mathematics

MDPI

Article

A Unified Graph Theory Approach: Clustering and Learning in Criminal Data

Haifa Al-Ibrahim * and Heba Kurdi

Computer Science Department, College of Computer and Information Sciences, King Saud University, Riyadh 11451, Saudi Arabia; hkurdi@ksu.edu.sa
* Correspondence: haifa.i.alibrahim@gmail.com

Abstract: Crime report clustering plays a critical role in modern law enforcement, enabling the identification of patterns and trends essential for proactive policing. However, traditional clustering approaches face significant challenges with the complex, unstructured nature of crime reports and their inherent sparse relationships. While graph-based clustering shows promise, issues of noise sensitivity and data sparsity persist. This study introduces a unified approach integrating spectral graph-based clustering with Graph Convolutional Networks (GCN) to address these challenges. The proposed approach encompasses data collection, preprocessing, linguistic feature extraction, vectorization, graph construction, graph learning, and clustering to effectively capture the intricate similarities between crime reports. The proposed approach achieved significant improvements over existing methods: a Silhouette Score of 0.77, a Davies–Bouldin Index of 0.51, and consistent performance across varying dataset sizes (100–1000 nodes). These results demonstrate the potential for enhanced crime pattern detection in law enforcement operations.

Keywords: crime report analysis; graph-based clustering; graph learning techniques; GCN; spectral clustering

MSC: 68R10; 05C90

check for updates

Citation: Al-Ibrahim, H.; Kurdi, H. A Unified Graph Theory Approach: Clustering and Learning in Criminal Data. *Mathematics* **2024**, *12*, 3865. https://doi.org/10.3390/math12233865

Academic Editor: Andrea Scozzari

Received: 8 October 2024
Revised: 30 November 2024
Accepted: 2 December 2024
Published: 9 December 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The analysis of crime reports and extracting valuable insights from these documents are crucial for developing effective crime intervention and prevention strategies [1]. However, the manual processing of large volumes of data presents significant challenges in terms of time, resources, and accuracy. To address these issues, researchers have increasingly turned to advanced machine learning techniques, particularly clustering approaches, which offer several advantages for crime report analysis. By grouping reports based on contextual similarities, crime report clustering enables investigators to efficiently identify specific crime types, uncover non-obvious relationships between seemingly disparate reports, and adapt to emerging crime trends without requiring pre-defined categories [2].

The particular difficulties presented by crime report data, such as the complex and their inherent sparse relationships, have restricted the use of clustering algorithms in crime analysis, despite their potential advantages. Traditional clustering algorithms, including K-means, struggle to handle these challenges effectively. Furthermore, K-means is extremely sensitive to its initialization parameters and tends to become stuck in local optima, leading to inconsistent clustering results across different initialization parameters [3,4]. Consequently, there is a pressing need for more advanced approaches that can overcome these limitations and unlock the full potential of crime report clustering.

Graph-based clustering approaches have emerged as a promising solution to the challenges faced by traditional clustering methods in crime report analysis [5]. By representing crime reports as a graph, these approaches can capture complex relationships within the data and identify clusters more effectively. However, when applied to real-world crime

report data, graph clustering algorithms face significant challenges due to the presence of noise, outliers, and inherent data sparsity [5].

In high-dimensional sparse spaces, such as those often encountered in crime report data, conventional distance metrics become increasingly unreliable, a phenomenon known as the 'curse of dimensionality' [6]. This issue can lead to poor clustering performance and difficulty in distinguishing between meaningful patterns and random variations. Moreover, sparse representations require specialized algorithms to maintain efficiency, further complicating the clustering process.

To address these challenges, graph learning techniques can be used as a preprocessing step. By learning informative representations of the graph elements, these techniques can help in identifying and handling outliers and noise more effectively, ultimately improving the quality of the clustering results. Graph learning has been successfully applied in various domains [7,8], where it has demonstrated its ability to uncover hidden patterns and enhance clustering performance.

Despite the potential benefits of graph learning in crime report clustering, its adoption in this domain remains limited. A recent survey conducted in 2023 [9] highlights the limited quantity of research exploring the integration of graph learning techniques with crime report clustering. This lack of research presents an opportunity to investigate the potential of graph learning in addressing the challenges posed by crime report data and improving the effectiveness of clustering approaches in this context.

This study aims to bridge this gap by proposing a unified graph-based approach that integrates spectral clustering with Graph Convolutional Networks (GCN) for graph learning. The proposed approach addresses the limitations of traditional clustering algorithms and leverages the power of graph learning to enhance the performance of crime report clustering. The main contributions of this research are:

1. The development of a unified graph-based approach that combines spectral graph-based clustering with GCN graph learning, inspired by different algorithms for graph construction [5,10], and graph learning [11].
2. The construction of a crime graph dataset using an ε-ball graph method [5] to represent criminal text.
3. An extensive performance evaluation of the proposed approach, comparing it with different methods using the Silhouette Score and Davies–Bouldin Index, which provide different perspectives on the quality of a clustering result.

The remainder of this paper is organized as follows: Section 2 reviews related work; Section 3 presents system design; Section 4 details system implementation; Section 5 discusses results; and Section 6 concludes with future research directions.

2. Related Works

This section presents a comprehensive analysis of existing approaches in crime report clustering and document clustering as extended for our paper [12]. It is organized around three key themes: traditional clustering approaches, graph-based methods, and learning-based techniques. Table 1 provides a comparative overview of key works discussed in this section.

2.1. Traditional Clustering

Early approaches to document clustering primarily relied on conventional clustering algorithms adapted for textual data. These methods established fundamental techniques for grouping documents but faced significant limitations in handling the complexity of data.

Mishra et al. [13] demonstrated these limitations in their analysis of social media data, where even with sophisticated preprocessing, including stop-word removal, sentence tokenization, POS tagging, sentiment analysis, and Named Entity Recognition, traditional K-means clustering achieved only moderate success (Silhouette Scores of 0.132). Their work highlighted the need for more sophisticated approaches that could better handle the inherent complexity of crime-related text data.

2.2. Graph-Based Clustering

Graph-based clustering approaches emerged as a response to the limitations of traditional methods, offering more nuanced analysis of relationships between crime reports. These methods can be categorized into three main approaches: partitioning methods, overlapping methods, and hierarchical methods.

P. Das et al. [14] proposed a hierarchical graph-based clustering technique for crime relation extraction, using 210,000 Indian crime reports. The method employed extensive preprocessing, Word2Vec, and weighted graph construction using cosine similarity. Their approach, while effective for single-category assignment, revealed limitations in handling cases where crimes span multiple categories.

Pramanik et al. [2,5] advanced this direction through graph-based fuzzy clustering, introducing more flexible assignment mechanisms. Their process involved comprehensive preprocessing, including stop-word removal and lemmatization. For vectorization, they employed advanced techniques such as SBERT [2] or Universal Sentence Encoder (USE) [5]. The researchers constructed an undirected graph using cosine similarity to represent relationships between reports. Their work with diverse datasets (58,951 Indian, 65,621 USA, and 67,852 UAE reports) demonstrated superior performance in handling multi-label scenarios, achieving improved cluster validation metrics: a Silhouette Score of 0.61, a Davies–Bouldin Index of 0.47, and a Dunn's Index of 1.69.

Furthermore, research has explored the use of fuzzy hypergraph-based models for predicting crime-prone areas. Pm et al. [15] introduced a multilevel structure that captured both geographical and contextual relationships in crime data. The proposed model incorporated various environmental and social factors, including proximity to ATMs and highways, literacy rates, and existing crime patterns. Their approach demonstrated particular strength in predicting crime-prone areas but highlighted scalability challenges when applied to large-scale textual analysis.

Recent advances in overlapping clustering include three-way clustering approaches [16], which explicitly model uncertainty in cluster assignments. While these methods show promise for handling ambiguous cases in some areas of document clustering, they typically require higher computational resources.

2.3. Learning Approaches in Graph Clustering

The integration of learning techniques with graph-based clustering represents the current advancement in document analysis. Zhang et al. [7] demonstrated significant improvements through attributed graph clustering, particularly in handling high-dimensional sparse spaces typical of crime reports. Their work explored the integration of graph learning to enhance clustering performance, addressing key challenges such as time and memory consumption typically associated with deep neural networks. The proposed approach was tested on diverse text datasets, including Cora, Citeseer, Pubmed, Wiki, Large Cora, and Coauthor Phy, highlighting its versatility and adaptability. This innovative approach efficiently learned smaller graphs from large attributed graphs, offering a scalable and efficient solution for complex data analysis tasks.

While existing research has advanced individual components of crime report clustering, there are significant gaps remaining:

- A lack of unified frameworks that systematically integrate preprocessing and post-processing techniques;
- Limited exploration of graph learning models specifically optimized for crime report characteristics.

Our work addresses these gaps through a unified approach that integrates spectral graph-based clustering with GCN graph learning, offering improved performance while maintaining computational efficiency.

Table 1. Summary of related works in graph clustering.

Ref.	Main Contributions	Algorithm Type	Data Type	Dataset	Preprocessing	Vectorization	Graph Construction	Clustering Measures
[14]	Hierarchical graph-based clustering for relation extraction	Hierarchical Clustering	Crime text	210 k crime reports	Stop-word removal, stemming, POS tagging, named entity recognition	Word2Vec	Weighted graph, cosine similarity	Internal cluster validation indices: (SF), (NI), (CH), (SL), (DB), (DN) External cluster validation indices: (Pr), (P), (R), (F), and (RI)
[5]	Graph-based fuzzy clustering to label the reports by multiple crime types	Overlapping Clustering	Crime text	−200 k news from Kaggle—Crime reports for −58,951 India, −65,621 USA −67,852 UAE	Sentence segmented, word tokenizer stop-word removal, lemmatization, POS tagging, named entity recognition	Universal Sentence Encoder	Undirected graph, cosine similarity	Internal cluster validation indices: SL, DN, DB, XB, CH, IN Overlapping cluster validation metrics: PC, PE, DI, GD, KI
[7]	Graph learning for attributed graph clustering, method scalable to large graphs to learn small graphs	Attributed graph clustering	General text	Cora, Citeseer, Pubmed, Wiki, Large Cora, Coauthor Phy	–	–	–	Clustering accuracy (ACC), (NMI) and (F1).
[15]	Fuzzy hypergraph-based model for prediction of crime prone area	Overlapping Clustering	Crime location	Excel sheet from Police Station	–	–	Google chart from Excel sheet	Accuracy, precision, recall, and F1 score
[2]	Graph-based fuzzy clustering algorithm for crime report labeling	Overlapping Clustering	Crime text	200 k news from Kaggle	Stop-word removal, lemmatization	SBERT	Undirected graph, cosine similarity	Internal cluster validation indices: SL, DN, DB, XB, CH, IN Overlapping cluster validation metrics: PC, PE, DI, GD, KI
[13]	Cluster-based knowledge graph and entity-relation representation present a visual representation of data using knowledge graph	K-means clustering	Tourism and economy	100 k tweets	Stop-word removal, sentence tokenizer, POS tagging, sentiment polarity, named entity recognition	Word2Vec	Directed graph	Silhouette Score
[17]	Community detection method for social network community detection	Network representation learning (NRL)	Network dataset	Cora, Wiki, Blog Catalog, Karate	–	Node2Vec	Undirected graph	Modularity ARI NMI

3. System Design

This section will cover the architecture of the proposed system in detail. The main objective of this research is to explore the potential of enhancing crime clustering through graph learning.

The proposed system, as shown in Figure 1, involves several steps: data collection, preprocessing, linguistic feature extraction, vectorization, graph construction, graph learning, and clustering. These steps can be described as follows:

- Collecting Crime Dataset: Crime-related news articles are collected from established U.S. news organizations published between 2012 and 2022, representing the most reliable publicly available crime reports. Data validation follows a comprehensive three-tier approach. First is ensuring source credibility by exclusively collecting articles from established news organizations. Second is standardization via preprocessing steps, including stop-word removal, lemmatization, and noun phrase extraction. Third is establishing reliability through multiple quantitative measures, including internal validation using established indices (Silhouette, Davies–Bouldin). The dataset has proven particularly valuable for crime analysis research, as evidenced by its use in studies focusing on crime report clustering [5]. This extensive use in peer-reviewed research further validates the dataset's reliability and applicability for crime analysis studies.
- Preprocessing: The collected crime texts go through a systematic preprocessing pipeline that directly impacts clustering quality. First, sentence segmentation and word tokenization transform unstructured text into analyzable units, reducing textual noise by establishing proper sentence boundaries and word delineation. This step enables accurate feature extraction and improves clustering accuracy by ensuring proper context preservation in crime reports. Second, stop-word removal significantly reduces dimensionality by eliminating non-informative words (e.g., 'the', 'and', 'of'), which improves cluster cohesion by focusing on content-bearing terms. Third, lemmatization consolidates different forms of words into their base form (e.g., 'killed', 'killing', 'kills' $\rightarrow$ 'kill'), reducing vocabulary sparsity while preserving semantic meaning. This normalization improves semantic similarity calculations between crime reports and enhances cluster distinctiveness.
- Vectorization: The generation of report embeddings for crime text is achieved through the utilization of the Universal Sentence Encoder (USE) [18]. This model is known for its effectiveness in generating sentence embeddings and is widely employed in various multi-task learning tasks, such as sentiment analysis, sentence similarity, clustering, and classification. The USE uses a combination of transformers and Deep Averaging Network (DAN) encoders to convert sentences into 512-dimensional vectors. The transformer model captures contextual information, while the DAN encoder computes embeddings for individual unigrams or bigrams.
- Graph Construction: The construction of a graph to represent the relationships between different crime texts plays a crucial role in the clustering process, as it enables the extraction of meaningful insights and patterns from the text data. This graph-based representation facilitates an enhanced understanding of the interconnections and similarities between crime texts. In this research, the graph is built to represent the relationships between crime texts, extracting meaningful insights and patterns. Each text is assigned a node, represented by vector embeddings using the Universal Sentence Encoder (USE). The graph's edges represent the relationships or similarities between the crime texts. By connecting the nodes with edges, the graph highlights the connections and proximity between different crime texts, forming clusters of related texts. This graph-based approach enables researchers to uncover valuable insights and patterns by examining the clustering structure and analyzing the relationships between crime texts.

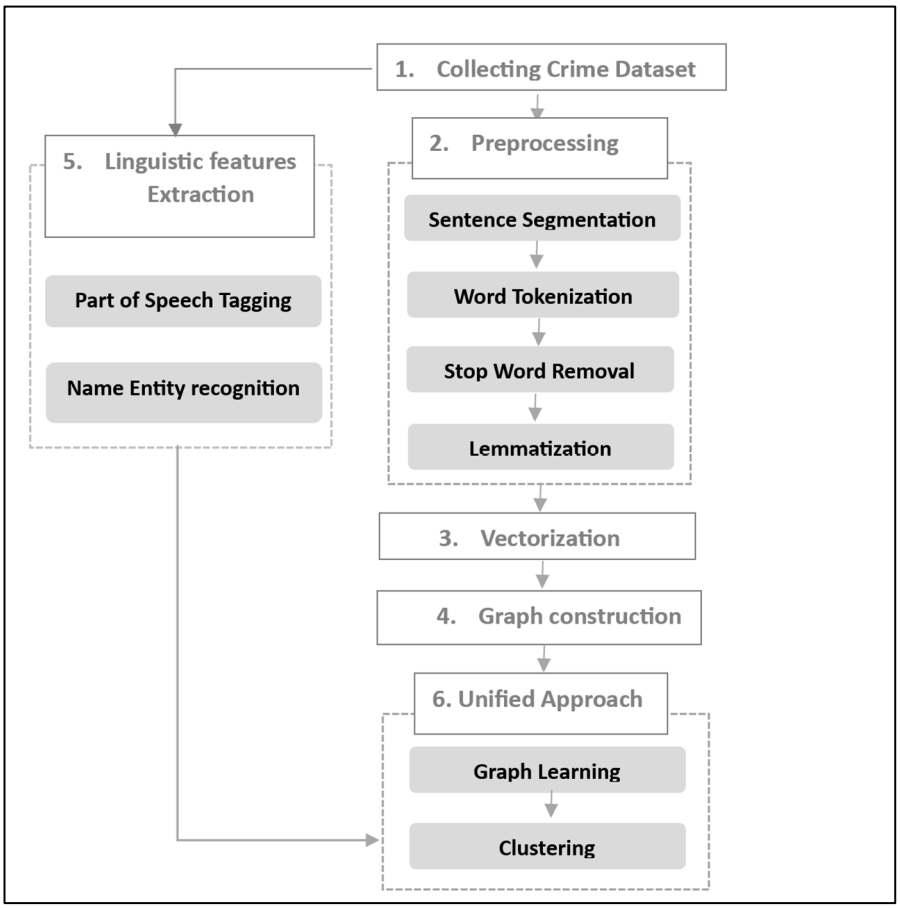


Figure 1. System architecture of the proposed solution.

- Linguistic Feature Extraction: This involves identifying Named Entities (NER) and Part of Speech (POS) tags. NER involves identifying and classifying entities in crime text, such as persons, locations, organizations, and dates, while POS tagging labels words with their corresponding parts of speech, such as nouns, verbs, adjectives, or adverbs., providing context and revealing patterns [19]. By incorporating NER and POS, researchers can identify clusters with common entities or similar patterns and gain a deeper understanding of crime data.
- Graph Learning: Graph-based deep learning approaches have emerged as powerful tools for processing graph-structured data and understanding complex relationships within the graph [20]. Graph Convolutional Networks (GCNs) have demonstrated particular success in this domain, achieving state-of-the-art performance on various graph-based tasks [7]. Our experimental results show that incorporating graph learning improved the clustering performance compared to traditional text-based methods, demonstrating its effectiveness in capturing both semantic similarity and relational information between the reports.
- Clustering: Once the graph embeddings are learned, clustering algorithms are applied to group similar crime reports together. The choice of clustering algorithm can vary depending on the specific requirements and characteristics of the data. Common clustering algorithms such as K-means, hierarchical clustering, or community detection algorithms can be employed [5].

4. System Implementation

The proposed system has been implemented by using a different set of Python 3.10 modules, including NumPy 1.12.0 for mathematical operations on arrays, NLTK for natural language processing, Spacy and NetworkX 1.11 for graph manipulation, the Py-

Torch library for deep learning, the PyTorch Geometric library for graph neural networks, and the scikit-learn library for clustering algorithms and evaluation metrics. Furthermore, there are multiple phases that were performed to implement the proposed unified solution, which will be explained in detail below with the results for each phase.

- Dataset Collection: The dataset was collected from Kaggle [21], encompassing approximately 210,000 news headlines spanning the period from 2012 to 2022. This dataset serves as a benchmark for computational linguistic tasks and comprises 42 news categories, including headline, authors, links, short description, and publication date. For the purposes of our evaluation, we concentrated on the crime news category, which consisted of approximately 3562 instances. In our analysis, we combined the short description with the headline of each report, excluding any other accompanying information.
- Preprocessing: The crime reports collected for our research were preprocessed using the NLTK module, which involved performing crucial tasks including sentence segmentation, word tokenization, stop-word removal, and lemmatization. This preprocessing was carried out to minimize noise and enhance the accuracy of the analysis, as demonstrated in Table 2.

Table 2. Description of dataset.

Crime Dataset		
Number of Reports	**Number of Words**	**Number of Words After Preprocessing**
3562	82,546	37,952

Several preprocessing steps were taken to prepare the data for analysis of the crime reports displayed in Figure 2. As shown below, each stage was completed focusing on a particular objective:

1. Sentence Segmentation: The crime report was first divided into separate sentences. Treating each sentence as a separate unit allows for a more concentrated analysis.
2. Word Tokenization: Next, the text was tokenized, breaking it up into individual words. This allows for a closer look at the text's substance and allows for further word-by-word analysis.
3. Lowercase Conversion: All terms were changed to lowercase in order to avoid case-related inconsistencies and to guarantee uniformity. By performing this standardization, redundant information is removed, and the accuracy of the ensuing analysis is improved.
4. Stop-Word Removal: Common articles and prepositions (such as "in", "a", "as", and "of") were eliminated from the tokenized terms. In order to concentrate on more important content, these stop words are usually removed from texts when they do not significantly add to their meaning.
5. Elimination of Punctuation: All commas and periods were removed from the tokenized words. In order to ensure accurate processing and further analysis, this stage simplifies the content.
6. Elimination of Non-Alphabetic Items: Characters that are not alphanumeric and words with no discernible significance were eliminated. By removing unnecessary information, this cleansing procedure helps improve the data's quality.
7. Lastly, lemmatization was used to condense terms to their most basic or dictionary forms. Words such as "charged" were changed to "charge", and "friends" to "friend". By grouping related terms together, this normalizing technique helps decrease the dimensionality of the data.

By accurately executing these preprocessing steps, the crime reports were refined and standardized, facilitating subsequent analysis and clustering. The resulting informative representations of the graph elements, combined with linguistic feature extraction, enable

the grouping of similar crime reports together, contributing to a more comprehensive understanding and examination of the data.

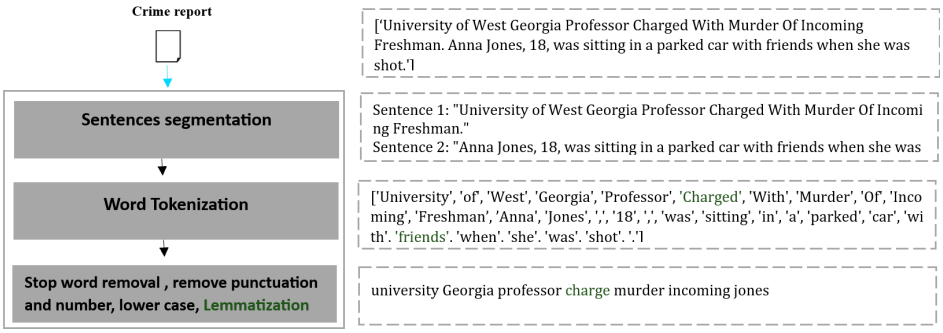


Figure 2. Preprocessing pipeline for crime report text data.

- Vectorization: The Universal Sentence Encoder (USE) was employed to convert textual data into high-dimensional vectors, chosen for its significant advantages in our analysis. The transformer-based architecture of the USE excels at capturing complex semantic relationships and contextual information, which are critical for analyzing crime reports. Specifically, the 512-dimensional embeddings provide an optimal balance between information richness and computational efficiency. This dimensionality is supported by empirical research indicating that such a size effectively captures nuanced semantic features while minimizing the risk of overfitting, especially when training data are limited [5,22]. The resulting vectors, referred to as report embeddings, served as valuable representations of the textual data and are utilized in constructing graphs. By using these embeddings as nodes in the graph construction process, we can accurately represent the semantic relationships and similarities between different crime reports. This approach allowed us to leverage the power of deep learning to transform the raw text data into meaningful numerical representations, facilitating more effective analysis and interpretation of the underlying semantic structure within the crime reports, as demonstrated in Figure 3.

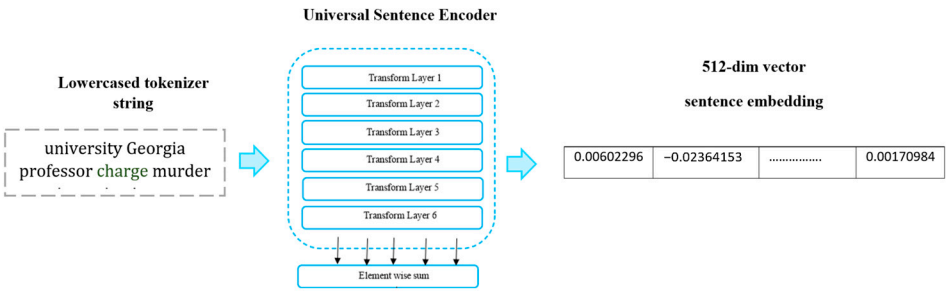


Figure 3. Vectorization with the Universal Sentence Encoder.

- Graph Construction: To construct a graph from unstructured crime text, the ε-ball graph method [5] was implemented using the network Python 3.10 library. We use an approach where each embedding of a crime report acts as a node in the graph. The cosine similarity between the two nodes' respective embeddings is used to calculate the weight of the edge linking them. This approach is illustrated in Algorithm 1.
- Linguistic Feature Extraction: To identify and extract Named Entities (NER) and Parts of Speech (POS) as node features, the Spacy Python library was used. This utilization of Spacy enabled us to incorporate additional syntactic and entity-based information into our analysis. For instance, consider the following example sentence: "University of West Georgia Professor Charged with Murder of Incoming Freshman". Using Spacy, we were able to identify the following Named Entities: "University of West Georgia" (Organization), "Anna Jones" (Person), "18" (Date), as shown in Figure 4.

Algorithm 1. Construct Graph

Inputs: —report_embedding: a list of embeddings for each report in the crime dataset
—threshold: the minimum cosine similarity threshold for creating an edge
Output: graph: a constructed graph based on the input data
Process:

1. **Initialize graph**

//Add node to graph

2. **For each** report_embedding in the crime dataset
3. Add_Node(graph, vector = report_embeddings)

//Add an edge between pairs of nodes.

4. **For each** report_embedding in the crime dataset
5. Calculate cosine similarity between pairs of nodes.
6. IF **similarity** > **threshold**
7. Add_Edge(graph, weight = similarity)
8. return graph

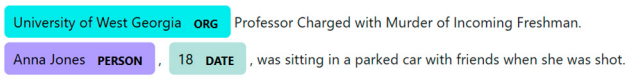


Figure 4. Named Entity Recognition for crime text.

Additionally, the Part of Speech (POS) tags extracted from the sentence included "University" (noun) "of" (preposition), "West" (proper noun), "Georgia" (proper noun), "Professor" (noun), "Charged" (verb) "With" (preposition), "Murder" (noun) as shown in Figure 5.

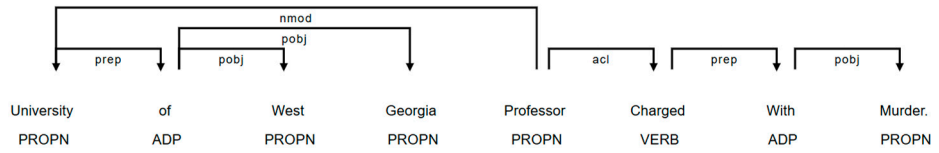


Figure 5. Part of Speech tag for crime text.

By incorporating these NER and POS features as node attributes, we enriched the graph representation of the text, enabling a more comprehensive analysis that takes into account both syntactic structure and specific named entities.

- Graph Learning: The graph learning implementation utilizes a Graph Convolutional Network (GCN) model, defined as a PyTorch module, to learn embeddings for the nodes in the graph. The GCN consists of two graph convolutional layers with ReLU activation, and the forward pass applies these layers sequentially to the graph data. The GCN model was trained using the Adam optimizer, minimizing the reconstruction loss between the output and input features. The Adam optimizer was selected based on the key advantages of Adam: adaptive learning rate adjustment reducing manual tuning requirements, momentum-based updates improving convergence in sparse graph structures, and efficient handling of noisy gradients common in crime report data. The trained model is then used to obtain the learned node embeddings, which are converted to a NumPy array for further use.
- Clustering: Clustering implementation utilizes the spectral clustering algorithm to cluster nodes in the input graph. It takes the graph and the desired number of clusters as parameters and returns a list of cluster labels for each node. To determine the optimal number of clusters, two methods are employed: Silhouette Analysis and the Elbow Method [23]. After determining the optimal number of clusters through these methods, the implementation creates an adjacency matrix representation of the graph using the NetworkX library. Then, it applies the spectral clustering algorithm from the scikit-learn library.

5. Results and Discussion

This section presents the results of constructing the graph as well as the performance results of the proposed unified solution that integrates the spectral graph-based clustering with GCN graph learning. Moreover, the results of the proposed solution were compared to different clustering algorithms, which include K-means clustering and spectral graph-based clustering.

5.1. Graph Construction

As mentioned previously, one of the main contributions of the proposed research is creating a crime graph dataset by constructing an undirected graph using an ε-ball graph method [10] to represent criminal text. In this section, we present our graph construction results and analyze their structural properties through quantitative metrics. To evaluate the structural characteristics of our constructed graphs, we employ the clustering coefficient (C), a fundamental metric in graph theory that quantifies the degree of node clustering in the graph. For each node i in the graph, we first compute its local clustering coefficient Ci using:

$$Ci = \frac{(2 \times Li)}{(ki \times (ki - 1)} \quad (1)$$

where Li is the number of links between neighbors of node i, and ki is the degree of node i. The global clustering coefficient C is then calculated by averaging local coefficients across all N nodes:

$$C = \frac{\sum_{i=1}^{N} Ci}{N} \quad (2)$$

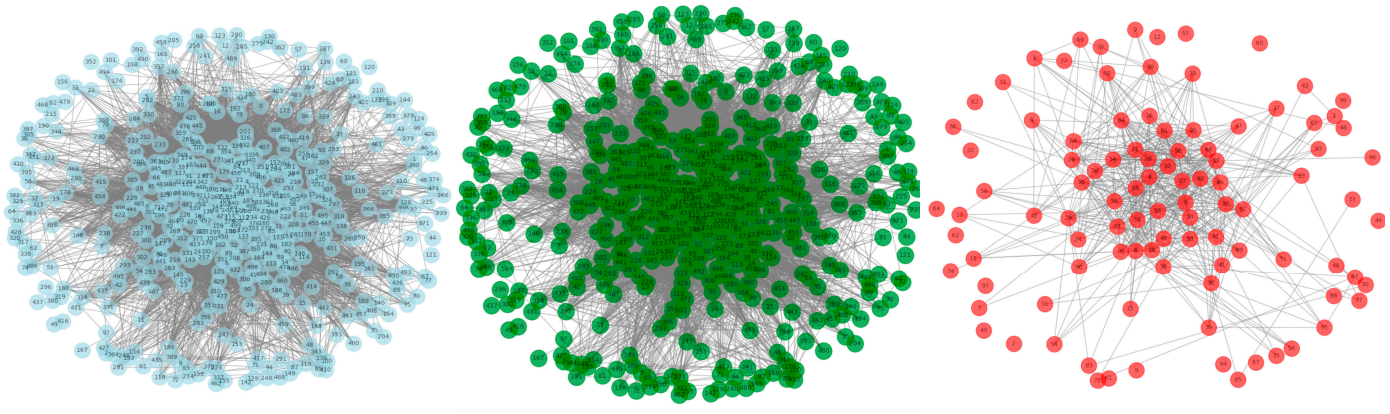
This metric ranges from 0 to 1, where higher values indicate stronger local connectivity patterns. In the context of crime report analysis, a high clustering coefficient suggests that related crime reports form well-defined communities, while lower values indicate more dispersed relationships [24].

Three crime report graphs were constructed to analyze how network properties scale with size (Table 3 and Figure 6). The analysis demonstrates a clear relationship between graph size and structural cohesion. In the small-scale graph (100 nodes), we observed 545 edges with a network density of 0.11 and a clustering coefficient of 0.46, indicating relatively sparse connectivity and a weak community structure. The medium-scale graph (500 nodes) exhibited markedly improved structural properties, containing 19,494 edges with a slightly higher density of 0.15 and a clustering coefficient of 0.58, suggesting a stronger community structure compared to the 100-node graph. Most notably, the large-scale graph (1000 nodes) demonstrated optimal structural characteristics with 83,226 edges, a density of 0.167, and a clustering coefficient of 0.62, indicating a strong community structure and strong local connectivity patterns.

The repetitive increase in the clustering coefficient across graph sizes provides quantitative evidence that larger datasets enable more reliable crime pattern detection. Particularly, the high clustering coefficient (0.62) observed in the 1000-node network suggests that crime reports naturally organize into cohesive communities, providing strong empirical validation for our graph-based analytical approach.

Table 3. Crime graph datasets.

Number of nodes	1000	500	100
Number of edges	83,226	19,494	545
Density	0.167	0.15	0.11
Clustering coefficient	0.62	0.58	0.46



(**a**) (**b**) (**c**)

Figure 6. The crime graphs that were constructed for each dataset: (**a**) 1000 reports, (**b**) 500 reports, and (**c**) 100 reports.

5.2. Clustering the Crime Reports

The performance of the resulting clusters is evaluated using different metrics, which are the Silhouette Score, Davies–Bouldin Index, and Calinski–Harabasz Index. These metrics assess the compactness and separation of the clusters, providing insights into the effectiveness of the clustering process.

- The Silhouette Score (S) is a widely used metric to validate the performance of clustering algorithms. It measures the quality of a clustering result by considering both the compactness of each cluster and the separation between different clusters. Whereas compactness indicates how closely related the objects in a cluster are to one another, separation indicates how distinct or well-separated a cluster is from other clusters. The score ranges from −1 to 1, where a higher value indicates a better clustering solution [25].

$$S = \frac{1}{NC}\sum_i \left\{ \frac{1}{ni}\sum_{x\in Ci} \frac{b(x)-a(x)}{max[b(x),a(x)]} \right\} \tag{3}$$

- The Davies–Bouldin Index (DB) is an internal evaluation metric that assesses the quality of a clustering result by calculating, for each cluster C, the similarity between C and all other clusters. The highest similarity value is assigned to C as its cluster similarity. Then, DB is obtained by averaging all the cluster similarities. A lower DB value indicates a better clustering result, as it suggests that the clusters are compact and well-separated from each other [26].

$$\mathrm{DB} = \frac{1}{\mathrm{NC}}\sum_i \mathrm{maxj,j} \neq \mathrm{i}\{[\frac{1}{\mathrm{ni}}\sum_{x\in \mathrm{Ci}} \mathrm{d(x,ci)} + \frac{1}{\mathrm{nj}}\sum_{x\in \mathrm{Cj}} \mathrm{d(x,cj)}]/\mathrm{d(ci,cj)}\} \tag{4}$$

- The Calinski–Harabasz index (CHI) evaluates clustering quality by measuring both cluster compactness and separation. It is calculated as

$$CH = \frac{BCSM}{(k-1)} \times \frac{(n-k)}{WCSM} \tag{5}$$

where n represents the total number of data points, k is the number of clusters, BCSM (Between Cluster Scatter Matrix) measures inter-cluster separation, and WCSM (Within Cluster Scatter Matrix) measures intra-cluster compactness. Higher CHI values indicate that clusters are both internally compact and well-separated from each other, similar to tight bundles with clear gaps between them [26].

To determine the optimal number of clusters, we conducted a systematic analysis using the Silhouette Score across different clustering configurations (k = 2, 3, 5, and 7). As demonstrated in Figure 7, Silhouette Scores show a clear pattern: starting at 0.73 for k = 2, increasing to 0.77 for k = 3, reaching a peak of 0.78 for k = 5, and then declining to 0.73 for k = 7. While k = 5 achieved marginally higher performance (0.78), k = 3 emerges as the optimal choice with a strong Silhouette Score of 0.77 when considering both clustering quality and computational efficiency. Given that computational complexity scales with the number of clusters and the minimal improvement (0.01) between k = 3 and k = 5, we selected k = 3 as the optimal configuration for our crime report clustering task.

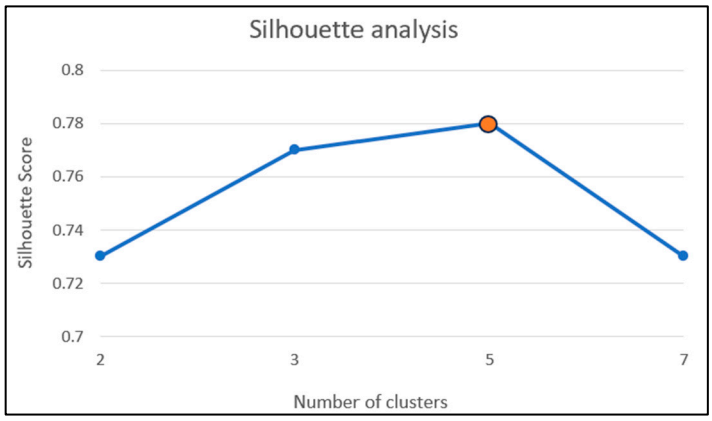


Figure 7. Silhouette analysis to find the optimal cluster value.

The performance measurements, Silhouette Score (S), Davies–Bouldin Index (DB), and Calinski–Harabasz Index (CHI), were calculated for the proposed unified solution that integrates spectral graph-based clustering with GCN graph learning. Moreover, the results of the proposed solution were compared to different clustering algorithms, which include K-means clustering and spectral graph-based clustering. The performance evaluation was conducted on crime report datasets for the following scenarios:

- K-means clustering, as a baseline clustering algorithm.
- K-means clustering with GCN learning, as a unified solution of a clustering algorithm and graph learning.
- K-means clustering with GCN learning with the support of linguistic features.
- Spectral graph-based clustering, as graph-based clustering.
- Spectral graph-based clustering with GCN learning is the proposed unified solution of graph-based clustering with graph learning.
- Spectral graph-based clustering with GCN learning with the support of linguistic features.

For the first performance measure, Figure 8a–c shows the Silhouette Score for different clustering methods across varying dataset sizes. The K-means clustering achieves the highest Silhouette Score of around 0.65 on the smallest dataset of 100 reports. This suggests that K-means performs well in creating compact and well-separated clusters when the dataset size is relatively small. However, spectral clustering alone performs poorly, with negative Silhouette Scores, indicating overlapping or poorly defined clusters.

As shown in Figure 8c, when the dataset size increases to 1000 reports, the idea of integrating the clustering algorithm with graph learning achieves the highest Silhouette Score of approximately 0.83 of the unified solution of K-means clustering with GCN learning. In addition, the proposed unified solution of spectral clustering and GCN (Spectral + GCN) achieves a Silhouette Score of approximately 0.77. This improvement in performance can be attributed to the ability of GCN to capture the underlying graph structure and node relationships, enabling more accurate clustering. The integration of graph-based information enhances the separation between clusters and the overall clustering quality.

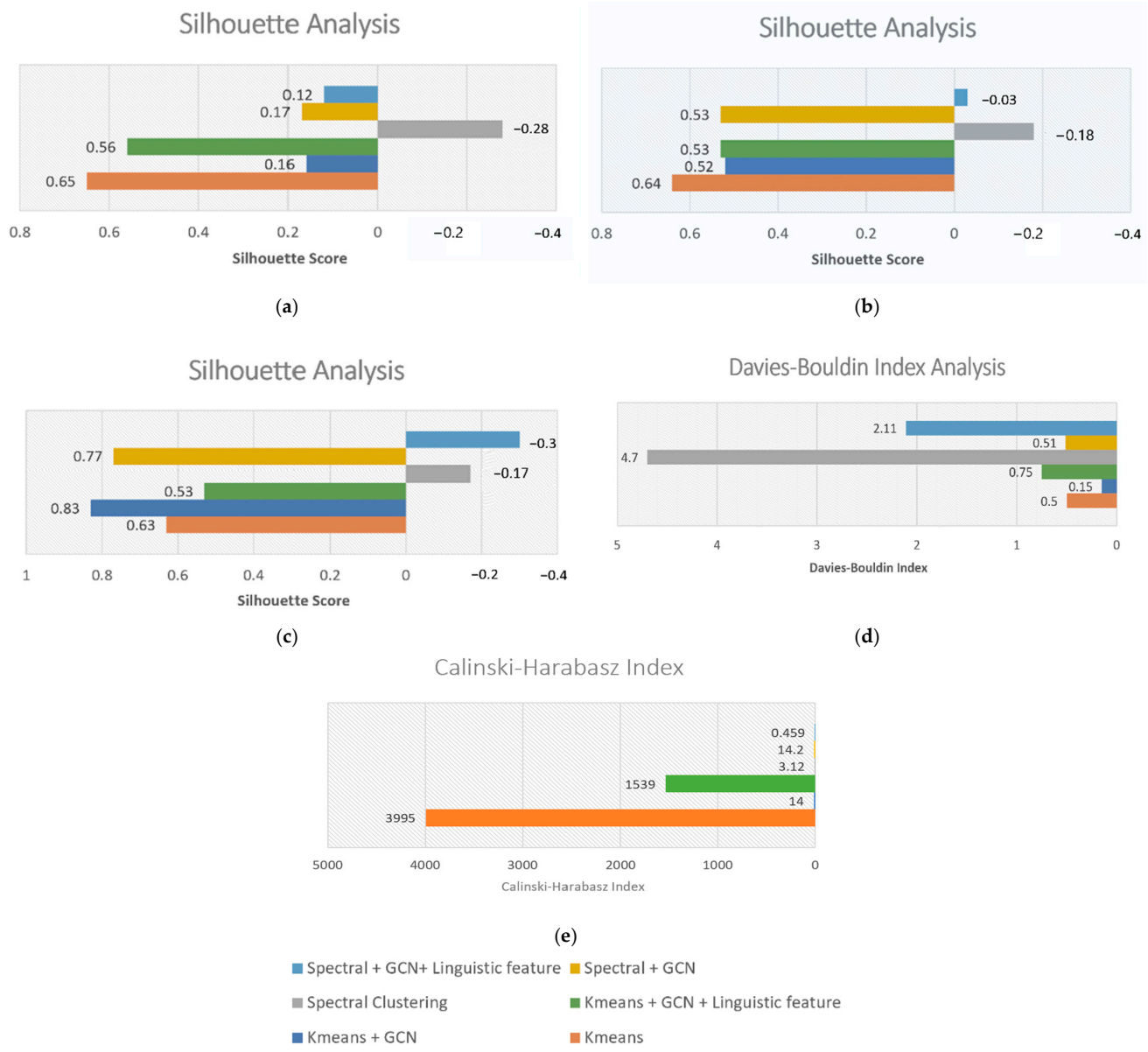


Figure 8. Performance evaluation of the proposed system: (**a**) Silhouette analysis for 100 reports, (**b**) Silhouette analysis for 500 reports, (**c**) Silhouette analysis for 1000 reports, (**d**) Davies–Bouldin Index analysis for 1000 reports, (**e**) Calinski–Harabasz Index for 1000 reports.

Furthermore, it should be mentioned that the addition of linguistic features revealed unexpected impacts on clustering performance across different experimental configurations. For the 1000-node graph, K-means clustering showed significant performance degradation when linguistic features were added, with the Silhouette Score decreasing from 0.75 to 0.53 and the Davies–Bouldin Index increasing from 0.15 to 0.75. This degradation was even more pronounced in spectral clustering, where the addition of linguistic features led to a sharp decline in performance: the Silhouette Score dropped from 0.70 to −0.30, while the Davies–Bouldin Index increased from 0.51 to 2.11. This consistent pattern of degradation can be attributed to increased feature space dimensionality, which exacerbates the curse of dimensionality, and the introduction of noise through non-relevant entity recognition. These findings suggest that while linguistic features potentially offer additional information, their current implementation may require more sophisticated integration methods to enhance rather than impair clustering performance.

Regarding the size of the dataset, we can say that the dataset of size 1000 is likely more representative than the smaller ones, as it exhibits consistent patterns, a stabilization of results, and provides a larger sample size.

Moving to the second measure, the Davies–Bouldin Index was calculated for different clustering methods. Lower DB values indicate better clustering performance. The DB is calculated by averaging all the cluster similarities, so the smaller index will achieve the best partition. As shown in Figure 8d, it is obvious that integrating K-means and spectral clustering with GCN (Spectral + GCN) achieves the lowest Davies–Bouldin Index with 0.15 and 0.51, respectively, showing that the integration of clustering algorithms with graph learning indicates that the clusters are distinct from each other and achieves the best partition.

The Calinski–Harabasz Index (CHI) results, as shown in Figure 8e, reveal significant variations in clustering performance across different methods. K-means alone shows the highest CHI value at 3995, indicating strong cluster separation in the base algorithm. The integration of GCN and linguistic features with K-means resulted in a substantially lower CHI of 1539, suggesting that these additional features may have reduced the distinctiveness between clusters. Spectral clustering demonstrated moderate performance with a CHI of 14.2, while K-means + GCN achieved a CHI of 3.12. Notably, spectral + GCN and spectral + GCN with linguistic features showed relatively lower CHI values. These results present an interesting contrast to other evaluation metrics (like the Davies–Bouldin Index), suggesting that while some methods may create well-separated clusters (high CHI), they might not necessarily provide the most optimal overall clustering solution when considered alongside other performance metrics.

To sum up, there are different findings that can be listed as follows:

- The use of a unified approach of the spectral graph-based clustering with GCN graph learning can enhance performance in clustering crime reports in terms of cluster separation and compactness. Also, using a unified approach of K-means clustering and GCN graph learning can enhance performance.
- This suggests that the combination of graph learning with clustering algorithms can effectively capture the underlying structure and similarities among crime reports. The GCN's ability to learn node embeddings by considering the graph structure and node features likely contributed to the improved clustering quality.
- The adding of linguistic features did not consistently enhance the clustering results across all methods. This suggests that the linguistic features, in their current form, may not always provide additional discriminative information for clustering crime reports and may even introduce noise or inconsistencies. This finding highlights the need for further investigation into the selection and representation of linguistic features for crime report clustering.

5.3. Computational Complexity Analysis

The algorithm's computational performance is primarily bounded by its spectral clustering component, which has a time complexity of $O(N^3)$ due to eigen decomposition calculations, making it most suitable for medium-sized crime report collections ($N < 10^4$). The space complexity is $O(N^2)$, driven by the storage requirements of the adjacency matrix. These complexities are influenced by several key parameters: N (number of nodes), E (number of edges), F (feature dimension), k (number of clusters), and the number of training epochs. Three main performance bottlenecks affect the algorithm's scalability: the cubic computational cost of spectral clustering, the quadratic memory requirements for storing the adjacency matrix, and the computational overhead from multiple graph traversals during GCN training. For larger datasets, these limitations could be addressed through optimization techniques such as sparse matrix representations, approximate spectral clustering methods, or batch processing implementations.

5.4. Effectiveness in Crime Analysis

Our graph-based clustering algorithm addresses key challenges in crime analysis through several mechanisms:

- First, our proposed graph-based clustering algorithm effectively groups similar crime news documents by type, helping analysts identify emerging trends. Law enforcement agencies frequently need to search historical cases for precedent review [27]; our system facilitates this by organizing and categorizing crimes into distinct clusters for further analysis and statistical evaluation.
- To perform a graph-based clustering algorithm, the crime-related news reports are collected from established U.S. news organizations, and we implement a comprehensive preprocessing pipeline including sentence segmentation, word tokenization, stop-word removal, and lemmatization. In order to make the required crime analysis and cluster the documents that have similar crime words, the transformers and Deep Averaging Network (DAN) encoders convert documents into 512-dimensional vectors, so the documents from D1 to D1000 are encoded to 1000 vectors. These vectors can be used in the next step to find semantic similarity between the crime reports. To find the relationships between different crime reports, the graph-based representation facilitates an enhanced understanding of the interconnections and similarities between crime reports, where the nodes of the graphs are the vectors, and the graph's edges represent the relationships or similarities between the crime texts. To validate our approach's practical utility in crime analysis, we evaluate performance using three quantitative metrics: Silhouette Score (S), Davies–Bouldin Index (DB), and Calinski–Harabasz Index (CHI), demonstrating pattern detection accuracy.

The effectiveness of our approach is demonstrated through clustering results shown in Table 4, where similar crime reports are successfully grouped into relevant categories (e.g., murders, drug-related offenses, robbery, and theft) despite varying narrative styles and contexts.

Table 4. Example of cluster results.

Report	Preprocessed	Cluster
Portland Hate Killer Ranted About Stabbings and Muslims on Facebook Amid Rising U.S. Hate Crime. Jeremy Joseph Christian faces a judge Tuesday in connection with the killings.	Portland hate killer rant stabbings Muslim Facebook rise hate joseph face judge connection killing	4 (Murders)
Designer Carolina Herrera's Nephew Killed in Kidnapping. Herrera blasts "carnage and murders" in her native Venezuela.	designer Carolina Herrera kill blast carnage murder venezuela	4 (Murders)
LA Political Donor Gets 30 Years in Prison for Fetish Deaths. A federal judge has sentenced a wealthy California Democratic donor to 30 years in prison for injecting two men with lethal doses of drugs.	donor get year prison judge sentence California donor year prison inject men dos drug	0 (Drug-Related Offenses)
Florida Officer Accused of Selling Drugs to Undercover Detective. Authorities say Dwayne Frazier made the deal out of his marked patrol car while in uniform.	officer accuse sell drug say Dwayne make deal mark patrol car uniform	0 (Drug-Related Offenses)
Bungling Robbery Suspects Receive Much-Deserved Dose of Karma. A gusty Mother Nature spoiled the getaway.	bungle robbery suspect receive gusty mother nature spoil getaway	2 (Robbery and Theft)
Bungling Bicycle-Riding Robbery Suspect Is Foiled By Wet Weather. Oops!	bungle robbery suspect foil wet	2 (Robbery and Theft)

6. Conclusion and Future Work

In conclusion, in the context of crime clustering, the use of graph-based clustering has gained significant attention due to its impressive performance. However, when applied to real-world data, the presence of noise and outliers, along with a sparse data structure, can pose challenges for clustering tasks. Therefore, there is a need to enhance the performance of graph-based clustering in document clustering. This research proposes a unified solution that integrates graph-based clustering with graph learning for clustering crime reports. This approach considers spectral clustering and Graph Convolutional Networks (GCN)

for graph learning. The research also provides different recommendations that can be considered as future works, which include:

- It is recommended to explore the field by integrating different graph-based clustering with different graph learning algorithms and assessing their performance in order to find the best clustering quality.
- Regarding graph construction, we could investigate alternative methods instead of the ε-ball graph method and examine their effects on the proposed system.
- For the vectorization process, we could explore different embedding methods to convert textual data into high-dimensional vectors rather than use a Universal Sentence Encoder (USE) and explore the effects on the proposed system.
- The proposed unified approach can be tested with a larger dataset that contains more than 1000 crime reports.
- The study also raises questions about the effectiveness of linguistic features and their integration with clustering methods, prompting further research in this area.

Author Contributions: The contribution of H.A.-I. is designing the model, programming, and writing—original draft preparation. H.K. has contributed to supervision, validation, and paper editing tasks. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by a grant from the Researchers Supporting Unit, Project number (RSP-2024/204), King Saud University, Riyadh, Saudi Arabia.

Data Availability Statement: https://www.kaggle.com/datasets/rmisra/news-category-dataset (accessed on 15 May 2024).

Acknowledgments: This research was supported by a grant from the Researchers Supporting Unit, Project number (RSP-2024/204), King Saud University, Riyadh, Saudi Arabia.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Dharsan, R.; Krishanthini, M.; Traveena, C.; Anubama, L.; Hansika, M.M.; Chandrasiri, S.S. The Future of Crime Prevention: Police Case Analysis Using Machine Learning. In Proceedings of the 2023 5th International Conference on Advancements in Computing (ICAC), Colombo, Sri Lanka, 7–8 December 2023; pp. 454–459.
2. Pramanik, A.; Das, A.K.; Ding, W. Graph based fuzzy clustering algorithm for crime report labelling. *Appl. Soft Comput.* **2023**, *141*, 110261. [CrossRef]
3. Pickens, A.; Sengupta, S. Benchmarking Studies Aimed at Clustering and Classification Tasks Using K-Means, Fuzzy C-Means and Evolutionary Neural Networks. *Mach. Learn. Knowl. Extr.* **2021**, *3*, 695–719. [CrossRef]
4. Chaudhry, M.; Shafi, I.; Mahnoor, M.; Vargas, D.L.R.; Thompson, E.B.; Ashraf, I. A Systematic Literature Review on Identifying Patterns Using Unsupervised Clustering Algorithms: A Data Mining Perspective. *Symmetry* **2023**, *15*, 1679. [CrossRef]
5. Pramanik, A.; Das, A.K.; Pelusi, D.; Nayak, J. An Effective Fuzzy Clustering of Crime Reports Embedded by a Universal Sentence Encoder Model. *Mathematics* **2023**, *11*, 611. [CrossRef]
6. Mirkes, E.M.; Allohibi, J.; Gorban, A. Fractional Norms and Quasinorms Do Not Help to Overcome the Curse of Dimensionality. *Entropy* **2020**, *22*, 1105. [CrossRef] [PubMed]
7. Zhang, X.; Xie, X.; Kang, Z. Graph Learning for Attributed Graph Clustering. *Mathematics* **2022**, *10*, 4834. [CrossRef]
8. Li, Z.; Huang, C.; Xia, L.; Xu, Y.; Pei, J. Spatial-temporal hypergraph self-supervised learning for crime prediction. In Proceedings of the the 2022 IEEE 38th International Conference on Data Engineering (ICDE), Kuala Lumpur, Malaysia, 9–12 May 2022; pp. 2984–2996.
9. Mandalapu, V.; Elluri, L.; Vyas, P.; Roy, N. Crime Prediction Using Machine Learning and Deep Learning: A Systematic Review and Future Directions. *IEEE Access* **2023**, *11*, 60153–60170. [CrossRef]
10. Liu, Z.; Barahona, M. Graph-based data clustering via multiscale community detection. *Appl. Netw. Sci.* **2020**, *5*, 3. [CrossRef]
11. Bianchi, F.M.; Grattarola, D.; Alippi, C. Spectral clustering with graph neural networks for graph pooling. In Proceedings of the International Conference on Machine Learning, Vancouver, BC, Canada, 13–19 July 2020.
12. Haifa, A.; Heba, K. Towards Designing a Graph-Based Clustring of Crime Reports using Graph Learning Techniques. In Proceedings of the Future Technologies Conference (FTC) 2024; Springer Nature: Berlin/Heidelberg, Germany, 2024.
13. Mishra, R.K.; Raj, H.; Urolagin, S.; Jothi, J.A.; Nawaz, N. Cluster-Based Knowledge Graph and Entity-Relation Representation on Tourism Economical Sentiments. *Appl. Sci.* **2022**, *12*, 8105. [CrossRef]
14. Das, P.; Das, A.K.; Nayak, J.; Pelusi, D.; Ding, W. A Graph Based Clustering Approach for Relation Extraction from Crime Data. *IEEE Access* **2019**, *7*, 101269–101282. [CrossRef]

15. Pm, D.; Pb, R.; Cletus, N.; Joy, P. Fuzzy Hypergraph Modeling, Analysis and Prediction of Crimes. *Int. J. Comput. Digit. Syst.* **2022**, *11*, 648–661. [CrossRef]
16. Wang, P.; Yang, X.; Ding, W.; Zhan, J.; Yao, Y. Three-way clustering: Foundations, survey and challenges. *Appl. Soft Comput.* **2024**, *151*, 111131. [CrossRef]
17. Li, M.; Lu, S.; Zhang, L.; Zhang, Y.; Zhang, B. Community Detection Method for Social Network Based on Community Embedding. *IEEE Trans. Comput. Soc. Syst.* **2021**, *8*, 308–318. [CrossRef]
18. Cer, D.; Yang, Y.; Kong, S.; Hua, N.; Limtiaco, N.; John, R.S.; Constant, N.; Guajardo-Cespedes, M.; Yuan, S.; Tar, C. Universal sentence encoder. *arXiv* **2018**, arXiv:180311175.
19. Chiche, A.; Yitagesu, B. Part of speech tagging: A systematic review of deep learning and machine learning approaches. *J. Big Data* **2022**, *9*, 10. [CrossRef]
20. Ahmedt-Aristizabal, D.; Armin, M.A.; Denman, S.; Fookes, C.; Petersson, L. Graph-Based Deep Learning for Medical Diagnosis and Analysis: Past, Present and Future. *Sensors* **2021**, *21*, 4758. [CrossRef] [PubMed]
21. News Category Dataset. Available online: https://www.kaggle.com/datasets/rmisra/news-category-dataset (accessed on 24 May 2024).
22. Vaswani, A.; Shazeer, N.M.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. In Proceedings of the Neural Information Processing Systems. 2017. Available online: https://www.semanticscholar.org/paper/204e3073870fae3d05bcbc2f6a8e263d9b72e776 (accessed on 27 November 2024).
23. Malik, F.; Khan, S.; Rizwan, A.; Atteia, G.; Samee, N.A. A Novel Hybrid Clustering Approach Based on Black Hole Algorithm for Document Clustering. *IEEE Access* **2022**, *10*, 97310–97326. [CrossRef]
24. Needham, M.; Hodler, A.E. *Graph Algorithms: Practical Examples in Apache Spark and Neo4j*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2019; ISBN 978-1-4920-4765-0.
25. Xu, M. Understanding Graph Embedding Methods and Their Applications. *SIAM Rev.* **2021**, *63*, 825–853. [CrossRef]
26. Rachwał, A.; Popławska, E.; Gorgol, I.; Cieplak, T.; Pliszczuk, D.; Skowron, Ł.; Rymarczyk, T. Determining the Quality of a Dataset in Clustering Terms. *Appl. Sci.* **2023**, *13*, 2942. [CrossRef]
27. Raaijmakers, S. Artificial Intelligence for Law Enforcement: Challenges and Opportunities. *IEEE Secur. Priv.* **2019**, *17*, 74–77. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.