Software Engineering Department Braude

Academic College

Capstone Project Phase A – 61998

Detecting and Evaluating Anomalously Cited Papers Over Time using Anomaly Detection in Dynamic Graphs via Transformer

24-2-R-7

Israel Shushan
Israel.Shushan@e.braude.ac.il

Noah Soskha
Noah.Soskha@e.braude.ac.il

Supervised by

Prof. Zeev Volkovich

Dr. Renata Avros

Book Repository

Detection-of-Anomalous-Cited-Papers

Table of Contents

1. Abstract....4
2. Introduction....4
3. Literature review....5
   3.1 Literature Review Chapter....5
   3.2 Early Approaches: Shallow Learning-Based Methods....6
   3.3 Deep Learning Methods....6
   3.4 Transformer Models in Dynamic Graphs....6
   3.5 Key Contributions of Transformer-Based Models....7
   3.6 Conclusion....7
4. Background....8
   4.1 Graph Theory and Dynamic Graphs....8
   4.2 Subgraphs and Graph Diffusion....8
   4.3 Node and Edge Embeddings....8
   4.4 Discriminative Knowledge....9
   4.5 Spatial and Temporal Information....9
   4.6 Citation Network....10
   4.7 Neural Networks and Transformer Models....10
   4.8 Attention Mechanisms....10
   4.9 TADDY: Transformer-based Anomaly Detection for Dynamic Graphs....11
5. Mathematical Background....11
   5.1 Graph Theory....11
      5.1.1 Dynamic Graphs....11
      5.1.2 Subgraph and Substructure....11
      5.1.3 Graph Diffusion....12
      5.1.4 Personalized PageRank (PPR)....12
      5.1.5 Heat Kernel....13
   5.2 Node Encoding....13
      5.2.1 Diffusion-based Spatial Encoding....13
      5.2.2 Distance-based Spatial Encoding....13

5.2.3 Relative Temporal Encoding 14
5.3 Transformer Model 15
5.3.1 Attention Mechanism 15
5.3.2 Multi-head Attention 15
5.4 Node and Edge Embedding 16
5.4.1 Node Embedding 16
5.4.2 Edge Embedding 17
5.5 Anomaly Detection 17
5.5.1 Anomaly Score Calculation 17
5.5.2 Binary Cross-Entropy Loss 17
5.6 Negative Sampling 18
6. Expected Achievements 18
7. Research Process 19
7.1 Model Overview 19
7.2 Dataset 19
7.3 Model Architecture 20
7.3.1 Initialization 21
7.3.2 Embedding 21
7.3.3 Training Stage 21
7.3.4 Testing Stage 21
7.3.5 Model Evaluation 22
8. Expected challenges: 23
8.1 Computation and Storage of Dynamic Data: 23
8.2 Threshold Adjustment for Anomaly Detection: 23
8.3 Interpreting Model Results: 23
8.4 Processing Time: 24
8.5 Adapting a Dynamic Citation Dataset to the Requirements of the TADDY Algorithm 24
9. Evaluation/Verification Plan 25
10. References 26

1. Abstract

In academic research, the integrity of citations is essential to the validity and reliability of scholarly work. However, the increasing complexity and volume of citations have led to challenges ensuring that all references are relevant and contribute meaningfully to research. The main problem lies in detecting anomalous cited papers that are irrelevant or manipulative. To address this challenge, we propose a systematic approach that uses anomaly detection techniques. Specifically, we utilize TADDY (Liu, et al., AUGUST 2015), a transformer-based anomaly detection framework tailored for dynamic graphs, within a dynamic citation network—where papers are nodes and citations are edges—we apply TADDY to generate an "anomaly score" for each paper in our dataset. This approach effectively identifies anomalous citations and adapts to changes over time. Our results demonstrate that TADDY enhances the integrity of academic references by reliably detecting irrelevant or manipulative citations.

Keywords: Citation Anomalies, Citation Network, Dynamic Graphs, Transformer Models, Anomaly Detection

2. Introduction

Citation networks are used in measuring the scholarly impact of academic papers (BAI, ZHANG, NI, SHI, & LEE, Jan 2020). These networks consist of nodes and edges, where papers are represented as nodes and the citations between them as directed edges. Anomalous citations—those that deviate from standard citation patterns—can indicate irregularities or manipulations in how papers reference each other. Examples include citations that do not logically support the citing paper's claims, those aimed at artificially boosting citation metrics, or those that are inappropriate given the paper's topic.

Detecting these citation anomalies is crucial for maintaining the authenticity and relevance of scholarly communication (Chakraborty, Chakraborty, Nandi, Dutta, & Pradhan, JUL 2018). Irrelevant references and manipulative citation practices threaten the integrity of academic work. Such anomalies can skew the perceived impact of research, mislead readers and researchers, and potentially create a skewed academic environment where true contributions are overshadowed by manipulated citation figures.

Our study focuses on detecting and analyzing these anomalies to determine how they evolve over time. As new papers are published, what may initially appear as an anomalous citation can eventually be justified, remain anomalous, or become non-anomalous as the dataset evolves. Understanding this evolution helps determine the true nature of initially anomalous citations.

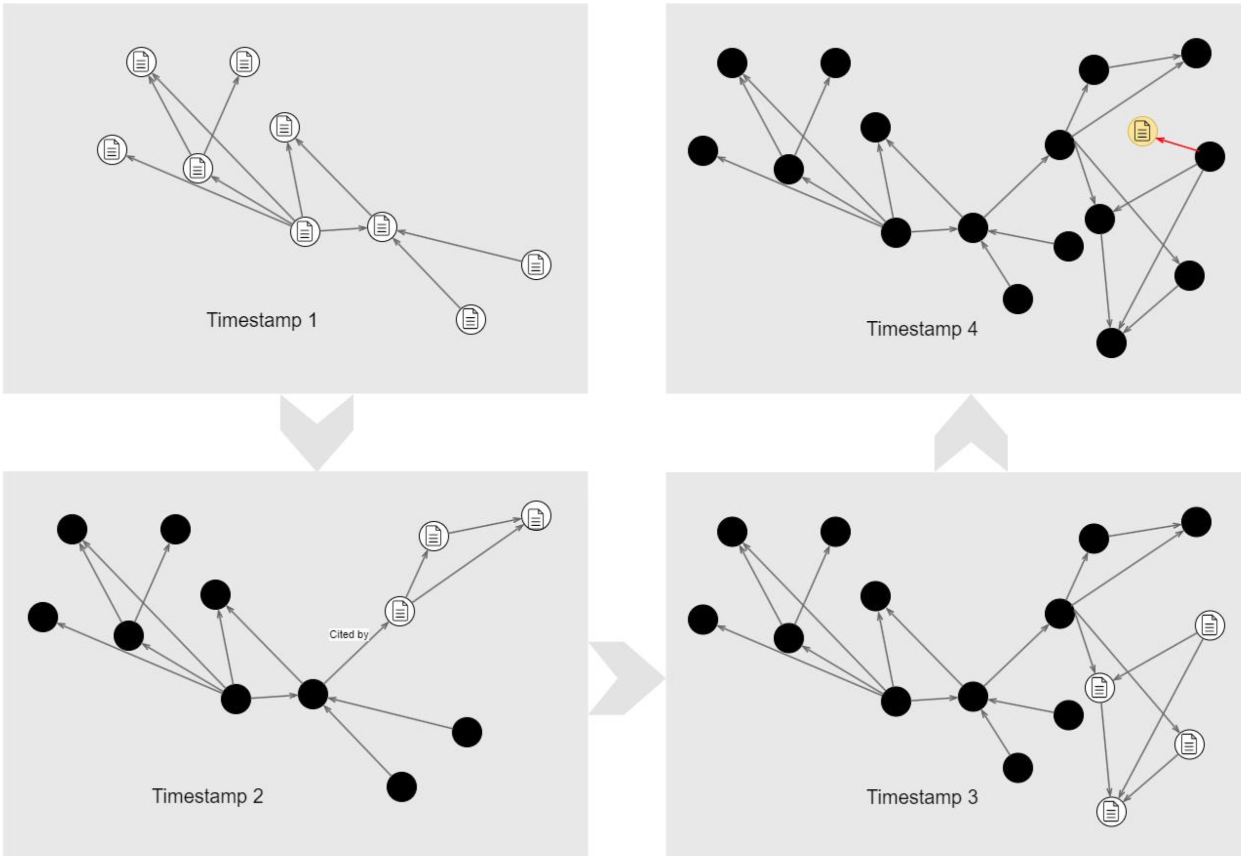

Figure 2-1: *a dynamic graph representation of a citation network with directed edges, showing the network's evolution over multiple timestamps. In the final timestamp (Timestamp 4), an anomalous cited paper is highlighted with a red edge, indicating suspicious or irregular citation behavior.*

3. Literature review

3.1 Literature Review Chapter

Anomaly Detection in Dynamic Graphs: A Transformer-Based Approach

Anomaly detection in dynamic graphs has gained considerable attention in recent years, primarily due to its widespread applications in areas such as social networks, cybersecurity, and e-commerce. The problem lies in identifying unusual patterns or behaviors that deviate from normal patterns in an evolving graph structure. Over the past decade, several methods, both shallow and deep learning-based, have been developed to address this issue.

3.2 Early Approaches: Shallow Learning-Based Methods

Early work in anomaly detection in dynamic graphs primarily relied on shallow learning approaches. These methods typically used handcrafted features, often focusing on graph connectivity and behavior modeling. One example is GOutlier (Aggarwal, Zhao, & Philip, 2011), which employed a structural connectivity model to detect anomalies by analyzing the connectivity behaviors of nodes in graph streams. Similarly, CM-Sketch (Ranshous, Harenberg, Sharma, & Samatova, 2016) introduced an edge anomaly detection model that utilized a sketch-based approach to track both local structural changes and historical behaviors of the nodes. These methods, while pioneering, suffered from limited scalability and inadequate performance in large-scale dynamic graphs.

3.3 Deep Learning Methods

The shortcomings of shallow methods paved the way for deep learning-based approaches. Among these, NetWalk (Yu, et al., 2018) made significant strides by using a random walk-based dynamic graph embedding technique combined with a clustering-based detector for anomaly detection. This method marked a turning point as it leveraged deep learning to generate node embeddings capable of capturing complex dynamic behaviors.

Subsequent research expanded on this foundation. AddGraph (Zheng, Li, Li, Li, & Gao, 2019) combined Graph Convolutional Networks (GCNs) for spatial information extraction with Gated Recurrent Units (GRUs) for temporal pattern recognition. The approach provided an end-to-end solution for detecting anomalies in dynamic graphs. However, a critical limitation of these models was their reliance on separate modules for spatial and temporal learning, which often failed to capture the coupled nature of the spatial-temporal information inherent in dynamic graphs.

3.4 Transformer Models in Dynamic Graphs

Recent advancements in graph learning, particularly the introduction of transformer models, have significantly improved anomaly detection in dynamic graphs. Transformers, originally designed for natural language processing tasks, have demonstrated remarkable capabilities in capturing long-range dependencies and learning complex representations. TADDY (Transformer-based Anomaly Detection in Dynamic Graphs) (Liu, et al., AUGUST 2015) , extends transformer models to the domain of dynamic graphs, addressing the challenges that previous methods encountered.

TADDY introduces a novel approach by leveraging a transformer-based model capable of simultaneously capturing spatial and temporal information in an end-to-end manner. Unlike prior models that handled spatial and temporal data separately, TADDY integrates these aspects into a single framework. Additionally, it proposes a comprehensive node encoding strategy that distills both global spatial and local temporal information, which significantly improves the detection of anomalous edges in dynamic graph streams.

3.5 Key Contributions of Transformer-Based Models

The key innovation of transformer-based models, such as TADDY, lies in their ability to capture coupled spatial-temporal information. By utilizing multi-head attention mechanisms, these models can learn intricate relationships across both space and time without the need for separate spatial and temporal modules. This results in a more unified and efficient learning process that leads to improved anomaly detection performance. Furthermore, transformer models benefit from scalability and the ability to generalize across several types of dynamic graphs, making them a appropriate solution for real-world applications.

3.6 Conclusion

The progression from shallow learning methods to transformer-based models in anomaly detection for dynamic graphs illustrates the evolution of the field toward more sophisticated, scalable, and accurate techniques. While early approaches laid the groundwork for graph-based anomaly detection, modern transformer models like TADDY offer a significant performance leap, particularly in their ability to handle the spatial-temporal coupling inherent in dynamic graphs.

4. Background

4.1 Graph Theory and Dynamic Graphs

Graph theory provides the foundational framework for analyzing relationships and structures within datasets (Liu, et al., AUGUST 2015). In academic citation networks, graphs represent papers as nodes and citations as directed edges. Dynamic graphs, also known as temporal graphs, are essential for understanding how these relationships evolve over time (Holme & Saramaki, December 2011). They enable us to analyze how citation links between academic papers develop and potentially deviate from expected patterns as new publications emerge.

4.2 Subgraphs and Graph Diffusion

Understanding subgraphs—smaller subsets of a graph's nodes and edges—is essential for focusing on local citation patterns (Liu, et al., AUGUST 2015) that might not be evident in the larger network. This focus allows for the detection of anomalies within specific communities or clusters of papers. Additionally, graph diffusion processes, such as Personalized PageRank, help disseminate information across the graph, depicting the influence and connectivity of nodes over time. These methods assist in identifying influential papers and detecting unusual citation behaviors.

4.3 Node and Edge Embeddings

To effectively detect anomalies in citation networks, it is crucial to represent nodes (papers) and edges (citations) in a way that captures their essential characteristics. Embedding techniques transform nodes and edges into a lower-dimensional vector space, facilitating the analysis of their properties and relationships. These embeddings serve as foundational inputs for anomaly detection algorithms, enabling the model to process complex graph structures more efficiently.

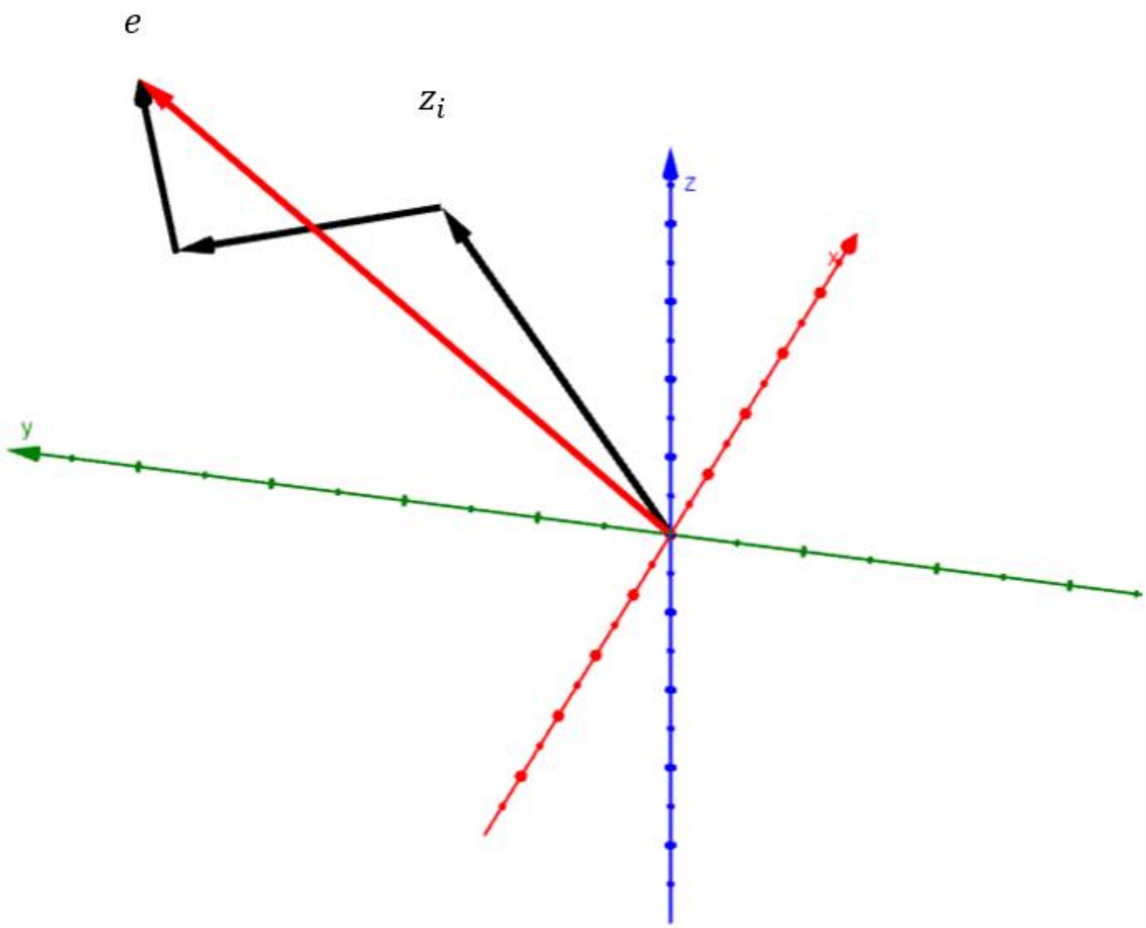

Figure 4-1 ***Figure 4-2:*** *After the pooling stage, all the encoded vectors representing the nodes in the substructure of the examined edge are summed and normalized into a single vector, which represents that edge.*

4.4 Discriminative Knowledge

Discriminative knowledge involves insights that distinguish between distinct categories within datasets by highlighting features uniquely associated with specific classes. In our research, we leverage discriminative knowledge to identify citation patterns that deviate from normative academic practices. This approach aids in the detection of anomalous citations by focusing on characteristics that are indicative of irregular citation behavior.

4.5 Spatial and Temporal Information

Spatial information relates to how nodes are positioned relative to each other within the graph and how these positions affect their interactions. Temporal information captures changes and developments over time, revealing trends, patterns, and anomalies in dynamic graphs. Analyzing spatial and temporal patterns in citation practices helps detect shifts that may indicate emerging trends or irregular citation activities. This combined analysis plays a key role to understand the context in which citations occur and evolve.

4.6 Citation Network

Citation networks are a type of directed graph where nodes represent academic papers, and directed edges signify citations from one paper to another. By examining the structure and connectivity of these networks, researchers can infer the influence of specific works and detect trends and patterns in academic disciplines.

citation networks are analyzed to uncover potentially anomalous citation behaviors. we identify unusual patterns of citations that may not align with typical scholarly practices.

4.7 Neural Networks and Transformer Models

Artificial neural networks (ANNs) are models inspired by the human brain's architecture (Y. Liu, 2023) enabling computers to learn from data and make informed decisions. **Transformer models**, a class of deep learning architectures, have revolutionized the processing of sequential and graph-structured data (Vaswani, Shazeer, Parmar, & al., 2017) Graph Transformer Models apply the principles of transformers to graph data, using attention mechanisms to dynamically weigh the relationships between nodes. This approach allows for the learning of complex patterns and dependencies within citation networks, making them particularly effective for anomaly detection tasks.

4.8 Attention Mechanisms

Attention mechanisms are components of neural network architectures that allow models to focus on relevant parts of the input data dynamically (Vaswani, Shazeer, Parmar, & al., 2017). In Graph Transformer Models, attention mechanisms enable the model to assign varying levels of importance to different nodes and edges, improving the ability to capture contextual relationships and detect anomalies. By considering the significance of each citation in relation to others, the model can more accurately identify irregular patterns.

4.9 TADDY: Transformer-based Anomaly Detection for Dynamic Graphs

TADDY (Transformer-based Anomaly Detection for Dynamic Graphs) is a framework designed to detect anomalies in dynamic graphs. By leveraging Graph Transformer Models and attention mechanisms, TADDY can manage the evolving nature of citation networks effectively. It integrates spatial and temporal information, node and edge embeddings, and discriminative knowledge to generate an anomaly score for each edge (citation) in the graph. This score reflects the likelihood of it being anomalous based on its deviation from learned patterns of normal citation behavior.

5. Mathematical Background

5.1 Graph Theory

5.1.1 Dynamic Graphs

A dynamic graph, is represented as a series of snapshots (Aggarwal & Subbian, April 2014) $G = \{G_1, G_2, \dots, G_T\}$, where each snapshot $G_t = (V_t, E_t)$ contains a set of nodes V_t and edges E_t at time t. The changes in the graph structure over time capture the dynamic interactions and relationships between entities.

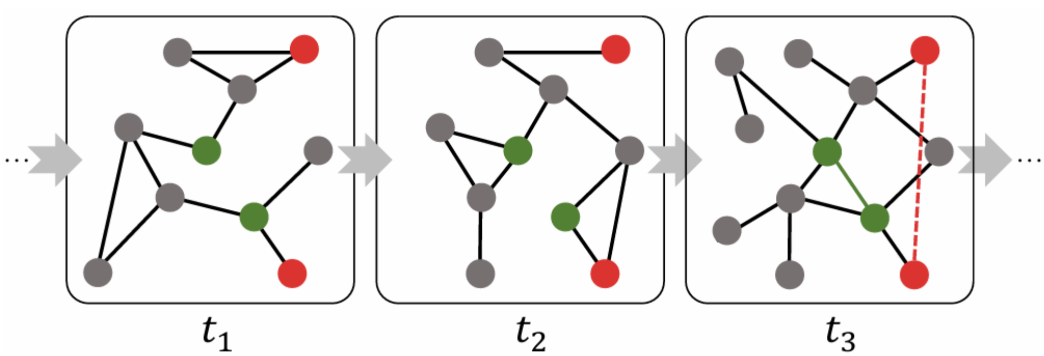

Figure 5-1: *Dynamic Anomaly Detection Example - A toy example to illustrate how the coupled information affects the detection of edges' legality. The three graphs are a fragment from a dynamic graph stream at a sequential timeline t_1 t_2 t_3 . The solid green line represents a normal edge at t_3, while the red dash line indicates an anomalous edge. We highlight the corresponding nodes in the previous timestamps with colors. ("Anomaly Detection in Dynamic Graphs via Transformer")*

5.1.2 Subgraph and Substructure

A subgraph is a subset of a graph's nodes and edges that forms a graph (Diestel, 2017). Substructures in anomaly detection refer to local subgraphs centered around target edges or nodes. These substructures help detect anomalies by focusing on local patterns that may not be apparent in the global graph structure.

This approach allows for efficient computation and localized analysis, which is crucial in large-scale dynamic graphs.

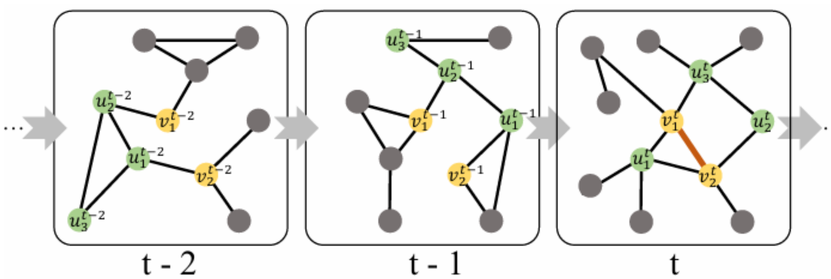

Figure 5-2: *Edge-based Substructure Sampling In edge-based substructure sampling, the target nodes (in yellow) and contextual nodes (in green) from multiple timestamps are sampled to construct the substructure node set, where neighboring node number k and time window size are both set to be 3*

.

5.1.3 Graph Diffusion

Graph diffusion processes are used to capture the global structure of a graph by spreading information from each node to its neighbors (Liu, et al., AUGUST 2015). Two common types of graph diffusion are Personalized PageRank (PPR) and the heat kernel.

5.1.4 Personalized PageRank (PPR)

Personalized PageRank (PPR) computes a ranking score for each node based on its connectivity to a specific node (Liu, et al., AUGUST 2015) (Ma, Guan, & Zhao, 1999). The diffusion matrix for PPR is defined as:

$$S_{PPR} = \alpha \left(I_n - (1 - \alpha) D^{-\frac{1}{2}} A\, D^{-\frac{1}{2}} \right)^{-1}$$

where α is the teleport probability, I_n is the identity matrix, D is the diagonal degree matrix, and A is the adjacency matrix.

This formulation allows for capturing both local and global information in the graph by balancing between direct connections and global reachability.

5.1.5 Heat Kernel

The heat kernel diffusion captures the spread of heat (information) over time in a graph. The diffusion matrix for the heat kernel is defined as:

$$S_{heat} = exp\ exp\ (\beta\ A\ D^{-1} - \ \beta)$$

where β is the diffusion time, and exp represents the matrix exponential.

The heat kernel is effective in modeling processes that involve gradual propagation, such as heat distribution or information flow in networks.

5.2 Node Encoding

Node encoding is essential for capturing the structural and temporal information of nodes in a dynamic graph (Hamilton, Ying, & Leskovec, 2017). Various encoding strategies are used to create informative representations of nodes.

5.2.1 Diffusion-based Spatial Encoding

Diffusion-based spatial encoding uses the diffusion matrix to capture the global structure of nodes in the graph. The rank-based encoding is computed using the following formula:

$$x_{diff(v_{ij})} = \ linear\left(rank\left(s_{ie_{tgt}}\,[idx(v_{ij})]\right)\right) \in \ \mathbb{R}^{d_{enc}}$$

where idx is the index function, rank is the ranking function, and linear is a learnable linear mapping.

This encoding captures the influence and importance of a node within the graph.

5.2.2 Distance-based Spatial Encoding

Distance-based spatial encoding represents the local structure around the target edge by computing the minimum distance to the target nodes. The encoding is defined as:

$$x_{dist(v_{ij})} = \ linear\left(min\left(dist\left(v_{ij}, v_{i1}\right), dist\left(v_{ij}, v_{i2}\right)\right)\right) \in \ \mathbb{R}^{d_{enc}}$$

where $dist$ is the distance function, min is the minimum function, and linear is a learnable linear mapping.

This encoding effectively captures the local neighborhood information, which is important for identifying local anomalies.

5.2.3 Relative Temporal Encoding

Relative temporal encoding captures the temporal information of nodes by computing the difference between the occurrence time of the target edge and the current timestamp. The encoding is given by:

$$x_{temp(v_{ij})} = linear(|t - i|) \in \mathbb{R}^{d_{enc}}$$

where $|t - i|$ is the absolute difference in time, and linear is a learnable linear mapping.

This encoding is crucial for capturing temporal dynamics and changes over time, enabling the detection of temporal anomalies.

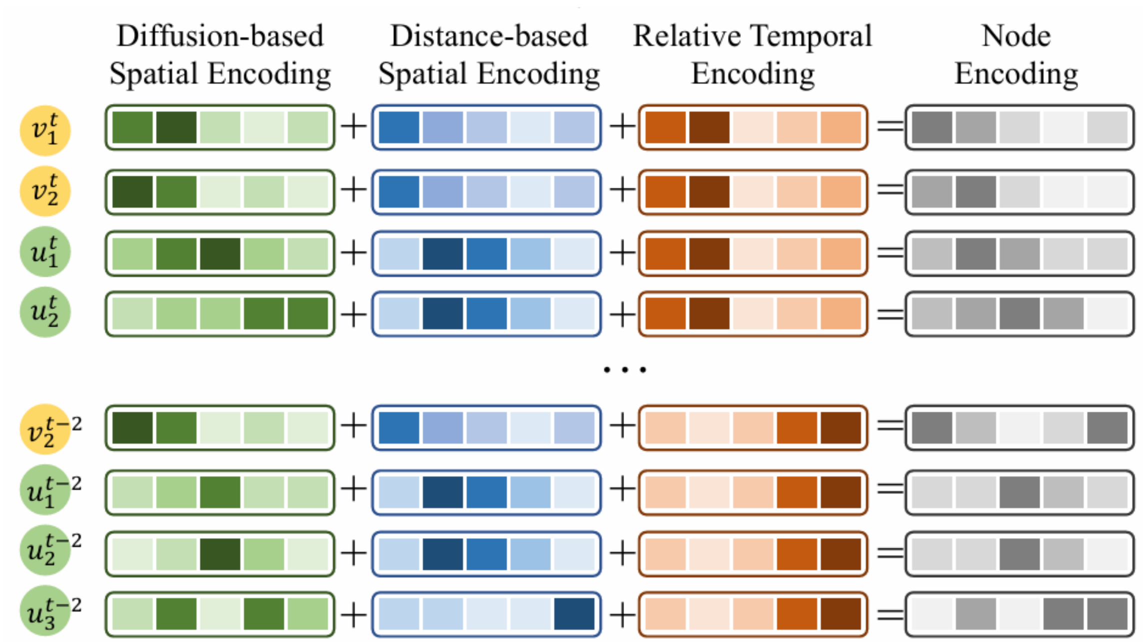

Figure 5-3: *Spatial-temporal Node Encoding - Three types of encoding are computed for each node and further fused into the node encoding*

5.3 Transformer Model

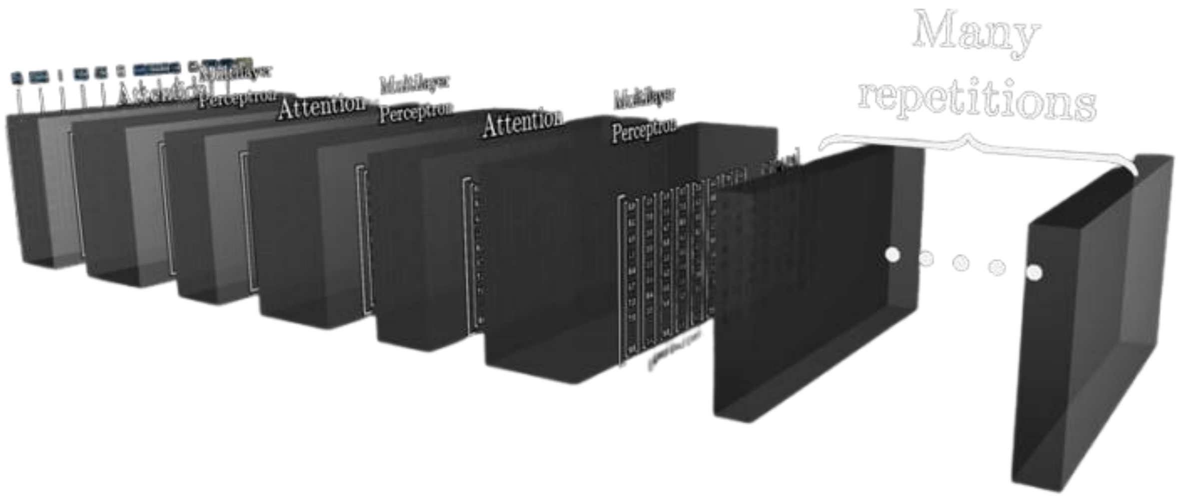

Figure 5-4: *Transformer model architecture depiction*

Transformers are a powerful class of neural networks originally designed for sequence data (Vaswani, Shazeer, Parmar, & al., 2017), but they have been adapted for graph data due to their ability to capture complex dependencies.

5.3.1 Attention Mechanism

The attention mechanism allows the model to focus on various parts of the input sequence (Vaswani, Shazeer, Parmar, & al., 2017), which is important for capturing relationships in graph data. The scaled dot-product attention is defined as:

$$Attention(Q,K,V) = softmax\left(Q\frac{K^T}{\sqrt{d_k}}\right)V$$

where **Q**ueries, **K**eys and **V**alues are input matrices, and d_k is the dimension of the keys.

This mechanism enables the model to weigh the importance of different nodes and edges dynamically.

5.3.2 Multi-head Attention

Multi-head attention extends the attention mechanism by applying it multiple times in parallel (Vaswani, Shazeer, Parmar, & al., 2017), with each attention head focusing on various parts of the input. It is defined as:

$$MultiHead(Q, K, V) = Concat(head_1, \ldots, head_h)W_O$$

where each head is an instance of the attention mechanism, and W_O is a learnable parameter matrix.

This allows the model to capture multiple aspects of the data simultaneously, enhancing its ability to learn complex patterns.

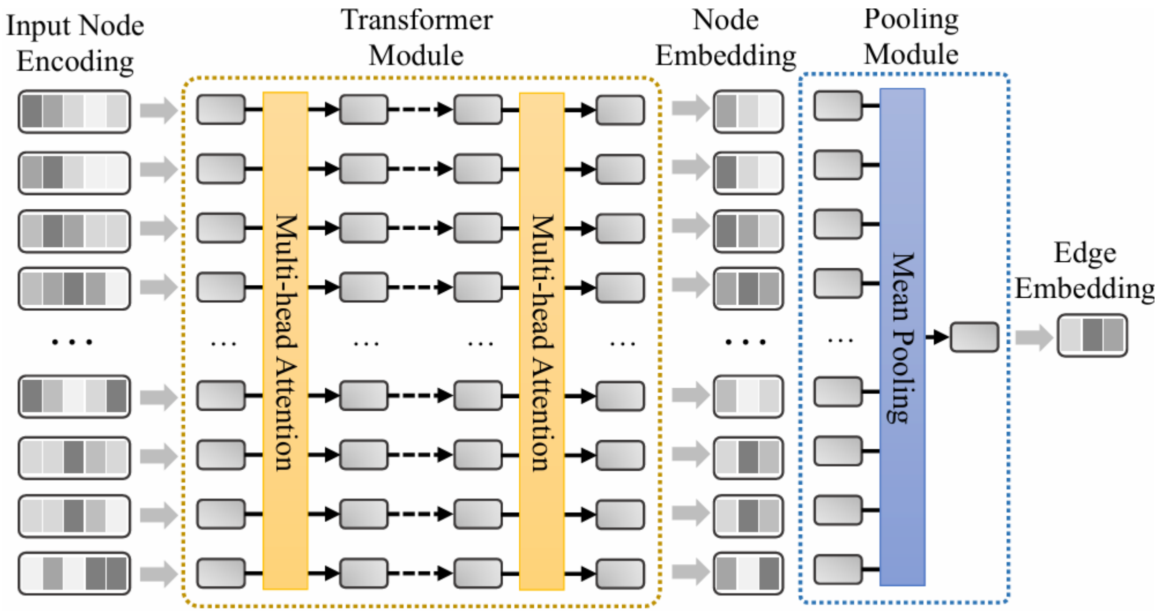

Figure 5-5: *Dynamic Graph Transformer Three types of encoding are computed for each node and further fused into the node encoding*

5.4 Node and Edge Embedding

Embedding techniques are used to represent nodes and edges in a lower-dimensional space (Hamilton, Ying, & Leskovec, 2017), capturing their essential characteristics.

5.4.1 Node Embedding

Node embeddings are computed using transformer layers, which aggregate information from the node's neighbors (Liu, et al., AUGUST 2015) (Grover & Leskovec, November 2016). The dimension of the embeddings is controlled by a parameter d_{emb}.

These embeddings capture the structural and relational properties of nodes in the graph.

5.4.2 Edge Embedding

Edge embeddings are obtained by pooling the node embeddings within a substructure . The mean pooling operation is used to aggregate the embeddings:

$$z\left(e_{t_{tgt}}\right) = pooling(Z) = \left(\frac{1}{n_s}\right)\sum_{k=1}^{n_s} Z_k$$

where Z is the matrix of node embeddings, and n_s is the number of nodes in the substructure.

This aggregation captures the overall relationship and interaction between nodes, which is crucial for edge-level anomaly detection.

5.5 Anomaly Detection

Anomaly detection involves identifying data points that deviate significantly from the expected patterns. In this context, the focus is on detecting anomalous edges in dynamic graphs.

5.5.1 Anomaly Score Calculation

Anomaly scores are computed using a fully connected neural network with a Sigmoid activation function. This neural network serves as the discriminative anomaly detector, which distinguishes between normal and anomalous edges based on their embeddings. The anomaly score for an edge is given by:

$$f(e) = Sigmoid(z(e)\, W_s + b_s)$$

where $z(e)$ is the edge embedding, W_s is a weight matrix, and b_s is a bias term.

This score indicates the likelihood of an edge being anomalous based on its embedding.

5.5.2 Binary Cross-Entropy Loss

The discriminative anomaly detector is trained using a binary cross-entropy loss function, which measures the discrepancy between the predicted and actual labels. The loss function is defined as:

$$L = -\sum_{i=1}^{m_t}\left[log\left(1 - f\left(e_{pos,i}\right)\right) + log\left(f\left(e_{neg,i}\right)\right)\right]$$

Where $\boldsymbol{e_{pos,i}}$ and $\boldsymbol{e_{neg,i}}$ are the positive and negative edges, respectively.

This loss function guides the model to distinguish between normal and anomalous edges.

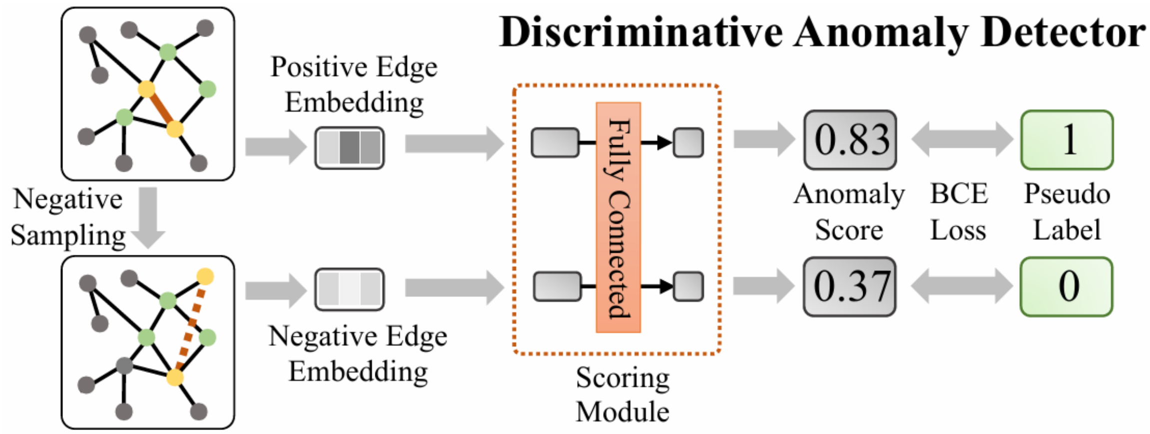

Figure 5-6: *discriminative anomaly detector - the negative edges are acquired by negative sampling. "The scoring module computes the anomaly scores for positive and negative edges." ("Anomaly Detection in Dynamic Graphs via Transformer - Academia.edu") The whole framework is trained with a binary cross-entropy loss in an end-to-end manner.*

5.6 Negative Sampling

Negative sampling is used to generate negative samples for training the anomaly detector. The strategy ensures that negative samples do not appear in the positive edge set, providing a strong training signal.

This technique is crucial for handling imbalanced datasets and improving the model's ability to discriminate between normal and anomalous edges.

6. Expected Achievements

Our project aims to achieve effective detection of citation anomalies, ensuring temporal adaptability, scalability, and usability in academic citation networks. We will assess success through key evaluation criteria: accuracy in detecting anomalous citations, the model's ability to adapt and update anomaly scores as the citation network evolves, computational efficiency in processing large datasets, and the clarity of visualizations. To measure these, we will use simulated citation networks for controlled testing, user studies for evaluating visualization effectiveness, and scalability tests. Success will be indicated by high detection rates, accurate temporal tracking, and efficient performance, ensuring the model meets practical use standards and contributes to improving academic citation integrity.

7. Research Process

7.1 Model Overview

We provide an overview of the application of the TADDY algorithm in our research, focusing on detecting and analyzing citation anomalies within dynamic citation networks. We extend the algorithm's capability to capture temporal dynamics more comprehensively by scoring anomaly nodes (papers) over time.

7.2 Dataset

We utilize the CORA dataset (Relational Dataset Repository, n.d.), recognized for its rich citation network. The dataset's complex network of citations provides an appropriate basis for evaluating our enhanced anomaly detection methods.

7.3 Model Architecture

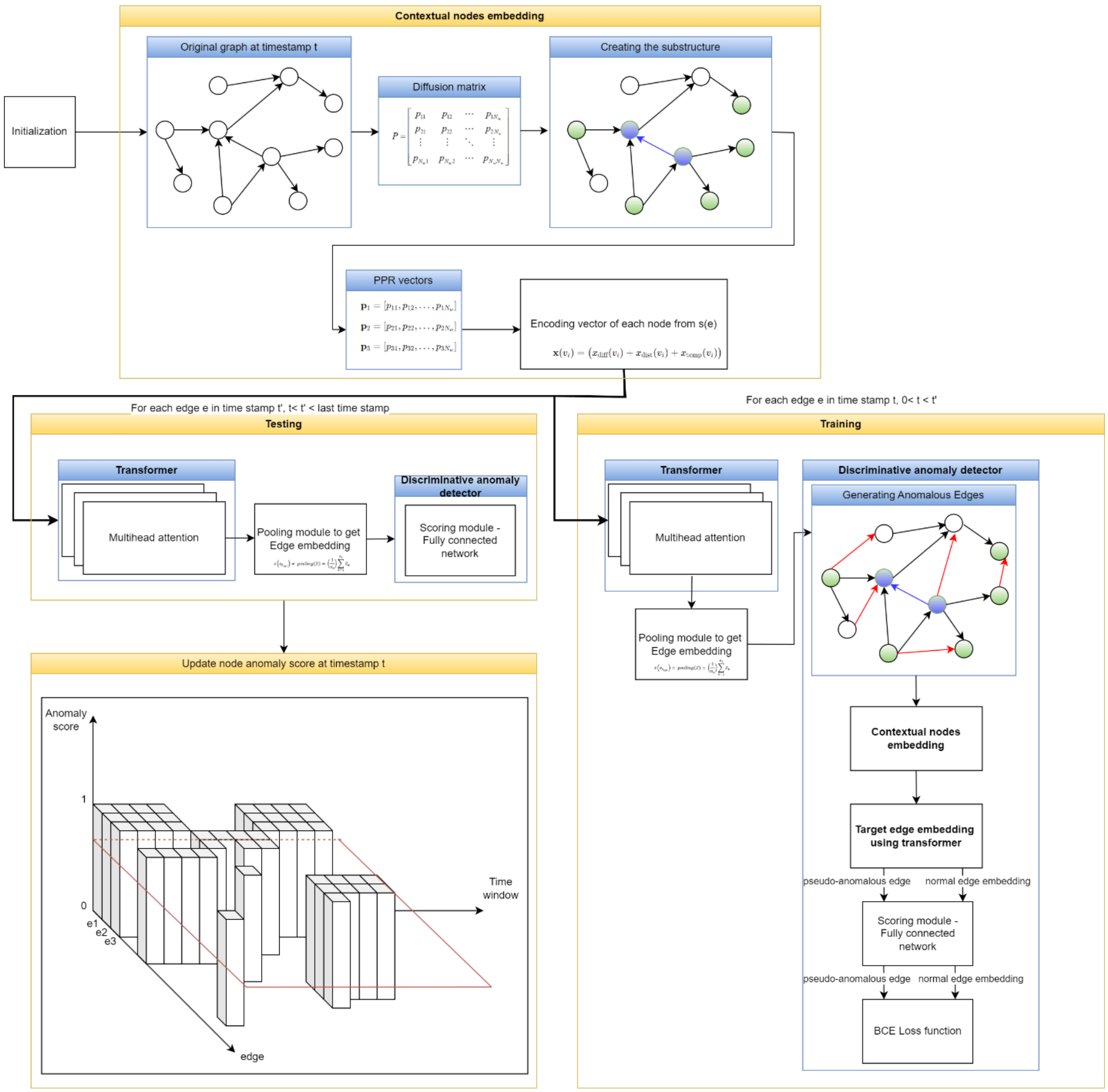

Figure 7-1: *Model Architecture*

7.3.1 Initialization

During the initialization phase of the TADDY model, we begin by importing and structuring the dynamic graph data. This involves representing each node with relevant feature vectors and setting up the relationships between nodes (edges) that can change over time. A dedicated data structure is then established to handle and track changes in the graph, such as the addition, removal, or modification of edges.

The dataset is split into training and testing sets, ensuring that edges do not overlap between the two sets. For each set, the model constructs subgraphs, or local views for each node, incorporating its immediate neighbors to capture the local context.

7.3.2 Embedding

In this stage, every node of the substructure of a specific edge from the graph is encoded while considering the three previously mentioned aspects: diffusion-based spatial encoding, relative temporal encoding and distance-based spatial Encoding, based on their corresponding diffusion values from the diffusion matrix.

7.3.3 Training Stage

Since no labeled anomalies are present, after the embedding, negative sampling is performed to generate pseudo-anomalous edges by sampling node pairs that do not exist in the graph. For each edge (both positive and negative), the local substructure is sampled, and a spatial-temporal node encoding is applied to capture the graph's relationships over time. These encoded representations are passed through the dynamic graph transformer to produce edge embeddings. These embeddings are then fed into the Discriminative Anomaly Detector, a fully connected neural network, which computes anomaly scores for each edge. The model is trained to minimize a loss function by distinguishing between positive (normal) edges and negative (pseudo-anomalous) edges. Over iterations, the model learns to detect anomalies by optimizing its parameters using backpropagation and gradient descent.

7.3.4 Testing Stage

For timestamps that come after the training stage timestamps, each edge in the tested timestamp is substructure sampled and spatial-temporal encoding is applied, like the training stage. The encoded information is then passed through the dynamic graph transformer to generate edge embedding. This edge embedding is used as an input to the Discriminative Anomaly Detector, which computes an anomaly score for each edge using the trained neural network. The score, ranging between 0 and 1, indicates the likelihood of an edge being anomalous, with scores closer to 1 suggesting an anomaly.

7.3.5 Model Evaluation

After getting the tested anomaly edges of the graph at a specific timestamp we extend the conventional application of the TADDY algorithm by developing a three-dimensional temporal analysis method. This enhancement improves our ability to analyze the progression of citation anomalies over time within academic papers.

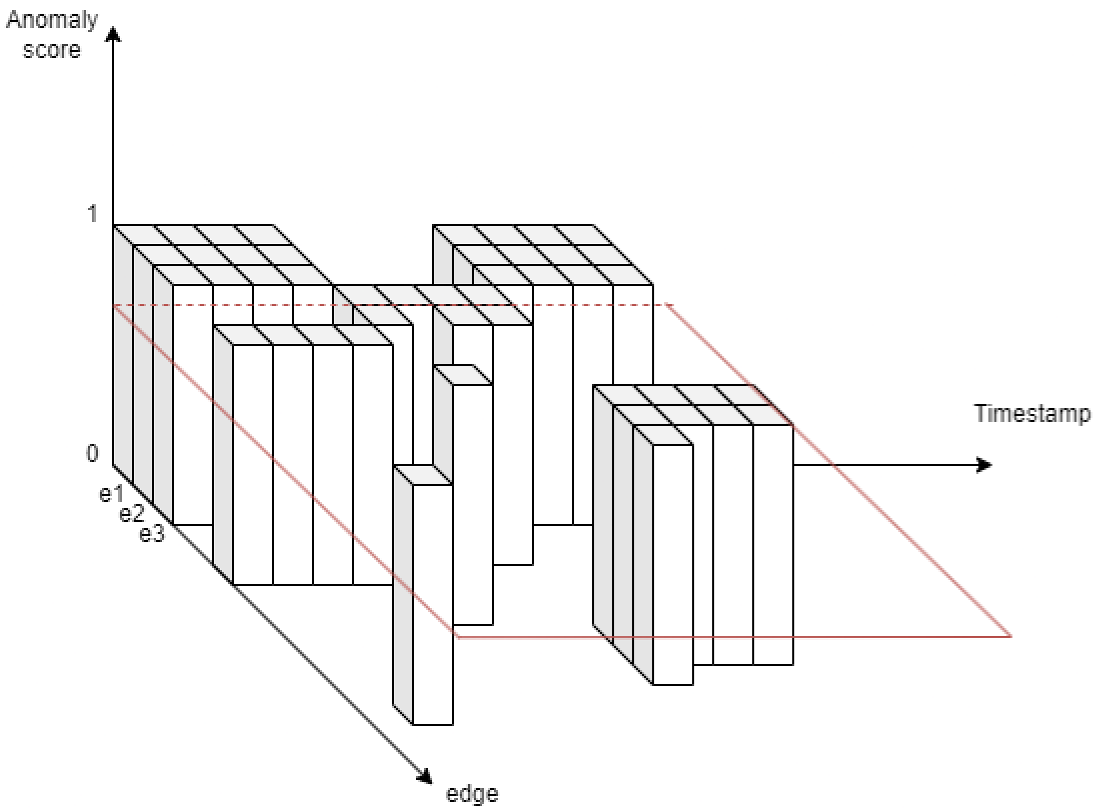

Figure 7-2: *This is an example of our score function for a specific paper.*
The coordinates of this function are: (Time window, edge from the graph, anomaly score).
Where the function is: $f: time\ window\ t\ \times\ edge\ e \rightarrow \{0,1\} \quad F(t, e) = z$
The red plane represents the threshold height that has been predefined.

To deepen our analysis, we adapt the TADDY algorithm to output a vector of anomalous edges not just at a single timestamp but across multiple time windows. This adaptation allows us to construct a dynamic 3D function for each paper, where the axes represent time windows, edges, and corresponding anomaly scores.

For each evaluated time window, the output vector from TADDY updates this 3D function with the anomaly scores associated with each targeted paper. Each edge $\boldsymbol{e}_i$ at its respective time window $\boldsymbol{t}_i$ contributes to the function's value, marking anomalies distinctly. This iterative update provides a strong framework for observing the evolution of citations.

By intersecting each function with a horizontal plane at a set threshold height, we calculate the cumulative sum of the function's values above this threshold. This approach not only captures but also quantifies the shifts in citation behavior over time, highlighting specific periods where anomalous activities are most pronounced.

8. Expected challenges:

8.1 Computation and Storage of Dynamic Data:

The three-dimensional analysis of dynamic graphs, which includes the temporal dimension, significantly increases computational complexity. Tracking changes over time requires constant updates to the graph with each new paper added, leading to extensive computation and storage requirements (Aggarwal & Subbian, April 2014). Managing this dynamic data efficiently is a significant challenge, especially as the size of the citation network grows.

8.2 Threshold Adjustment for Anomaly Detection:

Defining an appropriate anomaly score threshold is crucial for evaluating whether certain citations are anomalous. Setting the correct threshold is challenging because a high threshold might miss genuine anomalies, while a low threshold could result in numerous false positives. The threshold must account for the complexity of the citation network and its temporal changes to balance sensitivity and specificity effectively.

8.3 Interpreting Model Results:

Presenting the model’s results in an understandable visual format and interpreting them can be challenging. Analyzing anomalies at different points in time and displaying them in a three-dimensional manner requires a presentation method that allows researchers to easily comprehend the data and identify anomalies. Developing intuitive visualization techniques is essential for effective analysis and communication of the results.

8.4 Processing Time:

Because the analysis involves tracking changes over time, the model requires substantial processing time to accurately compute anomalies in each time window. This can be a significant issue when working with large-scale academic datasets, potentially impacting the feasibility of the approach for real-time or near-real-time analysis.

8.5 Adapting a Dynamic Citation Dataset to the Requirements of the TADDY Algorithm

Adapting a dataset of academic papers for use with the TADDY algorithm is ensuring it accurately captures dynamic citation relationships over time. This requires proper preprocessing, including time-stamped citation data, and aligning the dataset with the algorithm's input format. Incomplete or improperly structured data can lead to poor anomaly detection and misinterpretation of trends, undermining the effectiveness of the analysis.

9. Evaluation/Verification Plan

Test Case ID	Description	Function Tested	Expected Outcome
1.1	Verify dataset preprocessing to include time-stamped citation information	preprocess_dataset	Dataset is transformed into time-stamped citations compatible with TADDY input
1.2	Test embedding generation for nodes and edges (citations)	generate_embeddings	Nodes and edges are successfully transformed into lower-dimensional embeddings
1.3	Validate model's ability to adjust to evolving citation patterns	update_model_for_changes	The model adjusts its anomaly detection results as citation patterns change over time
2.1	Validate anomaly score calculation for a specific citation over time	detect_anomalies	Anomaly score correctly reflects citation behavior across time windows
2.2	Test anomaly score calculation for shorter time windows	calc_anomaly_score	Anomaly score is calculated correctly for shorter time windows
2.3	Test anomaly score calculation for longer time windows	calc_anomaly_score	Anomaly score is calculated correctly for longer time windows

10. References

Aggarwal, C. C., & Subbian, K. (April 2014). Evolutionary Network Analysis: A Survey. *ACM Computing Surveys*.

Aggarwal, C. C., Zhao, Y., & Philip, S. Y. (2011). Outlier detection in graph streams. *ICDE - IEEE*, 399–409.

BAI, X., ZHANG, F., NI, J., SHI, L., & LEE, I. (Jan 2020). Measure the Impact of Institution and. *10.1109/ACCESS.2020.2968459*.

Chakraborty, J., Chakraborty, T., Nandi, S., Dutta, H. S., & Pradhan, D. (JUL 2018). On Good and Bad Intentions behind Anomalous Citation. *arXiv preprint arXiv:1807.10804*.

Diestel, R. (2017). *Graph Theory, 5th edition.* Hamburg, Germany: Springer Nature.

Goodfellow. (Oct 2016). *Deep Learning.*

Grover, A., & Leskovec, J. (November 2016). node2vec: Scalable Feature Learning for Networks. *In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*

Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Inductive Representation Learning on Large Graphs. *In Advances in Neural Information Processing Systems (pp. 1024–1034).*

Holme, P., & Saramaki, J. (December 2011). Temporal Networks. *arXiv:1108.1780v2 [nlin.AO].*

Liu, Y., Pan, S., Wang, Y. G., Xiong, F., Wang, L., Chen, Q., & Lee, V. C. (AUGUST 2015). Anomaly Detection in Dynamic Graphs via Transformer. *JOURNAL OF LATEX CLASS FILES, VOL. 14, NO. 8*.

Ma, N., Guan, J., & Zhao, Y. (1999). The PageRank Citation Ranking: Bringing Order to the Web. *Stanford InfoLab*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). *Distributed Representations of Words and Phrases.*

Ranshous, S., Harenberg, S., Sharma, K., & Samatova, N. F. (2016). A scalable approach for outlier detection in edge streams using sketch-based approximations. *SDM*, 189–197.

Relational Dataset Repository. (n.d.). Retrieved from https://relational-data.org/dataset/CORA

Vaswani, A., Shazeer, N., Parmar, N., & al., e. (2017). Attention is all you need. *In Advances in Neural Information Processing Systems (pp. 5998–6008).*

Y. Liu, S. P. (2023). Brain-inspired learning in artificial neural networks. *arXiv*.

Yu, W., Cheng, W., Aggarwal, C. C., Zhang, K., Chen, H., & Wang, W. (2018). NetWalk: A flexible deep embedding approach for anomaly detection in dynamic networks. *SIGKDD*, 2672–2681.

Zheng, L., Li, Z., Li, J., Li, Z., & Gao, J. (2019). AddGraph: Anomaly detection in dynamic graph using attention-based temporal GCN. *IJCAI*, 4419–4425.