mathematics

Article

An Effective Fuzzy Clustering of Crime Reports Embedded by a Universal Sentence Encoder Model

Aparna Pramanik [1], Asit Kumar Das [1,*], Danilo Pelusi [2,*] and Janmenjoy Nayak [3]

[1] Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, Howrah 711103, West Bengal, India

[2] Department of Communication Sciences, University of Teramo, 64100 Teramo, Italy

[3] Post Graduate Department of Computer Science, Maharaja Sriram Chandra Bhanja Deo (MSCB) University, Baripada 757003, Odisha, India

[*] Correspondence: akdas@cs.iiests.ac.in (A.K.D.); dpelusi@unite.it (D.P.); Tel.: +91-9830342574 (A.K.D.)

Abstract: Crime reports clustering is crucial for identifying and preventing criminal activities that frequently happened in society. In the proposed work, named entities in a report are recognized to extract the crime-related phrases and subsequently, the phrases are preprocessed by applying stopword removal and lemmatization operations. Next, the module of the universal encoder model, called the transformer, is applied to extract phrases of the report to get a sentence embedding for each associated sentence, aggregation of which finally provides the vector representation of that report. An innovative and efficient graph-based clustering algorithm consisting of splitting and merging operations has been proposed to get the cluster of crime reports. The proposed clustering algorithm generates overlapping clusters, which indicates the existence of reports of multiple crime types. The fuzzy theory has been used to provide a score to the report for expressing its membership into different clusters, and accordingly, the reports are labelled by multiple categories. The efficiency of the proposed method has been assessed by taking into account different datasets and comparing them with other state-of-the-art approaches with the help of various performance measure metrics.

Keywords: crime report analysis; named entity recognition; universal encoder-based feature embedding; graph-based clustering; overlapping clusters; fuzzy theory

MSC: 05C72

check for updates

Citation: 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. https://doi.org/10.3390/math11030611

Academic Editor: Alexander Zeifman

Received: 15 December 2022
Revised: 12 January 2023
Accepted: 18 January 2023
Published: 26 January 2023

1. Introduction

The rate of crimes is occurring and frequently increasing in various places across the world. Technology advancements have made this information easily accessible on social media. This huge amount of information can be divided into different groups based on their crime categories to make it convenient for police personnel and investigators to take appropriate actions for reducing criminal activities in society. The clustering algorithms take an important role in this purpose, which group together the crime reports of similar crime types. There are many clustering methods [1,2] that can be used to cluster both structured and unstructured datasets. But research on report clustering over a long period of time has shown that it is neither an easy task nor a perfect solution yet. Here, we have proposed a novel overlapping clustering algorithm using the graph theory for partitioning the crime reports of different categories. The clustering has been carried out considering the concepts of graph theory, such as the clustering coefficient, degree of the nodes, and edge density of the graph. After generating the clusters of crime reports, a fuzzy technique has been introduced to set scores of each report, which give the degree of memberships of the report to reside in different clusters; and finally, the reports are labelled by multiple classes (i.e., crime types) based on their membership values. In the proposed work, data preprocessing techniques take important prior steps to represent the reports in a structured form, which not only helps for efficient clustering of the reports but also extracts the effective information from the reports to facilitate the clustering process.

Initially, the crime reports are collected [3] and the named entities are recognized [4] to select only the crime-related phrases. Next, the stopwords are removed, and lemmatization [5] is done on the extracted phrases to select only the meaningful root words of the crime-related phrases, which are finally used for report embedding. To achieve this, Universal Sentence Encoder (USE) [6], one of the most well-performing sentence embedding techniques, is applied. The key feature that inspires us to use it is its wide application in multi-task learning tasks like sentiment analysis, sentence similarity, clustering, and classification. The USE model is developed based on two encoders, namely Transformer and Deep Averaging Network(DAN). Both of these models are capable of taking a word or a sentence as input and generating embeddings for the same. The models take the sentences as input, tokenize them, and convert each sentence to a 512-dimensional vector, the average of which provides a 512-dimensional vector of the report. The function of the transformer is similar to the encoder module of the transformer architecture, and it uses the self-attention mechanism. The DAN computes unigram or bigram embeddings first and then average them to get a single embedding, which is subsequently passed to a deep neural network to obtain a final sentence embedding of 512 dimensions. We have used the transformer model in sentence embedding for its simplicity and efficiency.

1.1. Literature Survey

Community detection or partition of a graph into subgraphs is crucial for identifying the coherent groups or clusters where the elements inside a cluster are tightly connected. In literature, various partitioning algorithms are presented to detect the communities or partitions for different problems and the structures of these partitions are mostly hierarchical clusters [7], overlapping clusters [8] and disjoint clusters [9]. A semi-supervised graph partitioning algorithm has been introduced in [10], and it employs graph regularisation to blend past information with the network topology. Girvan et al. [7] proposed a graph-based method to make the clusters in a hierarchical way. In this approach, they removed the edge with the highest betweenness to make the clusters and at the last stage, every report has been placed separately. A graph clustering algorithm has been proposed by Bianchi et al. in their paper [11] with addressing the constraints of spectral clustering. They have applied the graph neural network model and embedded min-cut pooling operation to make the clusters. In [12], K. Taha utilized the concept of edge betweenness, relative importance score, and degree of association scores to find the disjoint cluster within a graph. But in real life, there exist many problems in which it has been seen that the clusters are overlapping in nature; therefore, many researchers have proposed different algorithms for generating overlapped clusters. Ghoshal et al. [13] have introduced an algorithm to detect disjoint and overlapping communities based on mean path length accompanying the modularity index in the Genetic Algorithm. In [14–16], different approaches have been highlighted to detect overlapping communities from a network. The node influence has been identified in [14] by measuring the degree centrality of a node, and another factor called agglomeration coefficient has also been considered for the task. In [16], the label propagation technique has been used for finding overlapping communities. Rezvani et al. [17] have detected overlapping clusters by proposing a novel community fitness metric, named as triangle-based fitness metric. Whang et al. [15] have proposed the neighborhood inflation technique to detect overlapping communities. Initially, they determined the good seed nodes in a graph. Later, the PageRank clustering scheme has been applied to optimize the conductance community score. The important step of their method is for neighborhood inflation, where seeds are modified to represent their entire vertex neighborhood, and the drawback of their method is that it produced much larger communities to cover the entire graph. The overlapping and non-overlapping communities have been detected in [18] by introducing vertex-based metrics called GenPerm. In [19], an overlapping community detection algorithm, named Scalable Spectral Clustering algorithm, is proposed, which is an extension of the notion of normalized cut and is able to find overlapping communities in a large network. In addition to these methods, there exist several fuzzy techniques to detect the communities that estimate the likelihood of each node belonging to each community. But, the majority of these algorithms require prior knowledge, such as community size, and community number. Su et al. [20] and Yazdanparast et al. [21] have applied the fuzzy method for community detection by modularity maximization. The concept of self-membership has been introduced in paper [22]. Here, the method allows all the nodes to grow their own community and the anchor nodes are those with a higher degree of self-membership which have the opportunity to grow the linked community. While incorrect or unnecessary anchors are eliminated, some new anchors may appear in subsequent iterations. In [23], the authors have proposed a multiobjective fuzzy clustering method where they have optimized the cluster compactness and level of fuzziness. The concepts of fuzzy F^*-simply connected spaces and fuzzy F^*-contractible spaces are presented by Madhuri et al. [24]. Later, they analysed some significant characteristics of fuzzy F^*-homotopy and also proved that each fuzzy F^*-loop based at any fuzzy point in fuzzy F^*-contractible space is equivalent to the constant fuzzy F^*-loop. Dhanya et al. [25] proposed a fuzzy hypergraph-based model to predict crimes in various locations. The crime fuzzy hypergraph contains two layers: an outer level and an interior level. Both levels have been subjected to morphological procedures like dilation and erosion. The authors in paper [26] have also used the fuzzy clustering method for text categorization. It follows some steps such as fuzzy transformation for dimensionality reduction, cluster membership assignment, cluster-to-category mapping, and finally, getting the assigned category by applying a threshold. Meng et al. [27] have introduced a new measure called the network motif, which is a small connected subgraph that contains multiple nodes and edges and represents the information interactions among the nodes. In our paper, we have proposed an innovative euclidean distance-based fuzzy clustering algorithm using graph splitting and subgraph merging operations for the clustering of crime reports.

1.2. Motivation and Objective

One of the major issues facing humanity is crimes, which pose the greatest danger to every human on the globe. As criminal activities are increasing day by day, crime report analysis is very important to prevent it in society. The main purpose of this work is to select crime-related information from a wide variety of crime reports and share it with police officers so that they can take preventive measures against criminal activities. Analyzing the crime reports manually is a very difficult task and quite impossible for the huge volume of a complex dataset. Therefore, different types of crime report analysis techniques have been presented, such as classification of crimes, clustering of crimes, location detection, and many more. In practice, most of the generated crime reports are unlabeled, so unsupervised learning, such as the clustering approach, is more effective for crime report analysis. Clustering of crime reports aids in identifying connections and linkages between illegal activity. In crime report clustering, crime reports are placed in different groups based on their context, so when the investigators want to investigate a particular crime type, they can focus on a particular cluster of reports, which reduces the time complexity of the investigation as well as helps to provide more effective information. Therefore, it is required to group the crime reports according to the crime types. However, it's possible for one piece of information about a crime to contain information about another type of crime. This creates an overlapping cluster dilemma where one crime incidence can fit into many crime types. For example, suppose in a crime incident, it has been found that someone kidnapped a person and then killed him. So, this crime incident falls into two categories. While many efforts have been made to locate overlapping groups of reports, relatively few have been successful in locating crime reports that contain information on several crime types. Additionally, there are numerous graph partitioning methods that yield an excessively large number of overlapping clusters and have significant computational costs. There are numerous edges connecting a node for a generic document to other nodes for comparable documents since some documents are very general and consequently similar to many other documents. To improve the partitioning quality, these kinds of edges must be eliminated. Therefore, a novel graph-based fuzzy clustering technique has been proposed to address the overlapping clustering problem effectively.

1.3. Contribution

As a contribution, we have applied an innovative data preprocessing method where only the noun phrases for each report have been bunched together. Then these newly formed reports have been processed and clustered by our proposed graph-based clustering algorithm. The main contribution of our work is to produce overlapping clusters with fuzzy membership values of each report in overlapping regions for the purpose of crime analysis. Initially, named entities of each report have been detected, and noun phrases are bunched together. Next, the extracted phrases of each report have been preprocessed by removing the stopwords from the sentences and selecting the root words using lemmatization. Then the preprocessed phrases of a report are embedded by applying transformer architecture based Universal Sentence Encoder (USE) [6] and obtain the report embedding by averaging all the phrase embeddings of the report. Subsequently, a graph has been constructed based on the ξ-ball graph construction method [28], where the vector representation of each report has been considered as a node and cosine similarity between a pair of nodes is represented by an edge in the graph. The cosine similarity between each pair of nodes is measured, and if the similarity crosses a threshold, then an edge is placed between them. Later, the overlapping clusters have been discovered by following two steps, namely splitting a graph into subgraphs, and merging subgraphs into a graph. The splitting operation partitions the graph by considering the clustering coefficient and degree centrality measures, whereas the merging operation fuses subgraphs based on the edge density measure of the graph to obtain the optimal set of clusters. Finally, an innovative fuzzy technique has been introduced for the reports those lie in multiple clusters to assign the degree of memberships, which helps to label the reports by multiple crime types.

The workflow diagram of the proposed work is shown in Figure 1, and the main contributions of the paper are concluded by the following few steps.

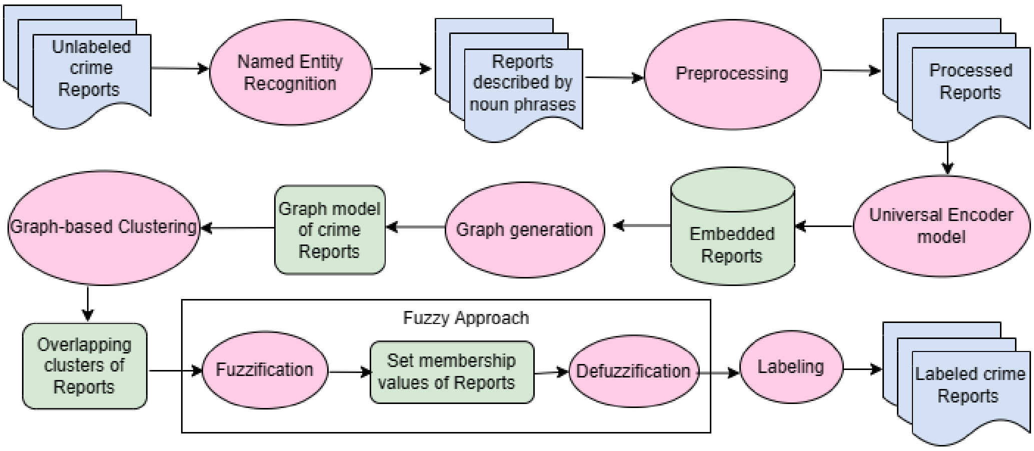

Figure 1. Workflow diagram of the proposed methodology.

1. After collecting the dataset, named entities are recognized to extract the noun phrases of the reports, which are subsequently preprocessed by following stopword removal and lemmatization operations. Then each report has been converted to a vector by applying a transformer architecture-based Universal Sentence Encoder model on the collection of extracted processed noun phrases of the report.
2. An undirected graph is constructed where each report vector is considered as a vertex, and an edge exists between a pair of vertices if the cosine similarity score between them crosses a predefined threshold.

3. A novel graph-based overlapping clustering algorithm has been deduced based on splitting and merging operations. In the splitting operation, a graph is split into subgraphs using the clustering coefficient and degree of the vertices, and in the merging operation, a graph is reformed by fusing two subgraphs based on edge density.
4. Fuzzy theorem is applied on overlapping clusters, where fuzzification is done to provide membership values to the reports lying in the overlapping regions, and defuzzification is done to label the reports by multiple crime types. Thus, reports outside overlapping regions of the clusters are of a single crime type and those in overlapping regions are of multiple crime types.

1.4. Summary of the Paper

The remaining sections of the paper are arranged as follows: Section 2 describes the preprocessing and report embedding process, and the proposed graph-based fuzzy overlapping clustering algorithm is described in Section 3. The experimental results and discussions are presented in Section 4. Finally, in Section 5, the conclusion and the future work have been discussed.

2. Preprocessing and Report Embedding

Here, the collected crime reports are preprocessed to remove the irrelevant words and extract only the root words of the reports. Also, each report is represented by a vector using a universal sentence encoder model.

2.1. Preprocessing of Reports

The unlabelled crime reports have been collected from an online platform and described by the short description together with the headline of the report and removed all other information from the report. The words of each report have been tokenized and assigned with the tag of part of speech. This operation is carried out by the Natural Language Tool Kit (NLTK)'s [29] built-in sentence segmenter, word tokenizer, and parts-of-speech tagger by default. The next step is to look for any named entities present in a sentence by bunching noun phrases [4]. This process has been depicted through an example in Figure 2. Here, PPR stands for Personal Pronoun, NN for Noun, VBD for verb. The example contains the named entities shown in two larger square boxes on both sides of the tokenized word 'killed', which has been tagged as VBD. These noun phrases have been collected and bunched together for each report. Then the stopwords have been discarded, and a lemmatization operation has been performed to find the root words. Thus after preprocessing, the sentence "Her husband killed their children" becomes "husband kill children". This noun phrase has a pair of named entities, namely "husband" and "children", which are related by "Kill", which is a crime-related word. Thus, our objective in this preprocessing step is to represent each report as the collection of preprocessed noun phrases, which are applied to the universal sentence encoder model for report embedding.

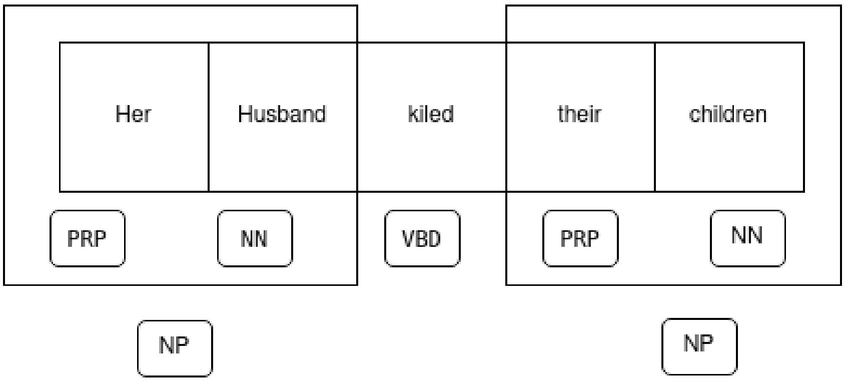

Figure 2. Bunching of noun phrases.

2.2. Report Embedding

We have used transformer architecture based Universal Sentence Encoder model [6] for the purpose of embedding. The model takes the input as a lowercase PTB tokenized string and produces output, a 512-dimensional vector as the sentence embedding. This model is a Transformer architecture, which provides better accuracy on downstream tasks but imposes significantly higher computational complexity due to its complex architecture. Its computation time scales dramatically with the length of the sentence. In our work, we have considered preprocessed phrases as individual sentences which are of very small length, and thus the encoder model is used efficiently. Also, we have used the publicly available pre-trained universal sentence encoder, which also reduces the time complexity of the proposed work. The transformer architecture's encoding sub-graph is used by the module to carry out the sentence embeddings [30]. This sub-paragraph employs attention to compute contextual word representation in a phrase that takes into consideration the identity and order of every other word. The sum of representations at each word location is calculated element-by-element to turn the contextual word representations into a fixed-length sentence encoding vector. Then the average of the encoded vectors of all extracted phrases of a report has been calculated and used to represent the report in vector form and it has been taken to accomplish the rest of the work.

3. Graph Based Fuzzy Clustering

The proposed clustering method takes the report embeddings as the input and extracts the inherent groups of similar crime reports naturally, without having prior knowledge about the number of the groups and the size of each of the groups. The relationships among the reports have represented by a graph $G = (V, E)$. The graph has been constructed with each report has been treated as a vertex for the graph. So, a vertex of the graph is basically a vector representation of a particular report. An edge has constructed between a pair of vertices if the cosine similarity between two respective report embeddings crosses a predefined threshold. This concept of graph construction is used in paper [28] (named as ξ-ball), which has been applied in our work to construct the graph. The created graph is undirected, and depending on the similarity value, it can even be disconnected. When the graph becomes connected, the proposed clustering algorithm based on Splitting, and Merging operations is applied to it to produce overlapping subgraphs, each of which produces a cluster of reports. If the constructed graph becomes a disconnected graph, then the proposed graph-based clustering algorithm applies to every component individually. The proposed algorithm is developed based on the concept of clustering coefficient of a vertex (the fraction of possible triangles through that vertex), degree of a node (number of edges incident on the vertex), and edge connectivity of the graph (the number of edges divided by the maximal number of edges) and followed two steps, namely Splitting, and Merging steps.

3.1. Splitting

We split the graph based on the clustering coefficient and degree of the vertices. The clustering coefficient and degree of each node have been calculated for all the vertices present in the graph, and the vertex with the highest clustering coefficient has been chosen for performing the splitting operation. In case of the existence of multiple vertices with the same clustering coefficient, we have considered a node with the highest degree from those vertices, and even after that, if the tie exists between multiple such vertices, then a vertex has been randomly selected from those tie sets. Considering this vertex, the partition has been made on the graph to get the subgraphs. If vertex v of the graph $G = (V, E)$ is the selected one for splitting the graph, then we create a set V_1 of vertices that consists of v and all its neighbours in G. Next, we create a graph $G_1 = (V_1, E_1)$, where E_1 is the subset of edges of E with end vertices of the edges in V_1. Next, we remove the subgraph G_1 from G to get G_2 without neglecting any edge of G. That is, though a vertex is in G_1, it may also appear in G_2 to keep all the edges in G_2 which are not in G_1. Thus the splitting operation provides the pair of overlapping subgraphs, G_1, and G_2. If G_2 is a null graph, the process terminates. Otherwise, the same splitting process is continued for graph G_2. Thus the process provides a list of overlapping subgraphs. If all the vertices of a subgraph are covered by some of the other subgraphs, i.e., if each vertex of a subgraph is a vertex of some other subgraphs in the list, then the subgraph is redundant and removed from the list of subgraphs. This operation has been illustrated through Figure 3. In Figure 3, we can see that vertex, a has the highest clustering coefficient and degree, and so the graph G has been split into G_1 and G_2 considering this vertex. In G_1, all vertices adjacent to a have been kept, and the connected edges between them have also been preserved. The remaining portion comes out as G_2. In G_2, the vertices v, d, e, and f are of the highest clustering coefficients and degrees, so we randomly select any one vertex, say v, for splitting G_2. Repeating this process we obtain the subgraphs G_{21}, G_{22}, G_{23}, and G_{24}. But, the vertices of G_{24} are covered by G_1, G_{21}, and G_{23}, and so G_{24} is removed. Thus, after performing splitting operation on the given graph G, we have the subgraphs G_1, G_{21}, G_{22}, and G_{23}, as shown by green color subgraphs in Figure 3. The pseudocode of the proposed splitting algorithm is given in Algorithm 1.

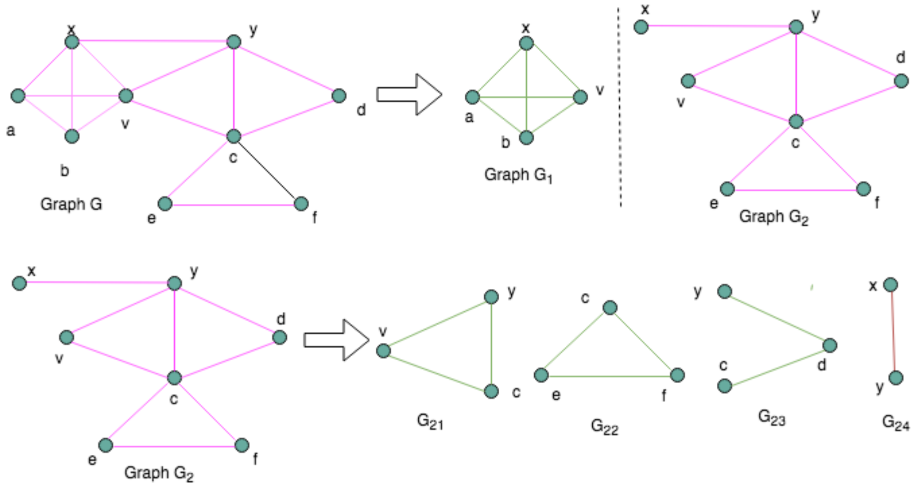

Figure 3. Splitting Operation.

3.2. Merging

After the splitting process, the merging operation performs over the subgraphs, based on the edge density of the subgraphs. In the merging process, two subgraphs are fused if the edge density of the resultant subgraph is greater or equal to the average of the edge densities of both the individual graph. Here, we check the overlapping region of every pair of subgraphs in the list S. We start to merge two subgraphs for which the overlapping region contains a maximum number of vertices. Next, among the resultant subgraphs, consider two subgraphs with a maximum number of common vertices for merging and so on. The process terminates if no more merging is possible. The merging step has been explained in Figure 4. After splitting operation, the graph G has been partitioned into a list of subgraphs, G_1, G_{21}, G_{22}, and G_{23}. Here, the subgraphs, G_{21} and G_{23} has maximum common vertices which are c and y. Here, the edge density of G_{21} and G_{23} are 1.0 and 0.66, respectively. So the average edge density is 0.83. If we merge them, then the resulting graph becomes G_{213} and its edge density is 0.83, which is equal to the average edge density of G_{21} and G_{23}. Thus we get resultant subgraphs, G_1, G_{22}, and G_{213}. Though there are common vertices between G_{22}, G_{213} and G_1, G_{213} but based on the condition of merging, they fail to merge. So the final set of subgraphs is $\{G_1, G_{22}, G_{213}\}$, and the clusters are $\{a, b, v, x\}$, $\{c, e, f\}$, and $\{c, d, v, y\}$. The pseudocode of the merging operation is described by Algorithm 2.

Algorithm 1: Split a Graph into subgraphs - $SPLIT(G, S)$

```
Data: Graph $G = (V, E)$
Result: $S$ = list of subgraphs
begin
    $S = \emptyset$;
    while $(G \neq NULL)$ do
    end
    Compute clustering coefficient ($CC$) and degree of each vertex $v \in V$;
    $V^{'}$ = Set of vertices of $G$ with maximum $CC$;
    $V^{''} \subseteq$ V' is the set of vertices with maximum degree;
    $v$ = a vertex randomly selected from $V^{''}$;
    $N_v$ = Set of all neighbours of $v$;
    Compute $V_1 = \{v\} \cup N_v$;
    Compute $E_1$ = edges of $E$ exist between each pair of vertices in $V_1$;
    Let, $G_1 = (V_1, E_1)$;
    $E_2 = E - E_1$ and $V_2$ = end-vertices of all edges in $E_2$;
    Let, $G_2 = (V_2, E_2)$;
    $S = S \cup \{G_1\}$;
    $G = G_2$ i.e., $V = V_2$ and $E = E_2$;
    for each subgraph $G_s = (V_s, E_s)$ in $S$ do
    end
    $R_{G_s}$ = True;
    for each $v$ in $V_s$ do
    end
    if $v \notin \cup_{\forall G_s} V_s - V_s$ then
    end
    $R_{G_s}$ = False;
    break;
    if $(R_{G_s})$ then
    end
    $S = S - \{G_s\}$;
    Return S;
end
```

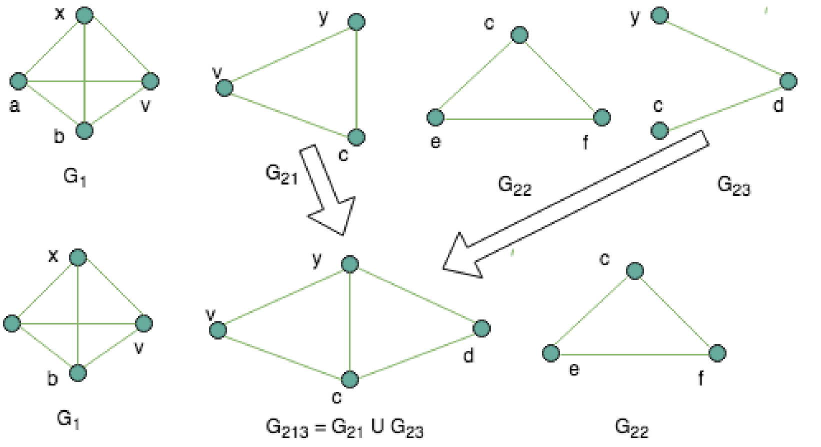

Figure 4. Merging Operation.

Algorithm 2: Merge subgraphs into graphs-$MERGE(S)$

```
Data: $S$ = List of subgraphs obtained by SPLIT(G, S)
Result: $S$ = Final set of subgraphs
begin
    for each pair of subgraphs $G_i$ and $G_j$ in $S$ do
        $No\_Merge$ = 0;
        $L = \emptyset$ /*empty list with 3 items in each node*/;
        Compute $C_{ij} = |V_i \cap V_j|$;
        if $C_{ij} > 0$ then
            Insert node $< G_i, G_j, C_{ij} >$ into the list $L$;
        end
        Arrange the nodes of L in descending order of item no. 3;
        for each node $< G_i, G_j, C_{ij} >$ in $L$ do
            $G_k = G_i \cup G_j$ /* Merging of graphs*/;
            Compute $d_i$ = edge density of $G_i$;
            Compute $d_j$ = edge density of $G_j$;
            Compute $d_k$ = edge density of $G_k$;
            if $d_k > (d_i + d_j)/2$ then
                $S = S - \{G_i\} - \{G_j\} \cup \{G_k\}$;
                break;
            end
            else
                $No\_Merge = No\_Merge + 1$;
            end
        end
        if $No\_Merge == |L|$ then
            break;
        end
    end
    Return S;
end
```

3.3. Fuzzy Theory and Report Labelling

After applying the splitting and merging operations in G, we have found a set of overlapping subgraphs, which implies that some crime reports have been placed in more than one cluster. Therefore, A fuzzy theory has been applied to handle the overlapping problem. We have applied fuzzification by defining a euclidean distance-based membership function. This membership function gives the membership value by which a report belongs to a cluster. This has been applied only to the reports which belong to more than one cluster. We have already embedded each report in a 512-dimensional vector. First, we compute the mean of all elements of a cluster and considered it as the representative of that cluster. Let, a report, say r_i lies on $t-$clusters, say $C_1, C_2, \ldots, C_t$. Then the membership value, μ_{ij} by which report r_i lies in cluster C_j is defined by Equation (1), where d_{ij} is the euclidean distance between report r_i and cluster C_j. Thus, report r_i has $t-$membership values, $\mu_{i1}, \mu_{i2}, \ldots, \mu_{it}$ using fuzzification technique.

$$\mu_{ij} = 1 - \frac{d_{ij}}{\sum_{j=1}^{t} d_{ij}} \quad (1)$$

After assigning the membership values to the reports in the overlapping regions, we apply defuzzification. We consider a threshold δ for defuzzification and if the membership value of a report r_i to reside in a cluster C_j is less than δ, then the report r_i is removed from

C_j. After applying the defuzzification technique, a report may still be in a few clusters based on the δ value. Thus, after defuzzification, the report r_i may be in $l-$clusters where $l < t$. Next, for labelling the reports, we first label the clusters by different crime types. As each report is described by the noun phrases and two named entities in a phrase are related by some crime words, so we select a set of such words for each cluster. The highest frequency word is selected from the set and the cluster is labelled by this crime-related word. So, each report in the non-overlapping region of a cluster is labelled by the label of the cluster. But, if a report r_i of the overlapping region has $l-$membership values after defuzzification, then r_i is labelled by the label of corresponding $l-$clusters. So, when we want to investigate the reports of some particular crime type, we simply extract the reports labelled by this crime type, which makes the investigation process simpler. The pseudocode of the crime report labelling technique is described by Algorithm 3.

Algorithm 3: Fuzzy Theory based Crime Report Labelling-FTCRL(S, R)

```
Data: $S$ = List of subgraphs obtained after MERGE($S$)
Result: $R$ = Set of labelled reports
begin
    $CLUS = \emptyset$ /*set of clusters of reports*/;
    for each subgraph $G_s = (V_s, E_s)$ in $S$ do
        $C_s = \emptyset$ /*cluster of reports*/;
        for each $v \in V_s$ do
            $C_s = C_s \cup \{v\}$;
        end
        $CLUS = CLUS \cup \{C_s\}$;
    end
    /*Labelling of clusters*/
    for each $C_s \in CLUS$ do
        $W = \emptyset$;
        for each report $r \in C_s$ do
            $W_r$ =Set of all words between pair of named entities in $r$;
            $W = W \cup W_r$;
        end
        Find word $w_s \in W$ of highest frequency;
        Label cluster $C_s$ by crime type $w_s$;
    end
    $All\_report = \cup_{\forall C_s \in CLUS} C_s$;
    $OVR = NON\_OVR = \emptyset$ /*overlapping and non-overlapping regions*/;
    for each $r \in All\_report$ do
        $Count = 0$;
        for each $C_s \in CLUS$ do
            if $r \in C_s$ then
                $Count = Count + 1$;
            end
        end
        if $Count$ == $1$ then
            $NON\_OVR = NON\_OVR \cup \{r\}$;
        end
        else
            $OVR = OVR \cup \{r\}$;
        end
    end
    $R = \emptyset$ /* Set of all labelled reports*/;
end
```

Algorithm 3: *Cont.*

```
for each report $r_i \in NON\_OVR$ do
    for each cluster $C_s \in CLUS$ do
        if $r_i \in C_s$ then
            $R = R \cup \{< r_i, w_s >\}$ /* $r_i$ is labelled by $w_s$*/;
        end
    end
end
for each report $r_i \in OVR$ do
    Let $r_i \in \cap_{j=1}^{t} C_j$;
    $label = \emptyset$;
    for $j = 1$ to $t$ do
        Compute $\mu_{ij}$ using Equation (1);
        if $\mu_{ij} > \delta$ then
            $label = label \cup \{w_j\}$;
        end
    end
    $R = R \cup \{< r_i, label >\}$;
end
Return $R$;
```

4. Experimental Results

The targeted task has been completed utilising a variety of Python 3.7 modules, including pytorch 1.12.0, numpy 1.12.0, matplotlib 2.2, and networkx 1.11. Initially, the news dataset of about 200,000 news items from various categories published in the United States of America between the years of 2012 and 2018 has been gathered from the website kaggle.com [3]. The efficiency of the proposed algorithm has been evaluated on five different categories of news datasets that have been made considering Crime, Women, Food and drinks, Environment, and College and named DS_1, DS_2, DS_3, DS_4, and DS_5, respectively for future reference in the paper. The crime report dataset used in paper [31] is also considered to evaluate our proposed model and named DS_6 in our paper. This dataset contains news of crime incidents that happened in different places in India, the USA, and the UAE between 2008 to 2016 years.

4.1. Cluster Analysis

After collecting the datasets, the proposed graph-based fuzzy clustering algorithm has been applied to all the datasets for making the clusters. The clusters that have been found by applying our proposed algorithm are overlapping in nature. The description of the dataset with the information about the clusters by applying the proposed clustering algorithm for each dataset has been given in Table 1.

4.2. Performance Evaluation

The comparison of the proposed algorithm's performance with some existing clustering algorithms has been done in this section. Here, some algorithms have been chosen which make overlapping clusters, and some disjoint clustering algorithms also have been selected for making the comparison. After applying the fuzzification technique to the overlapping clusters, we have kept the report in a single cluster based on their degree of membership to get the disjoint clusters. In the case of disjoint clustering algorithms, internal indices are considered, and in the case of overlapping clustering algorithms, overlapping indices are considered.

Table 1. Description of Datasets and Clustering of Reports.

Dataset Name	Number of Reports	Number of Clusters	(Cluster Number, No. of Reports)
DS_1	3405	24	(C1,389), (C2,178), (C3,190), (C4,214), (C5,50), (C6,49), (C7,81), (C8,230), (C9,85), (C10,76), (C11,54), (C12,171), (C13,146), (C14,439), (C15,64), (C16,79), (C17,529), (C18,42), (C19,50), (C20,290), (C21,188), (C22,48), (C23,80), (C24,68)
DS_2	3490	26	(C1,392), (C2,196), (C3,68), (C4,59), (C5,143), (C6,214), (C7,77), (C8,138), (C9,158), (C10,168), (C11,111), (C12,97), (C13,263), (C14,78),(C15,121), (C16,204), (C17,96), (C18,170), (C19,95), (C20,145), (C21,212), (C22,144), (C23,297), (C24,146), (C25,110), (C26,163)
DS_3	6226	32	(C1,442), (C2,158), (C3,269), (C4,249), (C5,543), (C6,234), (C7,377), (C8,638), (C9,245), (C10,185), (C11,371), (C12,503), (C13,63), (C14,358), (C15,110), (C16,170), (C17,240), (C18,350), (C19,295), (C20,145), (C21,232), (C22,344), (C23,297), (C24,146), (C25,118), (C26,87), (C27,206), (C28, 49), (C29, 126), (C30,74) (C31, 78), (C32,88)
DS_4	1323	16	(C1,124), (C2,96), (C3,65), (C4,54), (C5,113), (C6,96), (C7,77), (C8,138), (C9,95), (C10,276), (C11,49), (C12,103), (C13,53), (C14,78), (C15,110), (C16,98)
DS_5	1144	15	(C1,194), (C2,226),(C3,60), (C4,42), (C5,58), (C6,76), (C7,168), (C8,71), (C9,45), (C10,89), (C11,58), (C12,178), (C13,68), (C14,72), (C15,83)
DS_6	31515	33	(C1,516), (C2,396), (C3,1612), (C4,871), (C5,768), (C6,1482), (C7,416), (C8,3480), (C9,2945), (C10,1752), (C11,2551), (C12,790), (C13,3379), (C14,2591), (C15,3374), (C16,2861), (C17,2897), (C18,390), (C19,1682), (C20,1889), (C21,2975), (C22,814), (C23,2552), (C24,3701), (C25,4021), (C26,2896), (C27,3215), (C28,4498), (C29,3002), (C30,3169), (C31,4296), (C32,1289), (C33,4158)

4.2.1. Comparison Using Internal Cluster Indices

The disjoint clustering algorithms that have been considered here for comparison are (i) Modularity Optimization-based Community Detection (MOCD) [32], (ii) Label Propagation Algorithm using Node Influence (LPNI) [33], (iii) Community identification by Smart local Moving Algorithm (CSLMA) [34], (iv) Gini Index-based Community Detection Algorithm (GICDA) [35], and (v) Crime Report Clustering algorithm (CRCA) [36] and the internal cluster validation indices that have been taken into consideration are Dunn's index (DN) [37], Silhouette index (SL) [37], Davies-Bouldin index (DB) [37], Calinski-Harabasz index (CH) [37], Xie-Beni index (XB) [37], and I-index (IN) [37]. These are computed for all the datasets that have been mentioned earlier, and the results are listed in Table 2. The best results are marked by the boldface.

Table 2. Comparison of several clustering techniques using internal indices.

Dataset	Algorithm	SL	DN	DB	XB	CH	IN
DS_1	MOCD	0.72	1.20	0.51	0.42	419	528
	LPNI	0.75	1.37	0.49	0.69	406	523
	CSLMA	0.76	1.54	0.52	0.64	474	584
	GICDA	0.70	1.01	0.53	0.48	458	590
	CRCA	0.80	0.98	0.51	0.39	466	540
	Proposed	**0.81**	**1.94**	**0.42**	**0.34**	**474**	**591**
DS_2	MOCD	0.73	0.92	0.52	0.59	402	410
	LPNI	0.69	1.17	0.50	0.54	407	397
	CSLMA	0.68	1.06	0.49	0.56	399	389
	GICDA	0.63	0.98	0.58	0.52	372	377
	CRCA	0.76	0.93	0.56	0.33	396	467
	Proposed	**0.77**	**1.98**	**0.44**	**0.31**	**409**	**473**
DS_3	MOCD	0.71	0.97	0.49	0.45	411	496
	LPNI	0.68	0.92	0.51	0.47	407	368
	CSLMA	0.69	0.88	0.48	0.41	398	407
	GICDA	0.68	0.81	0.63	0.48	396	412
	CRCA	**0.72**	0.98	0.70	0.37	436	491
	Proposed	**0.72**	**1.16**	**0.42**	**0.36**	**443**	**507**
DS_4	MOCD	0.69	0.91	0.59	0.41	392	589
	LPNI	0.66	0.84	0.60	0.42	387	596
	CSLMA	0.68	0.78	0.58	0.39	396	593
	GICDA	0.64	0.76	0.62	0.41	404	508
	CRCA	0.72	1.07	0.65	0.33	431	579
	Proposed	**0.74**	**1.12**	**0.50**	**0.31**	**457**	**612**
DS_5	MOCD	0.61	0.94	0.71	0.49	205	310
	LPNI	0.64	0.91	0.68	0.41	192	302
	CSLMA	0.62	0.92	0.71	0.48	184	279
	GICDA	0.55	0.82	0.65	0.54	146	304
	CRCA	0.68	**1.10**	0.70	**0.37**	263	**593**
	Proposed	**0.69**	1.06	**0.63**	0.38	**315**	586
DS_6	MOCD	0.77	1.13	0.49	0.45	372	553
	LPNI	0.72	0.97	0.50	0.47	363	594
	CSLMA	0.78	0.95	0.48	0.41	405	579
	GICDA	0.71	0.83	0.57	0.53	368	571
	CRCA	**0.81**	1.19	**0.40**	0.37	436	**687**
	Proposed	**0.81**	**2.91**	**0.40**	**0.33**	**441**	589

In Table 2, it has been seen that for Datasets DS_5, the proposed algorithm does not provide the best index values of the XB index and IN index. It is a college dataset, where our proposed named-entity-based paraphrase vectorization possibly does not work well for the discrimination of the reports. It has also been seen for dataset DS_6 that the best value of IN index is provided by CRCA. But it can be seen clearly from the table that the proposed algorithm produces good index values for other remaining internal indices for all the datasets.

The average value of internal cluster validation indices for various approaches computed for all six datasets, and provided in Table 3 to show overall performance. The table shows that, for all indices except the IN index, the proposed algorithm offers the best result, while the CRCA algorithm does so for the IN index. Figure 5, shows a graphic representation of the average performance for easier visualisation. As it is known, the better outcome is indicated by the higher index values of SL, DN, CH, and IN and the lower index values of XB and DB; the figure demonstrates that, with the exception of Sl and IN, all internal indices receive better values from our Proposed algorithm.

Table 3. Average Internal indices of different algorithms.

Methods	**Internal Cluster Validation Indices**					
	SL	**DN**	**DB**	**XB**	**CH**	**IN**
MOCD	0.70	1.01	0.55	0.46	3.66	4.81
LPNI	0.69	1.03	0.54	0.50	3.6	4.63
CSLMA	0.70	1.02	0.54	0.48	3.76	5.21
GICDA	0.65	0.86	0.59	0.49	3.57	4.61
CRCA	**0.74**	1.04	0.58	**0.36**	4.03	**5.59**
Proposed	0.61	**1.69**	**0.47**	**0.36**	**4.05**	5.03

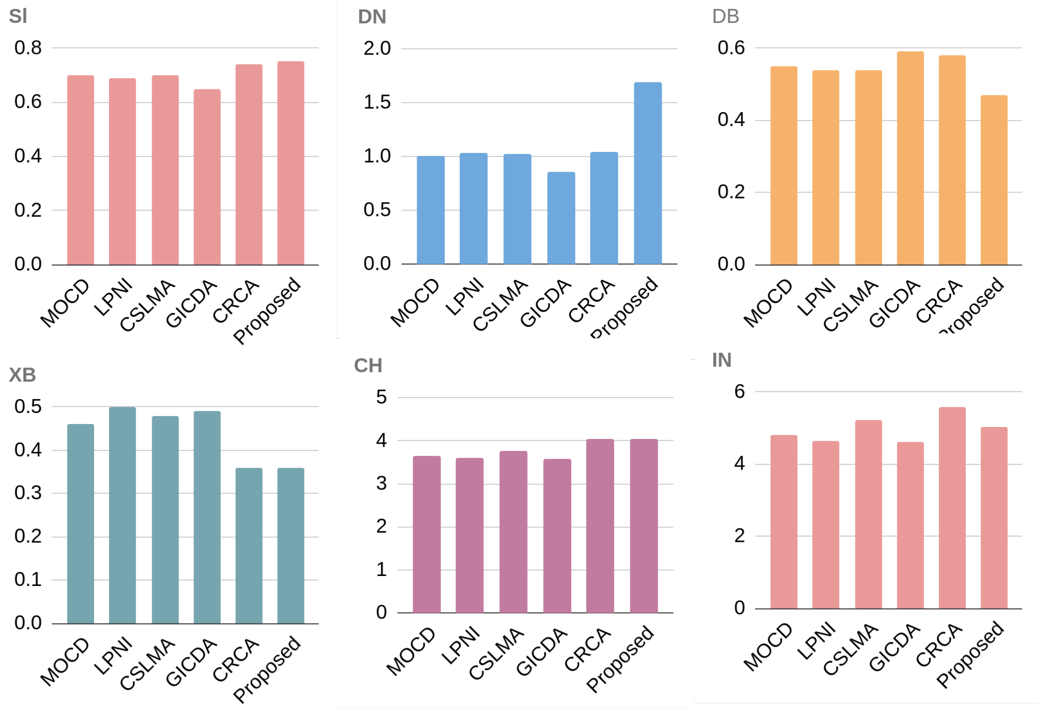

Figure 5. Average Comparison of Internal indices.

4.2.2. Comparison Using Overlapping Cluster Indices

The overlapping algorithms that have been chosen for comparison are (i) Overlapping Community detection by label Propagation (OCLP) [38] (ii) Seed Expansion based Overlapping Community identification (SEOC) [39], (iii) Fuzzy Clustering by Multiobjective optimization(FCMO) [23], (iv) Gini Index-based Community Detection Algorithm (GICDA) [35], and (v) Crime Report Clustering algorithm (CRCA) [36] and the overlapping cluster validation metrics that have been measured for comparing the performance of our proposed algorithm with mentioned overlapping clustering algorithms are (i) Partition

Coefficient (PC) [40], (ii) Partition Entropy (PE) [40], (iii) Dave Index (DI) [41], (iv) Graded distance index (GD) [41] and (v) Kwon Index [41]. In Table 4, these indices have been listed, and the best values are highlighted by the boldface.

Table 4. Comparison of several clustering techniques using overlapping indices.

Dataset	Algorithm	PC	PE	DI	GD	KI
DS_1	OCLP	0.73	0.31	0.71	0.52	8.98
	SEOC	0.70	0.28	0.73	0.50	8.86
	FCMO	0.71	0.32	0.70	0.48	8.49
	GICDA	0.63	0.33	0.52	0.45	9.14
	CRCA	**0.79**	0.29	0.78	0.51	8.94
	Proposed	**0.79**	**0.25**	**0.79**	**0.56**	**8.31**
DS_2	OCLP	0.80	0.35	0.73	0.68	9.38
	SEOC	0.78	0.37	0.74	0.67	9.15
	FCMO	0.76	0.37	0.78	**0.69**	10.08
	GICDA	0.71	0.41	0.73	0.56	10.04
	CRCA	0.81	0.33	**0.80**	0.62	**8.71**
	Proposed	**0.84**	**0.27**	0.79	0.65	9.26
DS_3	OCLP	0.77	0.34	0.71	0.55	9.14
	SEOC	0.73	0.31	0.74	0.58	9.02
	FCMO	0.77	0.35	0.72	0.58	9.10
	GICDA	0.71	0.37	0.68	0.52	9.15
	CRCA	0.80	0.28	**0.81**	0.57	8.72
	Proposed	**0.82**	**0.26**	**0.81**	**0.61**	**8.58**
DS_4	OCLP	0.74	0.23	0.51	0.60	9.12
	SEOC	0.68	0.37	0.68	0.61	9.16
	FCMO	0.54	0.42	0.67	0.58	9.38
	GICDA	0.80	**0.26**	0.51	0.50	9.44
	CRCA	**0.82**	**0.26**	0.80	0.61	8.41
	Proposed	0.80	**0.26**	**0.81**	**0.64**	**8.38**
DS_5	OCLP	0.76	0.25	0.81	0.68	8.25
	SEOC	0.70	0.29	0.84	0.65	8.28
	FCMO	0.71	0.31	0.73	0.65	8.21
	GICDA	0.73	0.38	0.78	0.69	9.52
	CRCA	0.81	0.17	0.86	0.70	**7.41**
	Proposed	**0.82**	**0.22**	**0.88**	**0.72**	8.46
DS_6	OCLP	0.81	0.25	0.78	0.79	7.78
	SEOC	0.81	0.28	0.75	0.83	8.04
	FCMO	0.83	0.34	0.79	0.77	8.14
	GICDA	0.79	0.36	0.83	0.62	8.39
	CRCA	0.85	**0.13**	**0.91**	0.84	7.24
	Proposed	**0.86**	**0.13**	0.85	**0.89**	**7.19**

From Table 4, it can be said that except for the indices value of DI and KI for the dataset DS_2, PC index value for the dataset DS_4, KI index value for the dataset DS_5, and DI index value for DS_6 the proposed RCASRR algorithm does not provide the best value. Hence, it can be said by analysing the index values of Tables 2 and 4 that the proposed algorithm is able to provide better clusters than the other clustering algorithms, which shows the efficiency of the Proposed method.

In Table 5, the average overlapping cluster validation indices of various approaches are shown for all the datasets. Figure 6 provides a graphic depiction of this Table. It shows that, with the exception of the KI index, all indices produce better overlapping index values when using the proposed algorithm. Since the higher index values of PC, DI, and GD and the lower index values of PE and KI produce the best clustering results.

Table 5. Average overlapping indices of different algorithms.

Methods	**Overlapping Cluster Validation Indices**				
	PC	**PE**	**DI**	**GD**	**KI**
OCLP	0.76	0.28	0.70	0.63	8.77
SEOC	0.73	0.31	0.74	0.64	8.75
FCMO	0.73	0.35	0.73	0.62	**8.91**
GICDA	0.72	0.35	0.70	0.55	9.28
CRCA	0.81	0.24	**0.82**	0.64	8.23
Proposed	**0.82**	**0.23**	**0.82**	**0.67**	8.36

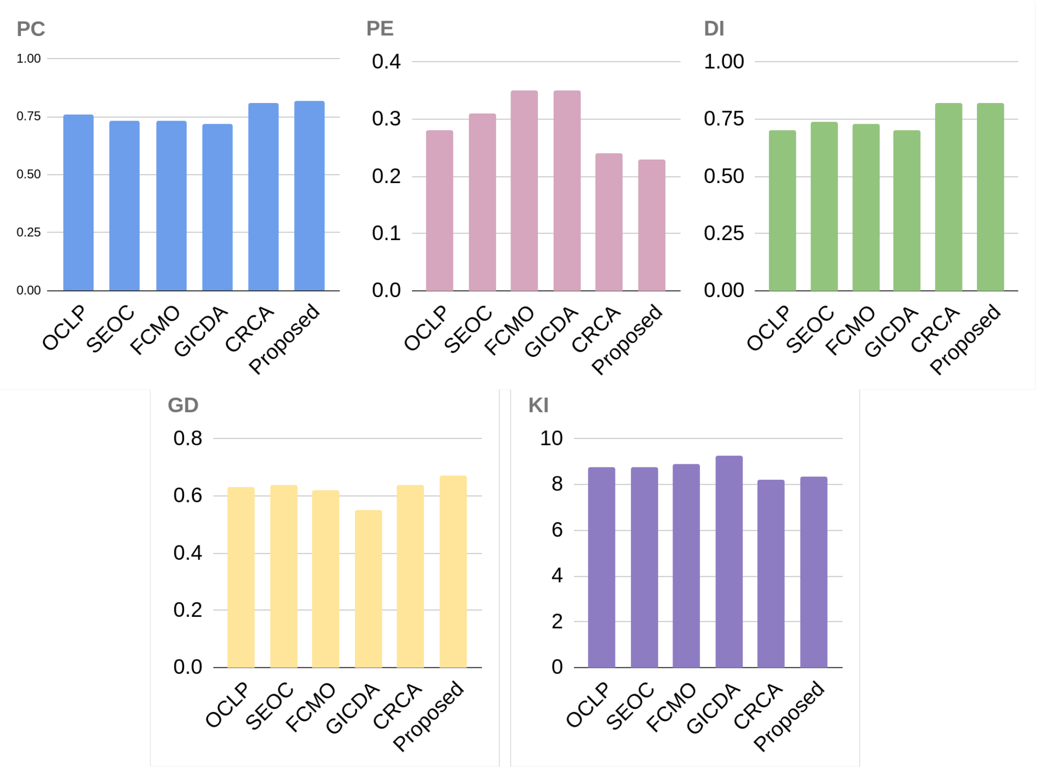

Figure 6. Average Comparison of overlapping indices.

5. Conclusions

The proposed work makes clusters of crime reports according to their context. When the reports are grouped together and the groups are labelled properly, it makes it easier for the police or other law enforcement organisations to evaluate them and recognise the various sorts of offences. This makes it easier to put the required preventive measures in place to stop illegal activity. To locate the overlapped clusters of crime reports, a novel graph-based clustering algorithm with a fuzzy technique has been developed and it also provides a degree of membership to the objects inside the clusters. Other types of datasets have also been employed using the suggested strategy, and it is clear from the results of the experiments that the method works just as well for other applications. As the proposed algorithm makes overlapping clustering, therefore, it is advantageous for applications where objects may belong to multiple classes.

An innovative cluster labelling technique is proposed to understand the nature of the clusters, where each cluster is labelled according to its category, which is a beneficial step for unlabelled datasets. However, the suggested work has two drawbacks, one is that our proposed clustering algorithm can not identify some of the most suitable clusters for a report when a report resides in multiple clusters, and another drawback is that the proposed clustering algorithm can not identify larger outliers. In our future work, we will try to address these problems.

Author Contributions: The contribution of A.P. is designing the model, Programming, and Writing—Original draft preparation. A.K.D. has contributed to Data curation, Conceptualization, Supervision and Validation. The Investigation, Validation and Paper Editing tasks have been carried out by D.P. J.N. has contributed to Visualization, Validation and Reviewing process. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Not applicable.

Data Availability Statement: https://www.kaggle.com/datasets/rmisra/news-category-dataset (accessed on 15 December 2022).

Acknowledgments: This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

1. Saeed, M.Y.; Awais, M.; Talib, R.; Younas, M. Unstructured Text Documents Summarization With Multi-Stage Clustering. *IEEE Access* **2020**, *8*, 212838–212854. [CrossRef]
2. Li, L.; Yang, B.; Zhang, F. Clustering for Complex Structured Data Based on Higher-Order Logic. In Proceedings of the 2008 International Conference on Computer Science and Software Engineering, Wuhan, China, 12–14 December 2008; Volume 4, pp. 390–393. [CrossRef]
3. Misra, R. News category dataset. *ResearchGate* **2018**, *3*, 11429. [CrossRef]
4. Das, P.; Das, A.K. Graph-based clustering of extracted paraphrases for labelling crime reports. *Knowl.-Based Syst.* **2019**, *179*, 55–76. [CrossRef]
5. Khyani, D.; B S, S. An Interpretation of Lemmatization and Stemming in Natural Language Processing. *Shanghai Ligong Daxue Xuebao/J. Univ. Shanghai Sci. Technol.* **2021**, *22*, 350–357.
6. Cer, D.; Yang, Y.; Kong, S.y.; Hua, N.; Limtiaco, N.; John, R.S.; Constant, N.; Guajardo-Cespedes, M.; Yuan, S.; Tar, C.; et al. Universal sentence encoder. *arXiv* **2018**, *arXiv:1803.11175*.
7. Girvan, M.; Newman, M.E. Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA* **2002**, *99*, 7821–7826. [CrossRef] [PubMed]
8. Baadel, S.; Thabtah, F.; Lu, J. Overlapping clustering: A review. In Proceedings of the 2016 SAI Computing Conference (SAI), London, UK, 13–15 July 2016; pp. 233–237.
9. Hauff, B.M.; Deogun, J.S. Parameter tuning for disjoint clusters based on concept lattices with application to location learning. In *Proceedings of the International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing*; Springer: New York, NY, USA, 2007; pp. 232–239.
10. Yang, L.; Cao, X.; Jin, D.; Wang, X.; Meng, D. A unified semi-supervised community detection framework using latent space graph regularization. *IEEE Trans. Cybern.* **2014**, *45*, 2585–2598. [CrossRef] [PubMed]
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, PMLR, Virtual, 13–18 July 2020; pp. 874–883.
12. Taha, K. Disjoint community detection in networks based on the relative association of members. *IEEE Trans. Comput. Soc. Syst.* **2018**, *5*, 493–507. [CrossRef]
13. Ghoshal, A.K.; Das, N.; Das, S. Disjoint and overlapping community detection in small-world networks leveraging mean path length. *IEEE Trans. Comput. Soc. Syst.* **2021**, *9*, 406–418. [CrossRef]
14. Li, M.; Lu, S.; Zhang, L.; Zhang, Y.; Zhang, B. A community detection method for social network based on community embedding. *IEEE Trans. Comput. Soc. Syst.* **2021**, *8*, 308–318. [CrossRef]
15. Whang, J.J.; Gleich, D.F.; Dhillon, I.S. Overlapping community detection using neighborhood-inflated seed expansion. *IEEE Trans. Knowl. Data Eng.* **2016**, *28*, 1272–1284. [CrossRef]
16. Lu, M.; Zhang, Z.; Qu, Z.; Kang, Y. LPANNI: Overlapping community detection using label propagation in large-scale complex networks. *IEEE Trans. Knowl. Data Eng.* **2018**, *31*, 1736–1749. [CrossRef]
17. Rezvani, M.; Liang, W.; Liu, C.; Yu, J.X. Efficient detection of overlapping communities using asymmetric triangle cuts. *IEEE Trans. Knowl. Data Eng.* **2018**, *30*, 2093–2105. [CrossRef]
18. Chakraborty, T.; Kumar, S.; Ganguly, N.; Mukherjee, A.; Bhowmick, S. GenPerm: A unified method for detecting non-overlapping and overlapping communities. *IEEE Trans. Knowl. Data Eng.* **2016**, *28*, 2101–2114. [CrossRef]
19. Van Lierde, H.; Chow, T.W.; Chen, G. Scalable spectral clustering for overlapping community detection in large-scale networks. *IEEE Trans. Knowl. Data Eng.* **2019**, *32*, 754–767. [CrossRef]
20. Su, J.; Havens, T.C. Quadratic program-based modularity maximization for fuzzy community detection in social networks. *IEEE Trans. Fuzzy Syst.* **2014**, *23*, 1356–1371. [CrossRef]

21. Yazdanparast, S.; Havens, T.C.; Jamalabdollahi, M. Soft overlapping community detection in large-scale networks via fast fuzzy modularity maximization. *IEEE Trans. Fuzzy Syst.* **2020**, *29*, 1533–1543. [CrossRef]
22. Biswas, A.; Biswas, B. FuzAg: Fuzzy agglomerative community detection by exploring the notion of self-membership. *IEEE Trans. Fuzzy Syst.* **2018**, *26*, 2568–2577. [CrossRef]
23. Gupta, A.; Datta, S.; Das, S. Fuzzy clustering to identify clusters at different levels of fuzziness: An evolutionary multiobjective optimization approach. *IEEE Trans. Cybern.* **2019**, *51*, 2601–2611. [CrossRef] [PubMed]
24. Madhuri, V.; Bazighifan, O.; Ali, A.H.; El-Mesady, A. On Fuzzy-Simply Connected Spaces in Fuzzy-Homotopy. *J. Funct. Spaces* **2022**, *2022*, 9926963. [CrossRef]
25. PM, D.; PB, R.; Cletus, N.; Joy, P. Fuzzy Hypergraph Modeling, Analysis and Prediction of Crimes. *Int. J. Comput. Digit. Syst.* **2022**, *11*, 649–661.
26. Lee, S.J.; Jiang, J.Y. Multilabel text categorization based on fuzzy relevance clustering. *IEEE Trans. Fuzzy Syst.* **2013**, *22*, 1457–1471. [CrossRef]
27. Meng, T.; Cai, L.; He, T.; Chen, L.; Deng, Z. Local higher-order community detection based on fuzzy membership functions. *IEEE Access* **2019**, *7*, 128510–128525. [CrossRef]
28. Liu, Z.; Barahona, M. Graph-based data clustering via multiscale community detection. *Appl. Netw. Sci.* **2020**, *5*, 3. [CrossRef]
29. Loper, E.; Bird, S. Nltk: The natural language toolkit. *arXiv* **2002**, arXiv:0205028.
30. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
31. Das, A.K.; Das, P. Graph based ensemble classification for crime report prediction. *Appl. Soft Comput.* **2022**, *125*, 109–215. [CrossRef]
32. Blondel, V.D.; Guillaume, J.L.; Lambiotte, R.; Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* **2008**, *2008*, P10008. [CrossRef]
33. Xing, Y.; Meng, F.; Zhou, Y.; Zhu, M.; Shi, M.; Sun, G. A node influence based label propagation algorithm for community detection in networks. *Sci. World J.* **2014**, *2014*, 627581. [CrossRef] [PubMed]
34. Waltman, L.; Van Eck, N.J. A smart local moving algorithm for large-scale modularity-based community detection. *Eur. Phys. J. B* **2013**, *86*, 471. [CrossRef]
35. Goswami, S.; Murthy, C.; Das, A.K. Sparsity measure of a network graph: Gini index. *Inf. Sci.* **2018**, *462*, 16–39. [CrossRef]
36. Das, A.; Nayak, J.; Naik, B.; Ghosh, U. Generation of overlapping clusters constructing suitable graph for crime report analysis. *Future Gener. Comput. Syst.* **2021**, *118*, 339–357. [CrossRef]
37. Liu, Y.; Li, Z.; Xiong, H.; Gao, X.; Wu, J. Understanding of internal clustering validation measures. In Proceedings of the 2010 IEEE International Conference on Data Mining, IEEE, Sydney, NSW, Australia, 13–17 December 2010; pp. 911–916.
38. Dong, S. Improved label propagation algorithm for overlapping community detection. *Computing* **2020**, *102*, 2185–2198. [CrossRef]
39. McDaid, A.; Hurley, N. Detecting highly overlapping communities with model-based overlapping seed expansion. In Proceedings of the 2010 International Conference on Advances in Social Networks Analysis and Mining, IEEE, Odense, Denmark, 9–11 August 2010; pp. 112–119.
40. Dave, R.N. Validating fuzzy partitions obtained through c-shells clustering. *Pattern Recognit. Lett.* **1996**, *17*, 613–623. [CrossRef]
41. Joopudi, S.; Rathi, S.S.; Narasimhan, S.; Rengaswamy, R. A new cluster validity index for fuzzy clustering. *IFAC Proc. Vol.* **2013**, *46*, 325–330. [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.