Journal of COMPUTER AND FORENSIC SCIENCES

CFS 2024, Vol. 3, Issue 2, pp. 28–42
https://doi.org/10.5937/jcfs3-55164
Original research paper

Comparative Analysis of Certain Clustering Algorithms That Do Not Require a Predefined Number of Clusters on the Articles of the Criminal Code of the Republic of Serbia

Sanja Raičević[1] and Vojkan Nikolić[2*]

[1] Ministry of Interior of the Republic of Serbia; sanjaraicevic09@gmail.com

[2] University of Criminal Investigation and Police Studies, Belgrade, Serbia; vojkan.nikolic@kpu.edu.rs

* Corresponding author: vojkan.nikolic@kpu.edu.rs

Received: December 3, 2024 • Accepted: December 24, 2024 • Published: December 30, 2024

Abstract: This paper explores the application of certain clustering algorithms for analyzing textual documents from the Criminal Code (CC) of the Republic of Serbia (RS). Clustering was performed using three popular algorithms: DBSCAN, mean-shift, and hierarchical clustering (agglomerative clustering). The input data consisted of textual documents in .txt format, where each document corresponds to a specific article of the law. The aim of this study is to identify thematic groups within the legislative texts and analyze the advantages and disadvantages of each algorithm in the context of the specific characteristics of legal documents. The clustering results using these three algorithms show that DBSCAN faces challenges with noise, while mean-shift effectively detects dense clusters, and hierarchical clustering allows for a detailed analysis at various levels of granularity. In conclusion, this paper provides valuable insights into the application of these clustering algorithms to legal texts and offers recommendations for their selection when analyzing similar datasets.

Keywords: clustering; DBSCAN; mean-shift; hierarchical clustering; legal documents.

1. INTRODUCTION

Clustering is one of the techniques in data analysis, particularly in the field of natural language processing (NLP) and text analysis. In the context of legal documents, clustering is used to identify similarities among different texts, which can contribute to better organization and understanding of legislative provisions. This paper focuses on the application of clustering algorithms to textual documents from the Criminal Code (CC), with the aim of exploring the efficiency of three clustering algorithms: DBSCAN (density-based spatial clustering of applications with noise), mean-shift, and hierarchical clustering (agglomerative clustering).

Legislative texts, such as articles of law, are characterized by specific language and terminology, posing a challenge for clustering algorithms, which must recognize thematic groups within the text while filtering out noise and irregularities in the data. Each of the

aforementioned algorithms has its advantages and limitations depending on the nature of the data to which they are applied. DBSCAN is known for its ability to identify non-spherical clusters and noise, mean-shift automatically detects dense clusters without the need for a pre-defined number of clusters, while hierarchical clustering allows for analysis at various levels of granularity.

Through the application of these algorithms to over 400 textual documents, where each corresponds to a specific article of law, the aim is to identify thematic groups in the legislative text by analyzing the clustering results and providing recommendations for the selection of the best algorithm for similar textual datasets. Furthermore, the research offers valuable insights into how the specifics of legal language affect the performance of clustering algorithms, as well as how different clustering methods can contribute to the analysis and understanding of legislative documents.

2. RELATED WORK

Recent years have seen clustering techniques become a key tool in the analysis of large datasets, particularly in areas such as big data analysis, image recognition, text processing, and natural language processing. Many studies have focused on comparing and optimizing different clustering methods to improve their efficiency in various applications. This section provides an overview of existing research relevant to the clustering algorithms discussed in this paper, with a focus on their applications, strengths, and weaknesses as presented in previous studies.

One of the key areas of research in clustering involves comparing different methods, such as K-means, DBSCAN, mean-shift, and hierarchical clustering. In their work, Smith and Brown [1] discuss the application of clustering techniques in the context of big data analysis, emphasizing the importance of selecting the right algorithm based on the data structure. Their research highlights the need for scalable solutions in large datasets, a theme echoed in the work of Silva and Johnson [2], who analyze fixed and variable techniques for choosing the number of clusters in K-means and K-medoids clustering. They emphasize the significance of cluster initialization and the selection of the optimal number of clusters, which remains a central challenge in clustering research.

Lopez and Wang [3] explore density-based methodologies and hierarchical methods, providing a detailed insight into the advantages of these approaches when the number of clusters is not pre-defined. This work is particularly relevant for DBSCAN, a density-based clustering method that automatically adjusts the number of clusters based on the data distribution. Their findings align with the analysis by Gonzalez and Kim [4], who consider the use of DBSCAN for clustering noisy data, demonstrating its resilience to outliers and effectiveness in conditions of high noise levels.

Mitchell and Jang [5] conduct a comparative analysis of DBSCAN, mean-shift, and hierarchical clustering. They conclude that DBSCAN is efficient for data with varying densities, while mean-shift clustering excels in applications such as image segmentation, where well-defined modes are present. This comparative analysis is extended in the work of Patterson and Chen [6], who explore the application of mean-shift clustering in image and

object recognition. Their work is significant because it demonstrates how mean-shift can be adapted to different domains, not just for textual and numerical data.

Chan and Gomez [7] provide a comprehensive review of hierarchical clustering methods, especially in smaller datasets, where the complexity of other algorithms may not be justified. This research builds upon the findings of Liu and Zhang [8], who investigate Python libraries designed for clustering and visualizing textual data, showing the usefulness of tools such as SciPy and Scikit-learn in real-world clustering tasks. They offer practical insights into implementing these algorithms using Python, making them accessible to a wider research and professional audience.

Clustering of textual data has also received significant attention. Chen and Fischer [9] focus on advanced libraries for clustering and visualizing textual data, while Molina and Albrecth [10] examine the application of clustering algorithms in text data analysis. Zhang and Li [11] extend this topic by addressing the specific challenges of clustering textual data in natural language processing, a field where clustering techniques are widely used to group similar documents or extract topics from large collections of texts. These studies provide a solid foundation for the application of clustering techniques in text data analysis and NLP tasks, further confirming the versatility of clustering algorithms in different domains.

Jovanović and Petrov [12] compare DBSCAN, mean-shift, and hierarchical clustering in the context of legal text analysis, highlighting the strengths and weaknesses of each of these methods in this specific application. This work is particularly relevant to the research presented in this paper, as it bridges theoretical clustering methods with their practical applications in specialized fields such as law and legal document analysis.

Nikolić and colleagues [13, 14] introduce advanced systems for quick response in e-government in the Republic of Serbia, demonstrating how clustering models can contribute to optimizing the processing of legal and criminal data. Their results are particularly significant for the implementation of models based on hierarchical methods and the DBSCAN algorithm. Jovanović and Petrov [12] analyze the effectiveness of these methods in legal texts, highlighting the practical challenges and potential advantages in specialized domains.

In conclusion, the related works demonstrate the wide application of clustering techniques across various domains, from big data analysis to image recognition and text data processing. The reviewed studies provide a comprehensive understanding of the strengths and weaknesses of each algorithm, as well as the challenges in optimizing clustering methods for specific applications. This paper builds upon the foundations established in previous research, applying and evaluating clustering techniques such as DBSCAN, mean-shift, and hierarchical clustering in the context of text data analysis, further exploring their potential in specific applications.

3. CLUSTERING

Clustering is an important technique in data analysis that enables the grouping of similar objects into clusters, facilitating the identification of patterns and structures within the data. This method has wide applications in various disciplines, including statistics, ma-

chine learning, and data analysis, aiming to extract valuable insights from unstructured information.

Clustering is a key technique for data analysis, and understanding the differences between methods that require a predefined number of clusters and those that do not can significantly enhance a researcher's ability to extract useful information from different datasets. The decision on which clustering method to use depends on the specific characteristics of the dataset, the objectives of the analysis, and the available resources and time for processing.

Different clustering methods bring their advantages and disadvantages, and thus it is important to analyze which methods will yield the best results based on the nature of the data and the research objectives in each particular case [1, 13].

3.1. Clustering with a Predefined Number of Clusters

One of the most well-known and widely used approaches in clustering is the K-means method, which requires a predefined number of clusters. In this method, the user must specify the number of clusters, denoted by K. The algorithm then randomly selects K cluster centers and assigns data points to the nearest center. It then iteratively updates the centers until convergence is reached.

The advantage of K-means lies in its speed and simplicity. However, the choice of the number of clusters can significantly affect the results. For example, a large K can lead to the fragmentation of the data into too many clusters, while a small K may result in the loss of information and important patterns. To determine the optimal K, researchers often use the Elbow method, which analyzes the change in within-cluster variance with respect to different values of K. However, this is not always precise, especially when the data is complex or highly dimensional.

In addition to K-means, other methods that require a predefined number of clusters are mentioned in clustering, such as the K-medoids method, which uses a similar approach but instead of cluster centers, uses actual data points (medoids). This method can be more robust to noise and outliers [2].

3.2. Clustering Without a Predefined Number of Clusters

In addition to clustering with a predefined number of clusters, there are clustering techniques that do not require the number of clusters to be specified in advance. These methods, such as DBSCAN (density-based spatial clustering of applications with noise) and hierarchical clustering, offer greater flexibility and can better handle complex data.

The DBSCAN method identifies clusters based on data density, where clusters represent areas with high point density, while sparse points are classified as noise. This method is particularly useful when clusters have different shapes and sizes, which is often the case in real-world scenarios. Furthermore, DBSCAN does not require prior definition of the number of clusters, making it suitable for analyzing data with unknown structures.

Hierarchical clustering is a method that builds a dendrogram, a visual representation showing how clusters are formed based on similarity. This method can be agglomerative (merging clusters) or divisive (splitting clusters), allowing researchers to analyze data at multiple levels. This is useful when wanting to understand the hierarchical structure of data or when exploring different aspects of clustering [3].

4. CLUSTERING ALGORITHMS: HIERARCHICAL, DBSCAN, AND MEAN-SHIFT

Clustering is one of the key techniques in data analysis, and different algorithms offer various approaches for grouping data. Three significant clustering algorithms are:

1) Hierarchical;

2) DBSCAN;

3) Mean-shift.

Each of these algorithms has its own characteristics, applications, advantages, and drawbacks, and understanding these differences can help in selecting the most appropriate method for solving a particular problem.

Hierarchical, DBSCAN, and mean-shift are powerful tools for clustering, each with its own specific advantages and challenges. The hierarchical algorithm is extremely useful for understanding relationships between clusters but can be computationally expensive. DBSCAN provides resilience to noise and can detect clusters of any shape, but it depends on the correct selection of parameters. Mean-shift allows flexible modeling of dense regions, but it is sensitive to the choice of window size.

The selection of the appropriate algorithm depends on the nature of the data and the specific goals of the analysis, which makes clustering a dynamic and challenging field of research in data analysis [5].

4.1. Hierarchical Algorithm

Hierarchical clustering is based on the formation of a dendrogram, which graphically displays the structure of clusters. This algorithm can be agglomerative or divisive. The agglomerative approach starts with each data point as a separate cluster and gradually merges the most similar clusters until only one cluster remains. In contrast, the divisive approach begins with a single cluster and divides it into smaller clusters.

Hierarchical clustering is commonly used in biology (for species classification), sociology, and text analysis. Its visualization, the dendrogram, helps researchers better understand the relationships between clusters. Hierarchical algorithms are most effective when working with small and medium-sized datasets, especially when there is a need to interpret relationships among clusters.

When considering the potential for error, one of the disadvantages of hierarchical clustering is its sensitivity to the choice of similarity metric and the presence of noise. Additionally, when the data is very large, this algorithm can become computationally expensive [7].

4.2. DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

DBSCAN (density-based spatial clustering of applications with noise) is an algorithm that identifies clusters based on data density. It defines a cluster as a group of points that are close to each other while increasing the distance between points that are sparse. The key parameters are epsilon (eps), which defines the maximum distance between points considered to belong to the same cluster, and the minimum number of points required to form a cluster.

DBSCAN is often used in geographic information systems, image analysis, and situations where there is noise in the data. This approach allows for efficient identification of clusters of any shape, making it suitable for complex datasets.

However, although DBSCAN is resistant to noise, it may struggle with cluster detection when the data is unevenly distributed or when clusters have different densities. The choice of the eps value is crucial and can significantly affect the results.

DBSCAN is most effective when working with data that has clear density patterns and when there is a significant amount of noise [4].

4.3. Mean-Shift

Mean-shift is a non-parametric algorithm that searches for density regions in the data space. The algorithm works by finding the "center of mass," or the center of the cluster, based on the density of points in the immediate vicinity. It uses a fixed-size window (bandwidth) that moves through the data space, and the centroids are updated until stability is achieved.

Mean-shift is applied in image analysis, object recognition, and situations where good cluster detection is required without predefined cluster numbers. This algorithm is popular in computer vision and data analysis and is one of the most suitable for data with clearly defined density regions, especially in image analysis and geometric processing.

The main challenge of the mean-shift algorithm is choosing the appropriate window size. A window that is too small can lead to excessive fragmentation of clusters, while one that is too large may result in a loss of detail. Additionally, the algorithm can be slow when applied to large datasets [6].

5. OVERVIEW OF LIBRARIES USED FOR CLUSTERING AND VISUALIZATION OF TEXTUAL DATA

The analysis of large textual data in modern science increasingly involves the use of sophisticated libraries for clustering and data visualization, enabling researchers to precisely identify patterns in the data, even when dealing with complex, high-dimensional datasets. This paper focuses on the essential libraries that facilitate efficient management of text

files, feature extraction, clustering, and dimensionality reduction to allow for a better understanding of structures within textual data.

Through the systematic application of specific libraries, scientific research in the field of textual data analysis achieves greater precision and insight into the internal structures of large textual datasets. By combining different clustering methods and visualization tools, researchers can analyze data from various perspectives and generate results that provide a deeper understanding of the data [9].

5.1. Working with File Systems

The "os" library in Python is a fundamental tool for working with file systems and allows for manipulation of files and directories. In scientific research, this library is useful for automatically loading and processing a large number of files from specific directories, simplifying the data preparation process for analysis.

5.2. Efficient Numerical Data Processing

"Numpy" is a key library for numerical processing that enables efficient manipulation and computation with large arrays and matrices. Its functionalities, such as handling high-dimensional data, simple matrix algebra functions, and statistical operations, make it an indispensable tool in scientific research, especially in combination with machine learning algorithms that require working with large datasets.

5.3. Cluster Visualization

For visualizing clusters in low-dimensional representations, "matplotlib.pyplot" is the primary library for generating plots in Python. It allows for the easy creation and customization of diagrams, scatter plots, histograms, and other visual representations that help researchers gain insight into the data structure and present the research results.

5.4. Feature Extraction from Text

The "TfidfVectorizer" library from the "sklearn.feature_extraction.text" module enables the conversion of unstructured textual data into numerical vectors suitable for clustering and other analyses. Tfidf vectorization (term frequency-inverse document frequency) assigns more weight to words that are unique to individual documents, which helps extract significant features from text and enables clustering algorithms to better recognize patterns.

5.5. Density-based Clustering

DBSCAN (density-based spatial clustering of applications with noise) is a clustering algorithm that recognizes clusters based on data density. This algorithm does not require prior determination of the number of clusters but instead uses parameters such as "eps" (radius for connecting points) and "min_samples" (minimum number of points per cluster), allowing for the identification of clusters of any shape and separating points that are not part of a cluster as noise. DBSCAN is particularly effective for analyzing unstructured textual data when clusters are of varying shapes and sizes.

5.6. Mean-Shift Algorithm

Mean-shift is a clustering algorithm based on shifting the centroid in the direction of data density until it reaches the density maximum, allowing for the formation of a natural number of clusters. The "estimate_bandwidth" function automatically determines the bandwidth for each cluster based on the data, which further contributes to the algorithm's automatic adjustment. Mean-shift is useful for clustering complex datasets when the number of clusters is not known in advance [14].

5.7. Hierarchical Clustering

Hierarchical clustering, which enables an agglomerative approach to data grouping, is used to recognize structures in data when clusters are hierarchically organized. In the "AgglomerativeClustering" module, this algorithm applies cluster merging using metrics such as ward, complete, and average, providing flexibility in choosing the merging strategy. Visualizing this process through dendrograms, available through "scipy.cluster.hierarchy," further allows researchers to adjust cluster boundaries and gain a better understanding of the data's complexity.

5.8. High-dimensional Data Visualization

The t-SNE (t-distributed stochastic neighbor embedding) algorithm is used for dimensionality reduction of high-dimensional data for visualization. The t-SNE transforms the data into a two-dimensional or three-dimensional space while preserving the local cluster structures, enabling easier recognition of patterns and identification of inter-cluster relationships in textual and other datasets [8].

6. DATASET

The data analyzed in this example are text files located in a directory defined by the variable directory (in this case, "C:/zakon_clanovi"). These files may contain articles, laws,

comments, or any other form of unstructured text. The goal is to analyze similarities among them based on their content.

Each text file represents an article of law, where each document is titled "Član X" (the article number), and within it, the textual content of these articles is included. Legal articles usually have a specific language (formal, legal), which may differ from everyday language and often includes similar terms and phrases.

Number of Documents: With over 400 text files, the data are sufficient for good clustering but not so large as to cause significant computational load for most algorithms, as is the case with hierarchical clustering.

Thematic Uniformity: Although the articles of law are thematically connected (all relate to legislative provisions), there may be different thematic subgroups dealing with different areas (e.g., criminal law, civil law, family law, etc.). This factor may cause some algorithms to be more efficient in detecting these subgroups.

Linguistic Specificity: The texts are formal and technical, with many specific legal terms that are frequently repeated within and between articles. These factors may influence how algorithms recognize similarities between texts.

The data for clustering are represented through TF-IDF vectors that quantify the importance of words in each document, enabling further analysis and segmentation of documents into clusters based on similarities in their topics or content.

7. ANALYSIS OF THE WORK

The goal of this work is clustering documents in Serbian, and in order to achieve this, Python libraries are used, which demonstrates how different clustering algorithms can be applied to textual data, with the aim of segmenting similar texts into groups, thereby allowing a better understanding of the structure and thematic connections among them. Although the code is written for analyzing textual data, the process can be applied to any dataset in the form of character strings (such as articles, documents, comments).

The functionalities implemented through the created code can be divided into several sections:

1. Loading Data:

The function "ucitaj_fajlove" searches the directory containing the text files and loads their content. These texts become the basic data for analysis. The function returns a list of texts and their corresponding file names, which is useful for later insights into the clustering results.

2. Text Transformation:

After loading the texts, the TF-IDF (term frequency-inverse document frequency) vectorizer (from "sklearn.feature_extraction.text") is used to transform the texts into a numerical format suitable for analysis. The TF-IDF method is used to calculate the importance of each word in the text in relation to the entire document set, thus creating a "feature matrix" containing numerical values of word relevance.

3. Dimensionality Reduction:

To enable visualization of these data in a 2D space, t-SNE (t-distributed stochastic neighbor embedding) is used. This algorithm reduces the dimensionality of the data from the original high-dimensional space to 2D, making it possible to graphically display the clustering results.

4. Clustering:

The code implements three different clustering algorithms:

– DBSCAN (density-based spatial clustering of applications with noise);

– Mean-shift;

– Hierarchical clustering (agglomerative clustering).

5. Cluster Visualization:

For each of the mentioned algorithms, the code generates a 2D chart with t-SNE reduced dimensions, displaying different clusters using colors and markers. In each of the three plots (one for each algorithm), it is clearly indicated which texts belong to which cluster.

6. Displaying Clustering Results:

After visualization, the clustering results are printed in the console, showing the clusters and their corresponding files for each algorithm. This section allows users to observe which files were clustered together by the algorithms, which can be useful for analyzing similarities among documents [10].

This code is a useful tool for analyzing and grouping textual data, especially when the goal is to organize documents into thematic groups. By using different clustering methods, users can gain various perspectives on the structure of their data and choose the most suitable algorithm for specific analysis needs [11].

8. ALGORITHM SELECTION

Before deciding which algorithm yields the best results, it is necessary to consider the data on which they are applied, as described in the DATASET section.

Considering the characteristics of the data, it is possible to show how each algorithm fits:

– *DBSCAN* (density-based spatial clustering of applications with noise)

The advantages of this algorithm include the discovery of non-spherical clusters and noise. When it comes to detecting non-spherical clusters, legal texts can be thematically related but not necessarily in perfectly spherical or homogeneous groups. DBSCAN is good at recognizing clusters of different shapes, which could be useful for data like laws, where themes are interconnected but not necessarily in a linear or "smooth" pattern. Detecting noise is also one of its advantages, as if some articles in the law are not related to the others (e.g., very specific or unusual provisions), DBSCAN can identify them as "noise." This can be useful, as it helps identify potentially unorganized or less important documents.

In addition to its advantages, there are some limitations, primarily its sensitivity to parameters. The "eps" (maximum distance between points considered similar) and "min_samples" (minimum number of points to form a cluster) parameters must be carefully set. If they are set too strictly, important clusters may be lost. If they are too loose, the algorithm might capture too much noise. In terms of limitations, the problem of a large number of small clusters can occur. If the texts are not sufficiently diverse, DBSCAN may form a small number of clusters and too much noise, which can later complicate interpretation.

DBSCAN could be useful if the goal is to identify thematic groups that are linked by the density of data and there is no desire to impose a pre-defined number of clusters. If some articles in the law are very specific and cannot easily be grouped with others, DBSCAN can recognize these articles as noise.

– Mean-Shift

Advantages of the mean-shift algorithm: One of the advantages of the mean-shift algorithm is its automatic determination of the number of clusters. Mean-shift does not rely on a predefined number of clusters, which makes it ideal for data such as legal articles, where the number of thematic clusters is not known in advance. Another important advantage is that it detects dense clusters. Therefore, if there are thematic areas that are dense in terms of specific legal terms (e.g., all articles related to criminal law), mean-shift will automatically detect these natural clusters.

Limitations of mean-shift: The limitations of this algorithm are related to the adjustment of the "bandwidth" parameter and computational complexity. Regarding the "bandwidth" parameter (which defines the width of the region used to form clusters), if it is not properly set, the algorithm may either "merge" clusters too much or create too many small, unnecessary clusters. On the other hand, for larger datasets, mean-shift can be slow, but for 400 texts, this should not be a major issue.

When to use mean-shift: Mean-shift is useful for automatically recognizing thematic groups in texts that may have natural densities. Since legal articles can be distributed across different thematic areas (criminal, civil, family law), Mean-Shift could effectively identify these areas.

– *Hierarchical Clustering* (Agglomerative Clustering)

Hierarchical clustering, like the previous two algorithms, also has two advantages that need to be mentioned: versatility in defining the number of clusters and flexibility with linkage methods. Although this type requires a predefined number of clusters, hierarchical clustering allows for exploration of different clustering levels using a dendrogram. This enables a deeper understanding of the data structure and easier decision-making regarding the number of clusters, demonstrating its versatility. When it comes to flexibility with linkages, the algorithm offers various ways to merge clusters (e.g., "ward," "single," "complete"), allowing fine-tuning depending on the nature of the data.

Limitations of hierarchical clustering: The algorithm also has some limitations, such as computational complexity. Hierarchical clustering can be computationally intensive, especially for larger datasets, because it requires calculating distances between all pairs of points. Additionally, the algorithm does not always provide a clear interpretation. While

it gives a good insight into the data structure, it can be challenging to interpret clusters in textual data, especially if the clusters are not distinctly separated.

When to use hierarchical clustering: Hierarchical clustering is useful when you want to explore various levels of granularity in clustering and when you have sufficient time and resources to compute the dendrogram. Additionally, if the number of clusters is known or if you want to test different options for the number of clusters, hierarchical clustering can be useful [12].

9. RESULTS

In this research, clustering of textual documents based on the articles of the Criminal Code was performed. The input data consisted of .txt files, where each document corresponded to one article of the law, and the file name indicated the article number. The clustering enabled the analysis of similarities between different articles and the identification of thematic groups that appear in the legislative text.

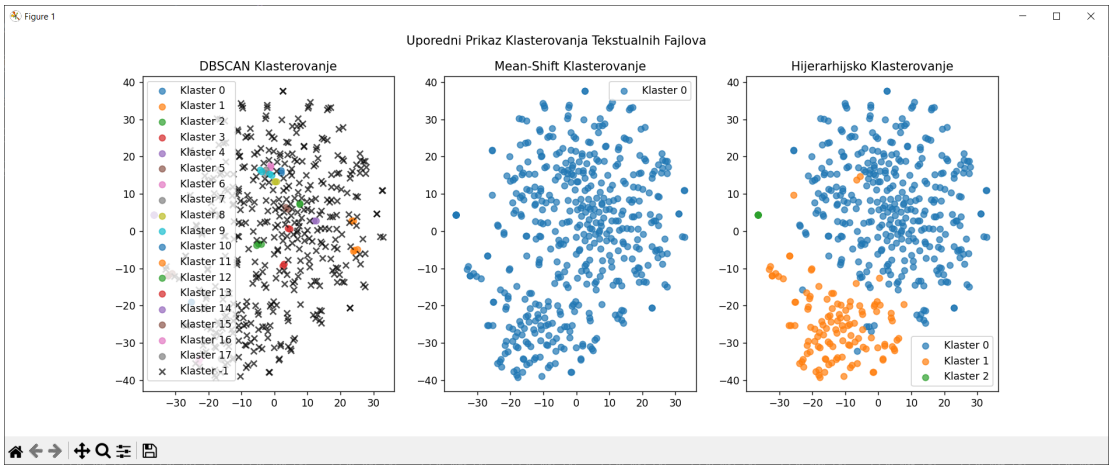

Figure 1. *Clustering results.*

1. Clustering Results using the DBSCAN Algorithm

DBSCAN identified several clusters, with most documents grouped into a large noise cluster (-1). Additionally, several smaller clusters (0-17) were formed with very specific sets of documents:

Cluster -1: The largest cluster, containing a wide range of documents.

• Clusters 0–17: Smaller, more coherent groups of documents, often including specific documents like Član_109.txt and Član_110.txt in Cluster 0, or Član_1.txt and Član_10.txt in Cluster 1.

2. Clustering Results using the Mean-Shift Algorithm

Mean-shift produced a similar division, mainly grouping most documents into one large cluster (0). However, smaller clusters were also formed:

• Cluster 0: Encompasses most of the documents, similar to DBSCAN's main cluster.

• Additional clusters: Several smaller groups were identified with documents such as Član_109.txt and Član_110.txt, which were grouped together in Cluster 0, while specific groups formed smaller clusters with documents like Član_208.txt and Član_223.txt.

3. Clustering Results using the Hierarchical Algorithm

Hierarchical clustering produced three clusters, indicating that the documents were grouped into three main groups at different levels of granularity. The results show that clusters were organized hierarchically, with smaller and more specific clusters formed within larger clusters. This method provides a clear division into three clusters with different levels of detail (Figure 1).

By reviewing the above results, the following conclusions can be drawn related to each algorithm:

• DBSCAN: Most documents were classified as noise (-1), with several distinct smaller clusters.

• Mean-shift: Similar to DBSCAN, with most documents in one large cluster, although smaller clusters also exist.

• Hierarchical clustering: This method produced three clusters, allowing finer control over the number of clusters and providing insight into the hierarchy of document groups.

Each algorithm provides valuable insights into the similarities and relationships among documents, with DBSCAN being particularly sensitive to noise, mean-shift focusing on high-density areas for clustering, and hierarchical clustering offering flexibility in choosing the number of clusters and enabling analysis at different levels of granularity.

10. CONCLUSION

The research on the application of clustering algorithms to textual documents from the Criminal Code has shown varying results depending on the chosen algorithm. Each of the analyzed algorithms—DBSCAN, mean-shift, and hierarchical clustering—has its own specificities that impact the quality and precision of the resulting clusters.

DBSCAN, while effective in recognizing non-spherical clusters and noise, presented some challenges in processing legal texts, where some clusters were too diverse, and noise was not entirely filtered out. On the other hand, the Mean-Shift algorithm successfully identified dense areas of the data, efficiently grouping thematically similar articles from the Criminal Code without the need for a predefined number of clusters. Although this method has a significant advantage in automatically determining the number of clusters, fine-tuning of parameters was necessary to achieve an optimal data partitioning. Hierarchical clustering allowed for detailed analysis and the creation of clusters at different levels of granularity, which is particularly useful when a broader picture of the relationships between various thematic areas in legislative texts is desired.

Based on the obtained results, we can conclude that no method is universally the best and that the choice of algorithm depends on the specific goals of the analysis. For legal texts such as the Criminal Code, hierarchical clustering proved to be the most efficient method

for identifying thematic groups with multiple levels of granularity. However, for faster analysis and detection of specific dense clusters, mean-shift yielded very useful results. DBSCAN can be useful in the context of larger, disordered datasets, but it requires additional parameter adjustment for optimal results.

This research provides valuable guidelines for further application of clustering algorithms in text analysis, as well as recommendations for their effective use in similar domains.

FUNDING

This research received no external funding.

INSTITUTIONAL REVIEW BOARD STATEMENT

Not applicable.

INFORMED CONSENT STATEMENT

Not applicable.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] A. J. Smith and K. L. Brown, "Data Clustering Techniques in Big Data Analytics," in *TRIP Symposium on Data Science Applications*, 2018.

[2] L. Silva and M. Johnson, "Analysis of Fixed and Variable Cluster Number Techniques in K-means and K-medoids Clustering," in *Proceedings of the 18th International Conference on Data Clustering and Machine Learning Applications (TRIPOLI)*, 2021.

[3] A. Lopez and Y. Wang, "Exploring Density-Based and Hierarchical Methods for Unspecified Cluster Numbers in Data Analysis," in *Proceedings of the 19th International Conference on Advanced Data Clustering Techniques (TRIPOLI)*, 2022.

[4] R. Gonzalez and Y. Kim, "Application of DBSCAN for High Noise Data Clustering," in *Proceedings of the 19th TRIPOLI Conference on Advanced Clustering Techniques*, 2020.

[5] R. Mitchell and T. Jang, "Comparative Analysis of Hierarchical, DBSCAN, and Mean-Shift Clustering Algorithms," in *Proceedings of the 22nd TRIPOLI Conference on Data Clustering and Analysis*, 2022.

[6] L. Patterson and Y. Chen, "Mean-Shift Clustering in Image and Object Recognition," in *TRIPOLI Conference Proceedings on Non-Parametric Clustering Approaches*, 2021.

[7] L. Chan and T. Gomez, "Hierarchical Clustering Methods and Their Applications in Small to Medium Data Sets," in *Proceedings of the 20th TRIPOLI Conference on Data Clustering and Visualization*, 2021.

[8] H. Liu and Z. Zhang, "Python Libraries for Clustering and Visualization of Textual Data," *Journal of Computational Data Analysis,* pp. 120–134, 2023.

[9] X. Chen and T. Fischer, "Advanced Libraries for Text Data Clustering and Visualization," in *Proceedings of the TRIPOLI Conference on Text Data Analysis*, 2022.

[10] S. Molina and M. Albrecth, "Application of Clustering Algorithms for Textual Data Analysis," in *Proceedings of the 8th International Conference on Advanced Data Science and Computing (TRIPOLI)*, 2023.

[11] Y. Zhang and T. Li, "Clustering Techniques for Textual Data in Natural Language Processing," in *Proceedings of the 8th International Conference on Advanced Data Science and Computing (TRIPOLI)*, 2023.

[12] M. Jovanović and D. Petrov, "Comparison of Clustering Algorithms for Legal Text

Analysis: DBSCAN, Mean-Shift, and Hierarchical Clustering," in *Proceedings of the 9th International Conference on Data Science and Artificial Intelligence (TRIPOLI)*, 2023.

[13] V. Nikolić, M. Čabarkapa, J. Mišić, D. Ranđelović, and S. Nedeljković, "An Advanced Quick-Answering System Intended for the e-Government Service in Republic of Serbia," *Acta Polytechnica Hungarica,* Vol. 16, pp. 153–174, 2019.

[14] B. Markoski, K. Kuk, D. Ranđelović, P. Čisar, and V. Nikolić, "Modelling the System of Receiving Quick Answers for e-Government Services: Study for the Crime Domain in the Republic of Serbia," *Acta Polytechnica Hungarica,* Vol. 14, pp. 143–163, 2017.