, citation edges are ranked according to their temporal stability scores. This ranking enables the identification of abnormal citation patterns and supports the analysis of potential manipulation or structurally weak citations within the evolving citation network [4,5,11].

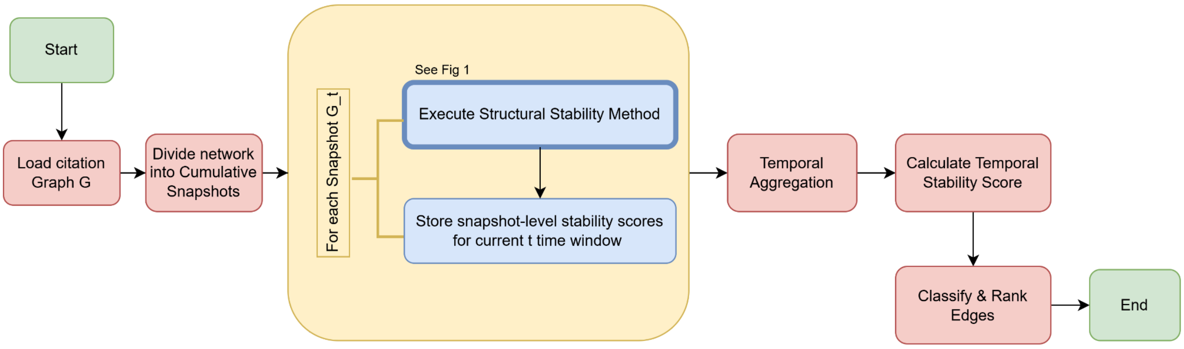
Start
Load citation Graph G
Divide network into Cumulative Snapshots
For each Snapshot G_t
See Fig 1
Execute Structural Stability Method
Store snapshot-level stability scores for current t time window
Temporal Aggregation
Calculate Temporal Stability Score
Classify & Rank Edges
End

Figure 2. A flowchart of the Temporal Stability Analysis.

6. Implementation Environment and Computational Tools

The implementation, training, and testing of the proposed system will be conducted using a modern data science environment, selected for its efficiency in processing large-scale graphs and supporting deep learning workflows.

6.1 Development Environment

- **Platform:** Google Colab.

  The project is planned to be executed on the Google Colab cloud-based platform, selected for its accessibility and availability of hardware acceleration.

6.2 Programming Language

- **Language:** Python

  Python will serve as the core programming language due to its widespread adoption in machine learning and network science, as well as its rich ecosystem of optimized libraries for graph analysis and representation learning.

6.3 Key Libraries and Frameworks

The system is designed to rely on several open-source libraries, each supporting a specific stage of the analysis pipeline:

- **Network Analysis:**
    - **NetworkX:** Will be used for the construction and manipulating the citation graph,as well as for calculation of basic topological properties.
- **Graph Representation Learning:**
    - **PyTorch Geometric (PyG):** A deep learning library built on PyTorch for implementing GraphSAGE model and efficiently handling irregular graph structures through message-passing.
    - **PyTorch:** Serves as the core underlying deep learning framework for training GraphSAGE models via PyTorch Geometric.
    - **Gensim:** Will be utilized for the efficient implementation of the **Node2Vec** algorithm, leveraging its optimized random walk and Word2Vec capabilities to generate static node embeddings.
- **Model Evaluation & Machine Learning:**
    - **Scikit-Learn:** Will be used for the link prediction task and for calculating performance metrics such as AUC-ROC, Precision, and Recall.
    - **SciPy:** Will be used for statistical computations and for measuring distribution distances between stability score distributions.
- **Data Handling & Visualization:**
    - **Pandas & NumPy:** For efficient data manipulation and numerical matrix operations.
    - **Matplotlib & Seaborn:** Will be used for visualizing stability score distributions illustrating the temporal evolution of the citation network.

7. Evaluation and Testing Plan

7.1 Evaluation Plan

The evaluation aims to validate the model’s capability to distinguish between justified, genuine citations and artificial or manipulated connections. Since real-world datasets typically lack explicit "ground truth" labels for manipulation, the evaluation strategy relies on a controlled **synthetic noise injection** process.

7.1.1 Artificial Edge Injection

In the first stage, we proactively inject p% artificial edges into the original citation network. These edges connect pairs of articles that have no genuine citation relationship or structural proximity, simulating manipulative behaviors such as random noise or organized "citation cartels". By systematically varying **p%**, we examine the injection level at which the model produces stable and consistent results, resulting in a mixed network containing both authentic and artificial connections.

7.1.2 Reconstruction Process

Following noise injection, the two proposed methods, based on **Node2Vec** and **GraphSAGE** embeddings, are applied to the mixed network. During the perturbation phase, both original and artificially injected edges are treated identically: they are randomly removed from the graph, and the system attempts to reconstruct them using link prediction based on the learned embeddings.

7.1.3 Expected Outcome

The validation focuses on contrasting reconstruction behavior between the two edge groups:

- **Artificial Citations:** Expected to exhibit **low reconstruction rates** and fail to be consistently recovered, as these edges were added without structural justification, removing them leaves no topological traces for the model to infer their existence.
- **Original Citations:** Expected to demonstrate **high stability scores** as they are supported by the natural topological structure of the scientific community.

Crucial Note:

It is hypothesized that a small subset of original citations may also exhibit low stability scores, similar to artificial edges. These cases are not considered as model errors but rather as candidates for **potential real-world manipulations** or structurally weak citations.

7.1.4 Quantitative Evaluation Metrics

To objectively assess the model's performance, the following metrics will be used to evaluate the separation between original and artificial distributions:

- **AUC-ROC (Area Under the Curve):** The primary metric, measuring the probability that a randomly selected original citation receives a higher stability score than a randomly chosen artificial edge. Values significantly above 0.5 indicate effective anomaly detection.
- **Detection Rate (Recall on Artificial Edges):** The proportion of injected artificial edges correctly identified as suspicious (i.e., received a stability score below the classification threshold).
- **Distribution Distance:** A statistical comparison of the stability score distributions of original versus artificial edges, assessing the degree of separation between authentic and manipulated patterns.

7.2 Testing Plan

The testing plan ensures correctness, robustness, and reproducibility of the implemented pipeline. It covers all main modules, from data ingestion and graph construction, through embedding generation (Node2Vec/ GraphSAGE), perturbation/noise injection, stability computation, to result export, verifying expected behavior under standard and edge-case conditions.

7.2.1 Unit Tests

Verify individual modules in isolation:

- **Graph construction:** verify indirected edge creation, node/edge counts, invalid/self-loop removal, and connectivity statistics.

- **Perturbation module:** confirm injecting p% edges produces the expected number of new edges, avoids duplicates, respects valid node pairs, and preserves graph validity.

- **Metrics module:** validate numerical ranges and stability-score computation.

7.2.2 Integration Tests

Verify correct module interactions:

- **Graph → Embeddings:** ensure graph is correctly consumed by Node2Vec/GraphSAGE and embeddings match configured dimensions for intended nodes.

- **Embeddings → Link prediction:** verify that reconstructed edge scoring runs successfully and returns outputs are consistent.

- **Reconstruction → Stability evaluation:** verify that stability scores are computed and stored correctly for both original and injected edges.

7.2.3 End-to-End System Tests

Validate the complete pipeline on a controlled dataset:

- Execute full workflow: load → build graph → inject noise → embed → reconstruct → compute stability → export results.

- Confirm that all expected artifacts are produced, and that the output schema remains consistent across runs.

7.2.4 Load Testing

Assess the system under extreme conditions:

- Run the pipeline with high injection/removal rates p% (i.e., substantial edge perturbations) and observe the system’s behavior.

- Evaluate challenging graph settings, such as extremely sparse graphs, a large number of isolated nodes, or a high number of temporal snapshots.

- Expected outcome: the system completes successfully, or terminates gracefully with clear error/warning messages, avoiding silent failures or misleading partial outputs.

8. Expected Challenges

8.1 Dynamic Network Evolution

Citation networks continuously evolve over time. Capturing meaningful temporal patterns requires careful construction of temporal snapshots and appropriate aggregation of stability scores across time.

8.2 Parameter Sensitivity

The model's performance relies heavily on multiple hyperparameters, such as perturbation fraction (Fr), similarity threshold (Tr), embedding dimensionality, and number of iterations (N). Choosing appropriate values is critical, as improper tuning may lead to overfitting or reduced sensitivity to anomalous behavior.

8.3 Scalability

Processing large-scale citation networks with Node2Vec and GraphSAGE embeddings, combined with multiple perturbation runs and temporal snapshots, may exceed the memory and computation limits of standard environments like Google Colab.

8.4 Lack of "Ground Truth Labels”

Real-world citation datasets typically lack explicit labels indicating which citations are manipulative. This limitation complicates the validation of the model’s performance on genuine anomalies beyond the injected synthetic noise.

8.5 Dataset Suitability and Temporal Metadata Availability

Constructing temporal snapshots of a citation network requires reliable temporal metadata such as publication year or citation timestamp. However, many public datasets contain incomplete or inconsistent temporal fields, which can hinder accurate modeling of network evolution. Addressing this issue may require additional preprocessing or the use of coarser time granularity, potentially reducing the sensitivity of the analysis.

9. AI Tools and prompts Used

9.1 Chat Gpt

We use ChatGPT to explain concepts clearly and help us understand complex topics.
https://chatgpt.com/share/696fc2c5-0654-8006-b9b0-96ddd8baa003
https://chatgpt.com/share/696fc773-b110-8006-ad2c-badcdbe51f05

9.2 google scholar

We use it to search for academic papers.

https://scholar.google.com/scholar?hl=ar&as_sdt=0%2C5&q=Citation+Networks&btnG=

https://scholar.google.com/scholar?hl=ar&as_sdt=0,5&q=Embeddings

https://scholar.google.com/scholar?hl=ar&as_sdt=0%2C5&q=GraphSAGE&btnG=

https://scholar.google.com/scholar?hl=ar&as_sdt=0%2C5&q=Node2VEC&btnG=

https://scholar.google.com/scholar?hl=ar&as_sdt=0%2C5&q=link+prediction+in+social+networks&oq=link+prediction

https://scholar.google.com/scholar?hl=ar&as_sdt=0%2C5&q=link+prediction&btnG=

9.3 DeepL

We use it to translate academic words and phrases.
https://www.deepl.com/en/translator

References

1. **Avros, R.**; Keshet, S.; Toledano Kitai, D.; Vexler, E.; Volkovich, Z. Detecting Pseudo-Manipulated Citations in Scientific Literature through Perturbations of the Citation Graph. *Mathematics*, 11(18), 3820, 2023.
2. **Avros, R.**; Toledano Kitai, D.; Volkovich, Z. Citation Steadiness Analysis with GraphSAGE Approach. Proceedings of the 14th International Conference on Data Science, Technology and Applications (DATA), 769-777, 2025.
3. **Benatti, A.**; de Arruda, H. F.; Silva, F. N.; Comin, C. H.; da Fontoura Costa, L. On the stability of citation networks. *Physica A: Statistical Mechanics and its Applications*, 610, 128399, 2023.
4. **Biagioli, M.** Watch out for cheats in citation game. *Nature*, 535(7611), 201, 2016.
5. **Fister Jr, I.**; Fister, I.; Perc, M. Toward the discovery of citation cartels in citation networks. *Frontiers in Physics*, 4, 49, 2016.
6. **Fong, E. A.**; Wilhite, A. W. Authorship and citation manipulation in academic research. *PLOS ONE*, 12(12), e0187394, 2017.
7. **Garfield, E.** Citation indexes for science: A new dimension in documentation through association of ideas. *Science*, 122(3159), 108–111, 1955.
8. **Grover, A.**; Leskovec, J. Node2vec: Scalable Feature Learning for Networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD ’16)*, San Francisco, CA, USA, August 13–17, 2016; pp. 855–864.
9. **Hamilton, W.**; Ying, Z.; Leskovec, J. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems (NIPS)*, Vol. 30, 2017.
10. **I, B. K.**; Mathi, A. R. P.; Sett, N. Evaluating link prediction: New perspectives and recommendations. *arXiv preprint arXiv:2502.12777*, 2025.
11. **Wilhite, A. W.**; Fong, E. A. Coercive citation in academic publishing. *Science*, 335(6068), 542–543, 2012.