



Software Engineering Department Braude College

Capstone Project Phase A

24-2-R-14

Anomaly Detection in Texts Using Graph-Based Methods: Leveraging Graph Fourier Transform and Laplacian

Students:

Alexander Kovtonyuk

Alexkobt3@gmail.com, 316436419

Supervisor:

Prof. Zeev Volkovich

GitHub:

[GitHub Repository](#)

Table of Contents

1 Abstract.	3
2 Introduction.	3
2 Literature Review	4
3 Background	5
3.1 Spectral Graph Theory	5
3.2 BERT Feature Extraction and Embedding	6
3.3 Graph Laplacian	7
3.4 Graph Fourier Transform	8
3.5 Semantic and Syntactic Relationships.	9
4 Expected Achievements	9
5 Engineering Process	9
5.1 Data Preprocessing	9
5.2 Graph Construcion	10
5.3 Computing The Graph Laplacian	10
5.4 Graph Fourier Transform (GFT)	11
5.5 Analyzing High-Frequency Components	11
5.6 Evaluating Anomalies	11
5.7 System Class Diagram	12
5.8 Hyper Parameters	13
6 Evaluation/Verification Plan	13
7 Implemntation Plan	15
8 Bibliography	15

Abstract

Detecting anomalies in textual data, such as articles, is challenging due to the complexity and variability of language. Traditional methods, like statistical models and machine learning techniques, often fall short due to their inability to capture complex semantic and syntactic relationships within a text effectively. This project proposes a graph-based solution based on the Graph Fourier Transform (GFT) for anomaly detection in textual material. Via a graph representation of text data employing the Laplacian operator based on GFT, we convert the graph signal, depicted in the text, into the frequency domain. Analyzing high-frequency components of the signal in the spectral domain allows us to recognize suspect anomalies. This method promises to enhance the robustness and accuracy of anomaly detection while maintaining computational efficiency.

Keywords

Anomaly Detection · Graph Fourier Transform (GFT) · Laplacian Matrix · Spectral Analysis · Text Processing

Introduction

Anomaly detection is a data mining technique focused on identifying rare or unusual occurrences within datasets. These anomalies, or outliers, deviate significantly from the norm and can have crucial implications across various domains, such as security, finance, and healthcare. Unlike traditional outlier detection methods that analyze data as independent points in multi-dimensional space, anomaly detection in graphs leverages the inherent relationships and dependencies between data objects [10].

However, anomaly detection in textual data presents unique challenges due to the complexity and variability of natural language. Context plays a critical role in text data, where the meaning and relevance of a term can change based on its surrounding words, sentences, or even paragraphs. This context-dependent nature makes it difficult for traditionally used methods to distinguish between normal variations in language and true anomalies [10].

Existing solutions for anomaly detection in textual data include statistical models, machine learning techniques, and deep learning approaches. While these methods have shown some success, they often struggle with the high dimensionality and intricate structure of text data [3]. Statistical models may oversimplify the relationships within the text, leading to

inaccurate anomaly detection. Additionally, deep learning models, though powerful, can be computationally expensive and require large labeled datasets for training, which limits their practicality for many applications [4, 14].

This project proposes a graph-based approach for anomaly detection in articles by utilizing the Graph Fourier Transform (GFT) and the Laplacian matrix inspired by "Graph Laplacian for image anomaly detection" article [2]. By representing text data as a graph where nodes correspond to text units (e.g., words, sentences or paragraphs) and edges will represent semantic and syntactic relationships. We construct the Laplacian matrix using the constructed graph's adjacency matrix and degree matrix. The eigen decomposition of the Laplacian matrix provides the basis for the GFT. By transforming the graph signal into the frequency domain, we will analyze high-frequency components to identify anomalies.

2. Literature Review

Anomaly detection in text data has been approached using various methods, including statistical models, machine learning, and deep learning techniques. These methods often face challenges in capturing the nuanced relationships within text data, which can result in suboptimal performance and high computational [3, 5, 7].

Traditional statistical models, such as those based on Gaussian distributions, often oversimplify the relationships within text data. They tend to assume a specific distribution of data points which may not accurately reflect the complexity of textual data. Which can lead to inaccurate anomaly detection. Machine learning approaches, such as clustering and classification algorithms, have been widely used for anomaly detection in textual data. These methods include Support Vector Machines (SVMs) [9], k-Nearest Neighbors (k-NN), and Random Forests. However, they require careful feature engineering and may not effectively capture the underlying semantics of text [3, 5].

Deep learning models, particularly those based on neural networks, have shown significant promise in anomaly detection. Techniques such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) are used to learn complex patterns within text data [6]. Despite their power, these models are computationally expensive and often require large labeled datasets for training, which can limit their practicality for many applications [6, 7].

Graph theory and spectral analysis offers a unique direction for anomaly detection in complex data structures. Graph-based techniques leverage the inherent relationships between data points, represented as nodes and edges in a graph, to detect anomalies. By utilizing the Graph Fourier Transform (GFT), these methods can analyze the spectral properties of graph signals to identify irregularities [2].

3. Background

3.1 Spectral Graph Theory:

Spectral graph theory involves analyzing the properties of graphs through the eigenvalues and eigenvectors of matrices such as the adjacency matrix or the Laplacian matrix. This mathematical framework helps understand the underlying structure of graphs and detect anomalies by examining their spectral properties. The adjacency matrix captures the connectivity between nodes, while the Laplacian matrix, defined as the difference between the degree matrix and the adjacency matrix, is particularly useful for identifying clusters and understanding diffusion processes within the graph.

By studying the eigenvalues (spectrum) and eigenvectors (modes) of these matrices, one can gain insights into various graph properties, such as connectivity, clustering, and community structure. Anomalies often manifest as deviations from expected spectral patterns, such as unusually dense subgraphs or nodes with unexpected connectivity. Techniques like spectral clustering and graph signal processing are employed to detect these irregularities. Spectral graph theory thus provides a powerful toolkit for analyzing complex networks, allowing for the identification of anomalies and the exploration of the intrinsic geometry of graphs.

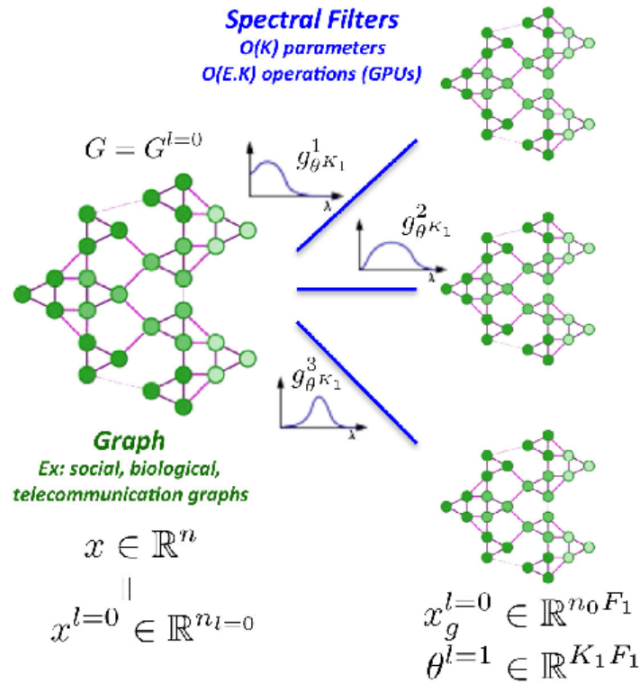


Figure 1: Visual representation of Spectral Graph Filtering by eigenvalues and eigenvectors. [15]

3.2 BERT Feature Extraction and Embedding:

BERT (Bidirectional Encoder Representations from Transformers) is a powerful pre-trained language model widely used for various natural language processing (NLP) tasks. One of the key steps in utilizing BERT for textual analysis is tokenization. Tokenization is the process of splitting text into smaller units, such as words or subwords, which can be effectively processed by the model. BERT uses a specific tokenization technique called WordPiece tokenization, which breaks down words into subword units. This method allows BERT to handle rare words and out-of-vocabulary terms more efficiently by representing them as a combination of known subwords.

Once the text is tokenized, each token is converted into a numerical representation, known as an embedding. BERT generates embeddings that capture the contextual meaning of each token in the sentence. These embeddings are 768-dimensional vectors (for BERT-base) that encapsulate rich semantic information. Feature extraction involves passing the tokenized text through the BERT model to obtain these embeddings. The tokenized text is then fed into the BERT model, which outputs embeddings for each token.

In this project we will be using BERT's model abilities of tokenization and feature extraction to create embeddings for our text units, which is a crucial step in building our graph.

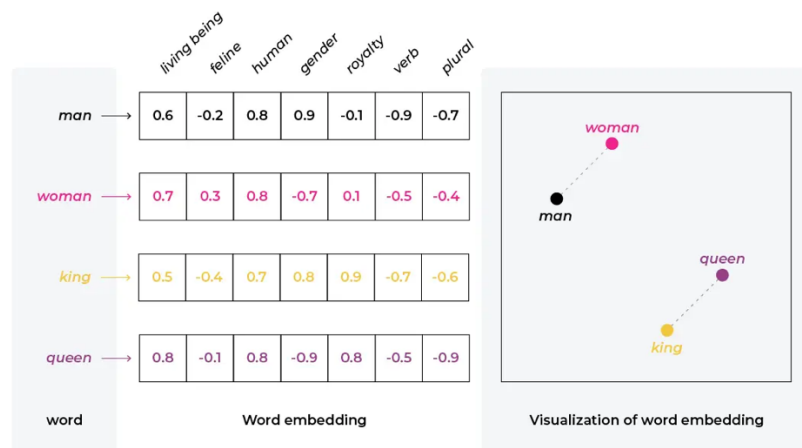


Figure 2: Visual representation of the embedding and feature extraction process. [16]

3.3 Graph Laplacian:

- The Graph Laplacian is a fundamental matrix used in graph theory to analyze the structure of a graph. It is constructed from the graph's adjacency matrix and degree matrix. Specifically, the degree matrix is a diagonal matrix where each element represents the sum of the weights of edges connected to a node. The Laplacian matrix L is then defined as the difference between the degree matrix D , a diagonal matrix whose a th diagonal element is equal to the sum of the weights of all edges incident to the node a , And the adjacency matrix W (i.e., $L = D - W$) [2].

$$D = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

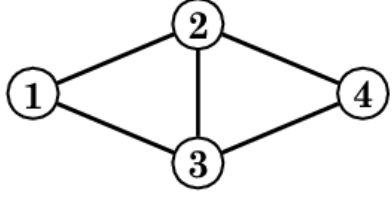
$$L = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ 0 & -1 & -1 & 2 \end{bmatrix}$$


Figure 3: Representation of the Laplacian matrix of a simple network using the Adjacency matrix and the Degree Matrix ($n=4$). [11]

The Laplacian matrix role in analyzing various structural properties of the graph, such as connectivity and clustering. By performing an eigen decomposition of the Laplacian matrix, we can obtain its eigenvalues and eigenvectors, which offer valuable insights into the graph's spectral characteristics [2].

The Laplacian matrix L is a symmetric positive semidefinite matrix with eigendecomposition:

$$L = U\Lambda U^T$$

In this decomposition, U is the matrix consisting of the eigenvectors of L as its columns, and Λ is a diagonal matrix with the corresponding eigenvalues on its diagonal. The matrix U is then used to compute the Graph Fourier Transform (GFT) of a signal s by [2]:

$$\tilde{s} = U^T s.$$

The inverse GFT is calculated by:

$$s = U\tilde{s}.$$

3.4 Graph Fourier Transform (GFT):

Consider an undirected, weighted graph $G = (V, E)$ composed of a vertex set V with n vertices and an edge set E , where each edge is defined by a pair of vertices (a, b) , where $a, b \in V$, and $w_{ab} \in \mathbb{R}^+$ is the edge weight between vertices a and b representing the strength of the connection between vertices a and b .

The weighted graph can be represented by an adjacency matrix W , where the element $W(a, b) = w_{ab}$ indicates the weight of the edge between vertex a and vertex b .

A graph signal assigns a specific value to each vertex, which can be represented as a vector $s = [s_1 s_2 \dots s_n]^T$. [2]

When computing the GFT a graph is constructed to capture the inter-node correlation and is used to compute the optimal decorrelating transform leveraging on spectral graph theory. It leverages the eigenvectors of the Graph Laplacian to transform signals defined on the graph into the frequency domain. By projecting the graph signal onto these eigenvectors, the GFT decomposes the signal into different frequency components. This transformation is particularly useful for analyzing and processing signals on irregular domains, such as graphs. High-frequency components in the transformed signal often correspond to anomalies or irregular patterns, making the GFT a powerful tool for anomaly detection in graph-based representations of data. By applying the GFT, we can effectively identify and isolate these high-frequency components to detect anomalies within the graph structure.

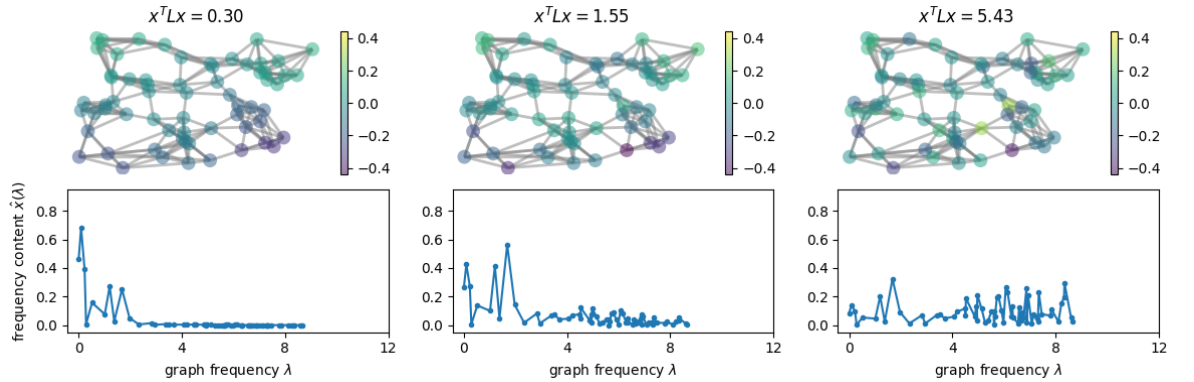


Figure 4: Visual representation of Graph Fourier Transform frequencies in a graph shown by eigenvalues. [17]

3.5 Semantic and Syntactic Relationships

- Semantic relationships capture the meaning or context of the text units. They indicate how similar or related the content of different text units is based on their meanings. Semantic relationships are typically calculated using text embeddings generated by advanced language models. These embeddings are high-dimensional vector representations that capture the semantic content of the text. The similarity between embeddings (e.g., cosine similarity) determines the strength of the semantic relationship.

$$\text{Cosine Similarity}(A, B) = \frac{A \cdot B}{\|A\| \cdot \|B\|}$$

- Syntactic relationships capture the structural or grammatical connections between text units. These relationships reflect the order and grammatical dependencies within the text. Syntactic relationships are often based on the sequential order of text units. For example, consecutive sentences in a paragraph are syntactically connected

4. Expected Achievements

This project aims to develop a robust anomaly detection system for textual data using a graph-based approach. Expected achievements include:

- A comprehensive graph model for representing text data.
- An efficient method for calculating the Graph Laplacian and performing GFT.
- An effective anomaly detection metric based on the transformed graph signal.
- Evaluation of the method on a dataset of articles with identified anomalies.

5. Engineering Process

5.1 Data Preprocessing:

Before constructing the graph, the textual data needs to be preprocessed and converted into a suitable format for analysis. This involves two main steps: tokenization and feature extraction. Tokenization involves splitting the text into smaller units, such as words, sentences or paragraphs. This can be formally defined as:

$$\text{Tokens} = \text{Tokenize}(\text{Text})$$

where $\text{Tokenize}(\text{Text})$ is the function that divides the text into tokens.

Feature extraction then generates embeddings for each text unit using a pre-trained language model. These embeddings capture the semantic meaning of the text and are used to calculate the semantic relationships between text units. Using a model like BERT, each tokenized sentence T_i can be converted into a high-dimensional embedding vector F_i :

$$F_i = \text{BERT}(T_i)$$

where F_i is a high dimensional embedding vector representing the semantic meaning of the sentence T_i .

These embeddings $F_1, F_2, F_3 \dots, F_n$ are high dimensional vectors that encapsulate the semantic information of each sentence. These vectors are subsequently used to compute the semantic relationships between sentences using cosine similarity, forming the basis for the edges in the graph.

5.2 Graph Construction:

Firstly, the textual data is represented as a graph $G = (V, E)$, where V is the set of nodes representing text units (such as sentences or paragraphs), and E is the set of edges representing the relationships between these text units with weight function w . The edges E capture two types of relationships: semantic and syntactic. Semantic weights calculated using cosine similarity:

$$w_{ij} = \text{Cosine Similarity}(F_i, F_j)$$

while syntactic weights are assigned a fixed value.

5.3 Computing The Graph Laplacian :

With the graph constructed, the Graph Laplacian L is computed. The weight matrix W contains the weights of the edges between nodes, while the degree matrix D is a diagonal matrix where each element $D(i, i)$ is the sum of the weights of the edges connected to node i . The Laplacian matrix is then defined as:

$$L = D - W.$$

5.4 Graph Fourier Transform (GFT):

Eigen decomposition of the Laplacian matrix is performed to obtain its eigenvalues (λ) and eigenvectors (U). The eigenvalues provide insights into the frequency components of the graph, while the eigenvectors form the basis for transforming the graph signal into the frequency domain through the Graph Fourier Transform (GFT).

$$\hat{s} = U^T s$$

5.5 Analyzing High-Frequency Components:

In the frequency domain, high-frequency components often indicate anomalies. The anomaly detection metric is based on the magnitude of these high-frequency components. For example, the anomaly score for each node can be calculated as:

$$\delta_{LAD}(x) = \sum_{j=1}^m \lambda_j \tilde{s}_j^2$$

where λ_j are the eigenvalues \tilde{s} and are the high-frequency components of the signal.

A threshold is then applied to classify nodes as anomalous or normal. Nodes with scores exceeding the threshold are flagged as anomalies. The threshold value is determined through validation or expert knowledge.

$$\delta_{LAD}(x) \geq \eta \implies \text{Node } x \text{ is an anomaly}$$

5.6 Evaluating Anomalies:

After running the anomaly detection process, the system's predicted anomalies are compared against a ground truth dataset, which contains known true anomalies. By calculating evaluation metrics such as precision, recall, and F1-score.

5.7 System Class Diagram:

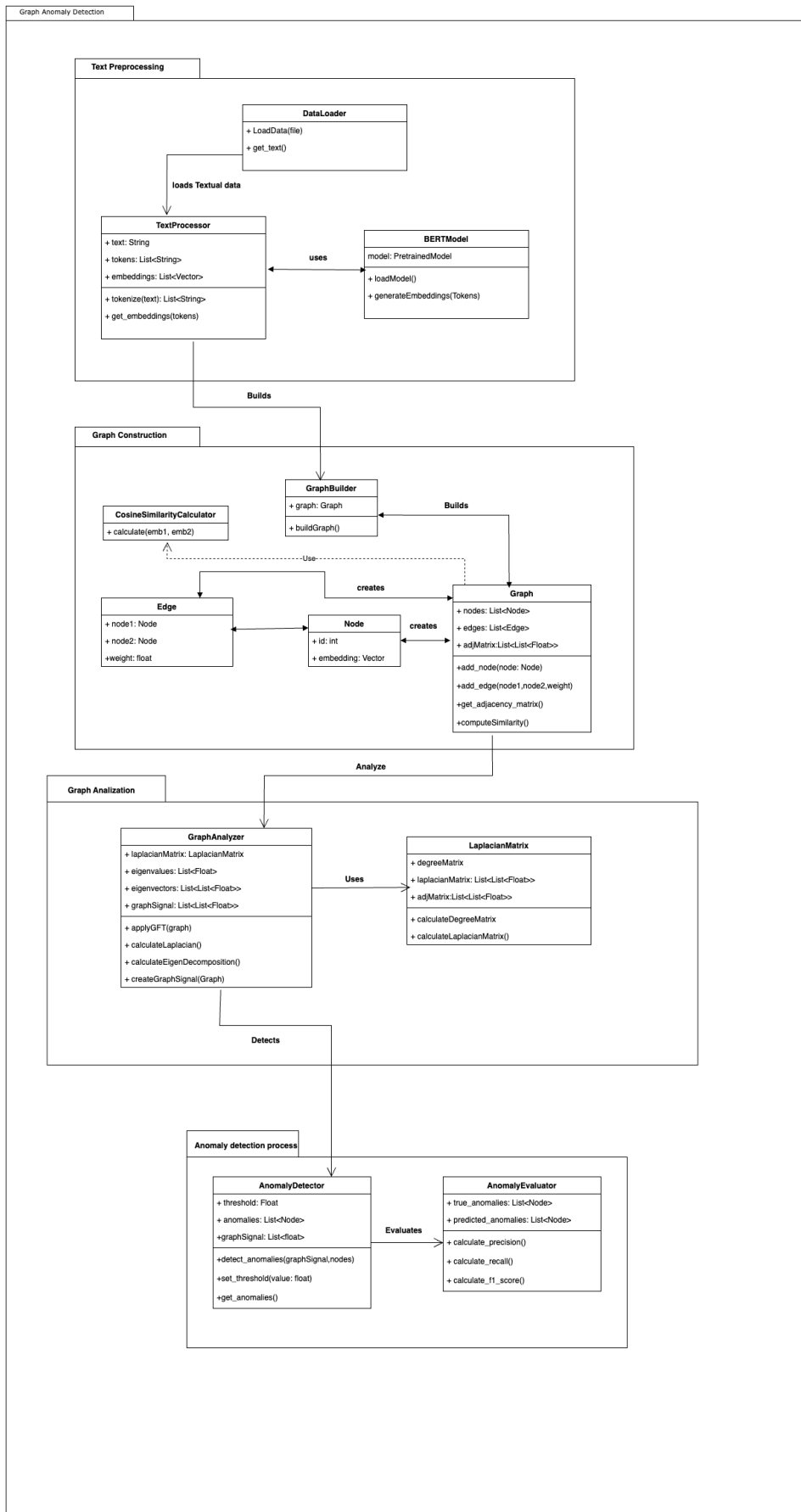


Figure 5: The following class diagram represents the structure of the anomaly detection system. It highlights the main components involved in data preprocessing, Graph construction, Graph Analyzation, Anomaly detection and the evaluation process.

5.8 Hyperparameters

1. **Text Embedding Dimension:** 768 (for BERT embeddings)
2. **Pre-trained Model:** BERT-base-uncased
3. **Semantic Similarity Threshold:** 0.8
4. **Fixed Syntactic Edge Weight:** 1
5. **Laplacian Type:** Symmetric normalized
6. **Number of Eigenvalues:** 50
7. **Anomaly Detection Threshold:** 1.5
8. **High-Frequency Component Range:** Top 10% of the frequencies
9. **Learning Rate:** 0.001 (if applicable)
10. **Number of Epochs:** 10 (if applicable)
11. **Batch Size:** 32 (if applicable)

These are the suggested values of the main leading hyperparameters which will be adjusted during the experimental studies.

6. Evaluation/Verification Plan

1. **Testing:**
 - Test the anomaly detection system on a diverse dataset with known anomalies to assess its performance.
 - Ensure that the system can correctly identify anomalies and distinguish them from normal text units.
2. **Validation:**
 - Refine the threshold and other parameters based on validation results to optimize the system's performance.

3. Units Tests:

Case	Test Case	Expected Result
1	Test extract features multiple sentences	Tokenized output should be a list of sentences split correctly from the input text.
2	Test extract features single sentence	Embedding should be a 768-dimensional vector for a single sentence.
3	Test graph construction node count	Graph should have a number of nodes equal to the number of sentences.
4	Test calculate weights cosine similarity	Weights should be within the range $[-1, 1]$ based on cosine similarity.
5	Test compute Laplacian symmetry	The Laplacian matrix should be symmetric.
6	Test apply GFT transformation	The transformed signal should have the correct dimensionality and represent the frequency components of the graph signal accurately.
7	Pass an empty string to the feature extraction function	The output should handle empty input gracefully, returning either an empty list or an appropriate message without error.
8	Verify that the number of edges created in the graph corresponds to the expected number of connections based on the text input and similarity measure.	The number of edges should match the pairwise relationships between the nodes based on cosine similarity.
9	Compute the eigenvalues of the Laplacian matrix.	Eigenvalues should be non-negative (since the Laplacian is a positive semi-definite matrix) and correctly computed.

7. Implementation Plan

1. **Data Collection:**
 - Gather a dataset of articles from various sources. Ensure that the dataset includes examples with known anomalies for testing and validation purposes.
2. **Graph Construction and Preprocessing:**
 - Implement preprocessing steps to tokenize the text and extract features.
 - Construct the graph model by defining nodes, edges, and calculating weights.
3. **Graph Laplacian and GFT:**
 - Compute the Graph Laplacian and perform the eigen decomposition.
 - Apply the GFT to transform the graph signal into the frequency domain.
4. **Anomaly Detection:**
 - Implement the anomaly detection metric and apply thresholding to identify anomalies.
 - Fine-tune hyperparameters to optimize detection performance.
5. **Evaluation and Iteration:**
 - Evaluate the system's performance on the test dataset.
 - Iterate on the model by refining preprocessing, feature extraction, and graph construction based on evaluation results.

8. Bibliography

[1] Mehrotra, K.G., Mohan, C.K., Huang, H. (2017). Model-Based Anomaly Detection Approaches. In: Anomaly Detection Principles and Algorithms. Terrorism, Security, and Computation. Springer, Cham. https://doi.org/10.1007/978-3-319-67526-8_5

[2] Verdoja, Francesco, and Marco Grangetto. "Graph Laplacian for image anomaly detection." *Machine Vision and Applications* 31.1 (2020): 11. [arXiv:1802.09843](https://arxiv.org/abs/1802.09843)

[3] Oswal, S., Shinde, S., Vijayalakshmi, M. (2023). A Survey of Statistical, Machine Learning, and Deep Learning-Based Anomaly Detection Techniques for Time Series. In: Garg, D., Narayana, V.A., Suganthan, P.N., Anguera, J., Koppula, V.K., Gupta, S.K. (eds) Advanced Computing. IACC 2022. Communications in Computer and Information Science, vol 1782. Springer, Cham. https://doi.org/10.1007/978-3-031-35644-5_17

[4] Jayabharathi, S., Ilango, V. (2023). Anomaly Detection Using Machine Learning Techniques: A Systematic Review. In: Das, S., Saha, S., Coello Coello, C.A., Bansal, J.C. (eds) Advances in Data-Driven Computing and Intelligent Systems. ADCIS 2022. Lecture

Notes in Networks and Systems, vol 698. Springer, Singapore. https://doi.org/10.1007/978-981-99-3250-4_42

[5] Jayabharathi, S., Ilango, V. (2023). Anomaly Detection Using Machine Learning Techniques: A Systematic Review. In: Das, S., Saha, S., Coello Coello, C.A., Bansal, J.C. (eds) Advances in Data-Driven Computing and Intelligent Systems. ADCIS 2022. Lecture Notes in Networks and Systems, vol 698. Springer, Singapore. https://doi.org/10.1007/978-981-99-3250-4_42

[6] Jafari, Amir. "A deep learning anomaly detection method in textual data." *arXiv preprint arXiv:2211.13900* (2022).

[7] [Comparative Analysis of Anomaly Detection Algorithms in Text Data](<https://aclanthology.org/2023.ranlp-1.131>) (Xu et al., RANLP 2023)

[8] Gorokhov, O., Petrovskiy, M., Mashechkin, I. (2017). Convolutional Neural Networks for Unsupervised Anomaly Detection in Text Data. In: Yin, H., *et al.* Intelligent Data Engineering and Automated Learning – IDEAL 2017. IDEAL 2017. Lecture Notes in Computer Science(), vol 10585. Springer, Cham. https://doi.org/10.1007/978-3-319-68935-7_54

[9] S. Khazai, S. Homayouni, A. Safari and B. Mojaradi, "Anomaly Detection in Hyperspectral Images Based on an Adaptive Support Vector Method," in *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 4, pp. 646-650, July 2011, doi: 10.1109/LGRS.2010.2098842. keywords: {Kernel;Pixel;Hyperspectral imaging;Estimation;Support vector machines;Detectors;Anomaly detection (AD);Gaussian kernel;hyperspectral images;support vector (SV) data description (SVDD)}

[10] Akoglu, L., Tong, H. & Koutra, D. Graph based anomaly detection and description: a survey. *Data Min Knowl Disc* **29**, 626–688 (2015). <https://doi.org/10.1007/s10618-014-0365-y>

[11] Sawada, Ryosuke & Sakumoto, Yusuke & Takano, Chisa & Aida, Masaki. (2016). Experimental study on relationship between indices of network structure and spectral distribution of graphs.

[12] Bertozzi, A., Flenner, A.: Diffuse interface models on graphs for classification of high dimensional data. *Multisc. Model. Simul.* 10(3), 1090–1118 (2012). <https://doi.org/10.1137/11083109X>

[13] A. Anandkumar, L. Tong and A. Swami, "Detection of Gauss–Markov Random Fields With Nearest-Neighbor Dependency," in *IEEE Transactions on Information Theory*, vol. 55, no. 2, pp. 816-827, Feb. 2009, doi: 10.1109/TIT.2008.2009855.

[14] Chalapathy, R., Chawla, S.: Deep learning for anomaly detection: a survey. *arXiv:1901.03407 [cs, stat]* (2019)

[15] Defferrard, Michaël, Xavier Bresson and Pierre Vandergheynst. "Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering." *Neural Information Processing Systems* (2016).

[16] Embeddings: Meaning, Examples and How To Compute, *arize* , <https://arize.com/blog-course/embeddings-meaning-examples-and-how-to-compute>, Accessed on September 9, 2024.

[17] Fourier Transform : Image Example , *pyGSP*, https://pygsp.readthedocs.io/en/latest/examples/fourier_transform.html, Accessed on September 13, 2024.