A Novel Approach of Clustering Documents: Minimizing Computational Complexities in Accessing Database Systems

Mohammed Alghobiri
Department of Management Information Systems
King Khalid University, Saudi Arabia
maalghobiri@kku.edu.sa

Khalid Mohiuddin
Department of Management Information Systems
King Khalid University, Saudi Arabia
kalden@kku.edu.sa

Mohammed Abdul Khaleel
Department of Computer Science
King Khalid University, Saudi Arabia
mkhlel@kku.edu.sa

Mohammad Islam
Department of Management Information Systems
King Khalid University, Saudi Arabia
maleslam@kku.edu.sa

Samreen Shahwar
Department of Information Systems
King Khalid University, Saudi Arabia
smrin@kku.edu.sa

Osman Nasr
Department of Management Information Systems
King Khalid University, Saudi Arabia
oanassr@kku.edu.sa

Abstract: *This study addresses the real-time issue of managing an academic program's documents in a university environment. In practice, document classification from a corpus is challenging when the dataset size is large, and the complexity increases if to meet some specific document management requirements. This study presents a practical approach to grouping documents based on a content similarity measure. The approach analyzes the state-of-the-art clustering algorithms performance, considers Hamiltonian graph properties and a distance function. The distance function measures (1) the content similarity between the documents and (2) the distances between the produced clusters. The proposed algorithm improves clusters' quality by applying Hamiltonian graph properties. One of the significant characteristics of the proposed function is that it determines document types from the corpus. Hence, this does not require the initial assumption of cluster number before the algorithm execution. This approach omits the arbitrary primordial option of k-centroids of the k-means algorithm, reduces computational complexities, and overcomes some limitations of commonly practicing clustering algorithms. The proposed approach enables an effective way of document organization opportunities to the information systems developers when designing document management systems.*

Keywords: *Clustering algorithms, document categorization, document clustering, hamiltonian graph, similarity measure.*

Received July 11, 2020; accepted February 21, 2021
https://doi.org/10.34028/iajit/19/4/6

1. Introduction

Over the decades, most academic institutions struggle to manage the vast volume of documents. These documents are generated during the delivery of an academic program in higher education-course assessments, students' performance samples, and other academic documents. Document heterogeneity is the primary concern while storing and retrieving the documents from the corpus [5]. Several document management systems are developed that support organization [21]. Some of these systems are organization-specific [9]. Most of these systems are developed for business organizations, and less focus has been given to educational institutions. Generally, these systems classify based on their data models, i.e., document-oriented, key-value, wide-column, and graph-based [4]. Indeed, documents-oriented databases or just document stores have to be a flexible model, easy to maintain, and potential enough in responding to users' random queries, and rich in application programming interfaces [11].

The increasing volume of documents in an educational institution is quite apparent but managing this corpus is challenging. It is more difficult to structure, organize, and retrieve these documents when needed. Using document management systems, organizations manage documents by clustering the documents based on the content from the corpora of documents [11]. The clusters of documents own a different structure that supports both flexibility and evaluation. Notably, the document management systems are schema-less that encourages developers to design such systems by avoiding relational model limitations.

Many organizations own document stores, and some document stores now classify as NoSQL databases [11]. A document store develops based on organizational needs, and no panacea exists that covers an organizational requirement [5]. Therefore, selecting a database is more challenging, fulfilling the organizations' specific requirements from the systems' plethora. Since organizations' requirements keep changing, they need to incorporate the required changes into the document stores. Therefore, developers should consider different document management techniques by providing a panacea for organizations [1].

This study compares some existing algorithms and the limitations (see Tables 1 and 2) for documents store and proposes an alternative documents clustering approach. It introduces an algorithm that clusters the documents based on contents of Substantial Resemblance (SR) in the corpus [1]. It determines the content similarity between the documents and considers binary-valued distance function (see Equations (1) and (2)) to identify documents' similarity. Remarkably, the study's approach applies Hamiltonian Graph properties for grouping similar pattern documents.

Indeed, academic institutions generate tons of documents in every academic cycle. Section 5.1. describes the sources of document generation and the formats of the document instances. The proposed approach applies to an academic data set to cluster the documents considering the substantial content resemblance. The result of clustering produces the clusters of similar documents [1], e.g., course files cluster.

Table 1. Most used clustering algorithms and their advantages.

Clustering algorithms	**Reference**	**Advantages**
1- *k*-mean clustering (partitioned), *k* is the desired number of clusters	[17]	Low computational complexity
2-Buckshot clustering: combination of *k*-mean and Hierarchical clustering	[19]	Low computational complexity
3-Hierarchical clustering	[19]	Doesn't require prior knowledge of number of clusters
4-Hybrid document clustering	[1]	It can automatically identify the number of clusters
5-Dynamic *k*-nearest neighbor	[7]	It shows good result for some image data
6-Time efficient *k*-mean algorithm	[3]	The number of computations can be significantly reduced
7-Bisecting *k*-mean algorithm	[5]	Optimize a desired clustering criterion function
8-Spectral clustering algorithm	[18]	Good for number of challenging clustering problems
9-Non-Negative Matrix Fi	[13, 14]	Sensitive with the initialization of one or both NMF factors
10-Concept Factorization (CF) based document clustering	[24]	The non-negative solution minimizes the reconstruction error of the data points.
11-Simple active clustering algorithm	[6]	Produces multiple clusters of the same data set user interest
12-Soft constraints algorithm	[22]	It improves the performance when the amount of prior knowledge is limited.
13-A semi-supervised NMF method	[13]	It is good for some real time corpora
14-MST based Clustering algorithm	[10]	Capable of detecting clusters with irregular boundaries (on *k*-partition)

Table 2. Identified clustering algorithms limitations.

#s	**Clustering algorithms limitations**
1	It requires prior knowledge of the number of clusters (seed points).
2	Here document corpora may be huge in size with high density, difficult to estimate the number of clusters.
3	It requires a prior assumption of the number of iterations that leads to complexity for a large data set.
4	It is difficult to select a valid *k*-value for an unknown text data set.
5	There is no universally acceptable way of identifying initial seed points for clustering the documents.
6	In this, the resulting clusters may not be accurate and lead to producing inappropriate clusters.
7	Here it does not represent accurately the whole data set during the clustering process.
8	In this the *k*-mean clustering aspect has not been considered precisely.
9	Stopping criteria requires terminating condition& it is difficult to identify such conditions for large data sets.
10	Merging clusters require prior knowledge of desired number of clusters that is unpredictable. and difficult to determine the valid *k* for text data set
11	Parameter sensitivity i.e. σ [10], wrong value of σ may highly degrade the quality of clusters/ difficult to select proper value of σ for documents collection and for high dimensional data set.
12	Here the random initialization of data value can produce inaccurate clusters.
13	In this, the restriction on both similar and dissimilar continents will lead to constrained optimization problems.
14	For the clustering, If the initial centroid correctly does not select, then it produces inaccurate clusters.

The study's approach adopts a partition-based clustering technique and proposes an algorithm that clusters documents based on a Hamiltonian path [11] by traversing XY- plane in the graph. This method traverses each node only once to find a Hamiltonian path in the graph (see section 4.1). In the graph dotted lines connect nodes (documents) on the XY plane; either horizontally or vertically, following two conditions:

1. They should not intersect any path while traversing from one node to another
2. Traversing should not be repeated the same path in any case. The traversing between the nodes follows the Hamiltonian path properties for creating documents' clusters.

The presented study addresses a real-time requirement of managing documents of an academic program in higher education. Remarkably, the proposed approach does not need an initial seed point, a number of clusters, and *k*-value, unlike *k*-mean, Buckshot, and Time-efficient *k*-algorithms [1, 3] (see Tables 1 and 2). Furthermore, this approach should be a significant contribution to application developers when developing document management systems. This approach can apply to any content documents and different data sets.

1.1. Contributions and Outline

First, this study addresses the real-time requirements of managing documents in a university environment. Second, it presents a careful review of the state-of-the-art clustering algorithms in section 2.1. Section 2.3. describes creating clusters by applying spanning-tree properties. Section 2.5. Presents a graph-based clustering algorithm, third, it introduces a substantial resemblance technique using the three cases in section 3. Section 3.1. Describes the measuring of content similarities between clusters. Furth, the proposed Hamiltonian Graph-based clustering algorithm discussed in section 4.1. And Figures 3 and 4 show the graphical representation of the study's approach. The rest of the paper completes by section 2, which discusses the motivation and related work. Section 5 discusses the result and the approach impact, and Section 6concludes.

2. Background

Most of the existing document clustering methods depend on two data models, hierarchical and partitioned [1, 8]. These methods classify a corpus of documents into subsets based on similarities. Some analytical methods apply to the corpus and distinguish the classified subsets into clusters [23]. These clusters are measured considering the intended outcomes, such as either content similarity or dissimilarity. There are some corpus-based measures for the classification of a data set (document store). The classification measure on text data supports similar objects' grouping [5] and continues with application-specific areas like document clustering.

More significance has been given to the two document clustering techniques in the literature, hierarchical and partitioned [1]. Many researchers consider other cluster analysis methods in the modern approach, such as Graph and spectral methods, density-based methods, grid-based clustering, model-based clustering, potential (Kernel) Function Methods, and other cluster analysis methods [23].

Hierarchical clustering creates a tree of clusters, and each cluster is considered a combination of clusters in the hierarchy to the next lower level [5]. Agglomerative and divisive are the two branches of hierarchical clustering techniques shown in Figure 1. In the Agglomerative Hierarchical Clustering (AHC) method [1], each document in the corpus is considered an individual cluster [13], compares content similarity with another cluster, and merges it into a new cluster. This process repeats at every step for all the corpus clusters until an assumed halting-condition is satisfied. Whereas in the divisive method, the whole corpus is considered a single cluster (corpus-cluster), and a termination condition is assumed. The method splits the corpus cluster into smaller clusters [5]. The process further applies to the smaller clusters until the assumed termination condition is satisfied. Usually, the least similarity cluster chooses first for splitting [20]. Several AHC algorithms did propose considering a halting condition, but there is no wide acceptance for the standard halting conditions. This process leads to meaningless clustering, and the resulting cluster fails to fulfill users' classification measures.

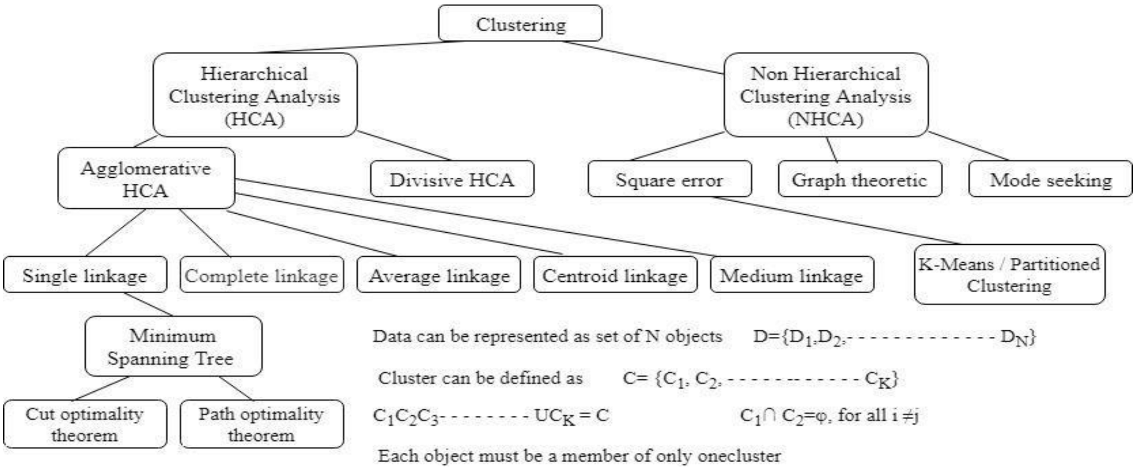
Clustering
Hierarchical Clustering Analysis (HCA)
Non Hierarchical Clustering Analysis (NHCA)
Agglomerative HCA
Divisive HCA
Square error
Graph theoretic
Mode seeking
Single linkage
Complete linkage
Average linkage
Centroid linkage
Medium linkage
K-Means / Partitioned Clustering
Minimum Spanning Tree
Cut optimality theorem
Path optimality theorem
Data can be represented as set of N objects $D=\{D_1,D_2,- - - - - - - - - - - - - D_N\}$
Cluster can be defined as $C= \{C_1, C_2, - - - - - - - - - - - - C_K\}$
$C_1C_2C_3$- - - - - - - - $UC_K = C$
$C_1 \cap C_2=\varphi$, for all $i \neq j$
Each object must be a member of only onecluster

Figure 1. Practicing clustering techniques commonly used.

AHC technique further represents three hierarchical variations-single-link, complete-link, and group-average for document clustering [1]. In a single link, the content similarity between a pair of clusters measures for most similarity where each document is considered a cluster. The complete-link method calculates the content similarity between a pair of clusters of the least similar documents, where each document is in a cluster of its own [1]. If the clusters have the least average similarity, then the group-average method applies to merge such clusters. In Graph-based methods, the document's corpus identifies as the set of vertices of an assumed graph ‘G’. A cluster can be a sub graph ‘Gs’ whose vertices do not communicate with the outside vertices.

2.1. Approach Significance

It investigates clustering algorithms, advantages, limitations, and techniques. It discusses graph-based spanning tree algorithms to illustrate the possible way of creating clusters. It describes Bachelor and Wilkin’s algorithm to measure the content similarity between the clusters using an undirected weighted graph. Table 3 includes the comparison of the limitations of existing algorithms and few of them has been addressed using the study’s approach. Table 4 shows an example that demonstrates how the *k*-means algorithm applies to document corpus, and Table 5 shows a distance measuring method using Square Euclidean distance. Algorithm (4) show a graph-based clustering technique using transitive closure property to demonstrate the logical relation between graph edges and vertexes for creating clusters. Section 3 describes the substantial resemblance between the documents using the cosine function.

Table 3. Algorithms’ limitations comparison using Table 2.

Clustering Algorithm	1	2	3	4	5	6	7	8	9	10	11	12	13	14
k-mean clustering	✓	✓	✓	✓	✓	X	X	X	X	X	X	X	X	X
Buckshot clustering	✓	✓	✓	✓	✓	✓	✓	✓	X	X	X	X	X	X
Hierarchical clustering	X	X	X	X	X	X	X	X	✓	X	X	X	X	X
Hybrid document clustering	✓	✓	✓	✓	✓	X	X	X	✓	X	X	X	X	X
Dynamic *k*-Nearest Neighbor	✓	✓	✓	✓	✓	X	X	X	X	✓	X	X	X	X
Time efficient *k*-mean algorithm	✓	✓	✓	✓	✓	X	X	✓	X	X	X	X	X	X
Bisecting *k*-mean algorithm	X	X	X	X	X	✓	X	✓	X	X	X	X	X	X
Spectral clustering algorithm	X	X	X	X	X	X	X	✓	X	X	✓	X	X	X
Non-Negative Matrix Factorization (NMF)	X	✓	X	X	X	X	X	X	X	X	X	✓	X	X
Concept Factorization clustering	✓	✓	✓	✓	✓	✓	X	X	X	X	X	X	X	X
Simple active clustering algorithm	X	X	X	X	X	X	X	X	X	X	X	X	✓	X
Soft constraints algorithm	X	X	X	X	X	X	X	X	X	X	X	X	X	✓
A semi-supervised NMF method	X	X	X	X	X	X	X	X	X	X	X	X	X	X
MST based clustering algorithm	✓	✓	✓	✓	✓	✓	X	X	X	X	X	X	X	X

Table 4. Node values from the graph Figure 2.

Example 1, validating the algorithms applying the graph											
Clusters	A	B	C	D	E	F	G	H	I	J	K
Variables (x_1, x_2)	1, 1	1, 2	3, 2	3, 5	6, 6	6, 7	7, 6	7, 7	7, 1	8, 2	8, 1

Table 5. Distance between clusters derived from the graph G.

Clusters	A	B	C	D	E	F	G	H	I	J	K
A	0	1.0	2.2	2.8	7.0	7.8	7.8	8.4	6.0	7.0	8.0
B	1.0	0	2.0	2.2	6.4	7.0	7.0	7.8	6.0	7.0	8.0
C	2.2	2.0	0	1.0	5.0	4.9	5.6	6.4	4.1	5.0	6.0
D	2.8	2.2	1.0	0	4.2	5.0	5.0	5.6	4.5	5.0	6.3
E	7.0	6.4	5.0	4.2	1.0	1.0	1.41	1.41	5.0	4.5	5.4
F	7.8	7.0	4.89	5.0	1.0	0	1.41	1.0	6.0	5.4	6.3
G	7.8	7.0	5.6	5.0	1.41	1.41	0	1.0	5.0	4.1	5.0
H	8.4	7.8	6.4	5.6	1.41	1.0	1.0	0	6.0	5.0	6.0
I	6.0	6.0	4.1	4.5	5.0	6.0	5.0	6.0	0	1.41	1.0
J	7.0	7.0	5.0	5.0	4.5	5.4	4.1	5.0	1.41	0	1.0
K	8.0	8.0	6.0	6.3	5.4	6.3	5.0	6.0	1.0	1.0	0

2.2. Clustering Algorithms in Review

We have conducted a literature review across multiple resources to identify the most practicing document management systems. We found several systems that establish the study’s approach [13]. These systems have demonstrated in the following sections, cited in the body text, and listed in the reference section. Here the objective is to investigate the suitability of such systems for clustering higher education academic documents. Further, to identify the best suitable technique for the academic document store and apply it to managing the academic documents [2].

2.3. Clustering Algorithms

Spanning tree clustering Algorithms (1) and (2) are used to create an undirected weighted graph shown in Figure 2. The spanning tree algorithms and Tables 6 and 4 have assumed to draw the graph ‘G.’ Applying these algorithms and spanning tree properties with total lesser weight have assumed to obtain at least one different edge*e*= (*u*,*v*) of the G. If any edge is lesser than ‘e’ that edge is assumed in the output tree. If the graph G forms a connected graph, then the final output must be connected [23].

Table 6. Distance between vertices in the graph.

8	C-D	E-F	E-G	F-H	G-H	E-H	F-G	B-C	A-C	B-D	A-D	D-E	C-F
1	1	1	1	1	1	1.41	1.41	2	2.23	2.23	2.82	4.24	4.89
C-E	D-F	D-G	C-G	D-H	B-E	C-H	A-E	B-F	B-G	A-F	A-G	B-H	A-H
5	5	5	5.65	5.65	6.4	6.4	7.07	7.07	7.07	7.81	7.81	7.81	8.48

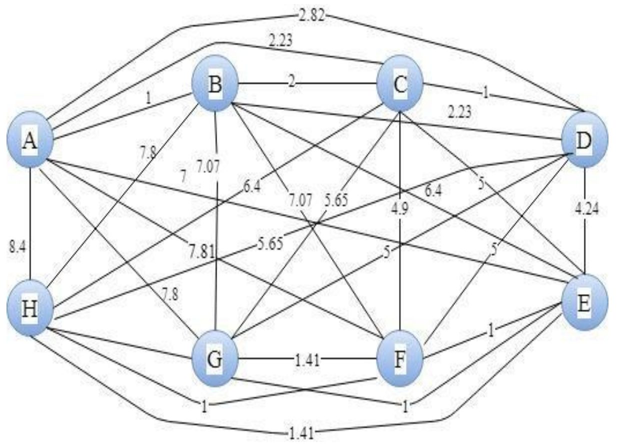

Figure 2. Undirected weighted graph used for measuring the distance (content similarity) between the documents.

Algorithm 1: Minimum Spanning Tree, or Cut Optimality condition

```
Every tree arc  $ij \in T^*$
$C_{ij} \le C_{kl}$, $\forall$ $kl, i, j$ edges cut from the graph and adding in $T^*$
Void cut optimality (Graph G, Tree T)
{int i; vertex u, v;
 Read graph (G, T)
/* read graph from adjacency list */
for (i=0; i<N; i++)            {
T[i]. known= false.
T[i]. dist.= infinity;
T[i]. path = not a vertex;  }
T[start]. dist. = 0
for (;   ; ){
u = smallest unknown distance vertex;
if (u= = not a vertex) break;
T[u]. known = True;
for each v adjacent to u
If (! T[v]. known) {
T[v]. dist. = min [T(v). dist., $C_{u,v}$]
T[v]. path = u;}}}
```

Algorithm 2: minimum spanning tree or path optimality condition

```
Every non-tree arc $kl$ (not in tree)
$C_{ij} \le C_{kl}$ , $ij \in T^*$
[A] = 0, array initialization
for each $v \in G.V$; (v is vertex of a tree)
SET (v), (creating a vertex of a tree)
for each (u, v) ordered by weight (u, v) increasing;
if SET (u) ≠ SET (v);
                    $A = A \cup (u, v)$
UNION (u, v)
Return A.
```

2.4. Creating Cluster Applying Spanning Tree

The minimum spanning tree technique is an effective method for creating clusters [10]. Generally, a threshold value 'θ' is defined for creating a cluster.

Assume, if d(θ)≥2, then from Table 7, nodes A and B are discarded, and a cluster 1 (A, B) is created. Similarly, the process applies for the other nodes of G, and clusters 2 and 3 are created. Further, it continues until *K* clusters have been created.

Table 7. Clusters creating techniques used in this study's approach.

Vertex	known	d_n(distance between vertex)	P_n(path from vertex)	Clusters	
A	~~0~~ 1	0	0		
B	~~0~~ 1	~~∞~~ 1	~~0~~A	I {A, B}	Threshold (θ) ≥ 2
C	~~0~~ 1	~~∞~~ ~~2.23~~ 2	~~0~~ ~~A~~ B		
D	~~0~~ 1	~~∞~~ ~~2.82~~ ~~2.21~~ 1	~~0~~ ~~A~~ ~~B~~ C	II {C,D}	
E	~~0~~ 1	~~∞~~ ~~7.0~~ ~~6.4~~ ~~5~~ 4.2	~~0~~ ~~A~~ ~~B~~ ~~C~~ D		
F	~~0~~ 1	~~∞~~ ~~7.8~~ ~~7~~ ~~4.9~~ 1	~~0~~ ~~A~~ ~~B~~ ~~C~~ E		
G	~~0~~ 1	~~∞~~ ~~7.8~~ ~~7~~ ~~5.6~~ ~~5~~ 1	~~0~~ ~~A~~ ~~B~~ C ~~D~~ E	III {E,F,G,H}	
H	~~0~~ 1	~~∞~~ ~~8.4~~ ~~7.8~~ ~~6.4~~ ~~5.6~~ ~~1.41~~ 1.0	~~0~~ ~~A~~ ~~B~~ ~~C~~ ~~D~~ E F		

2.5. Bachelor and Wilkin's Algorithm

The study's approach considers Bachelor and Wilkin's algorithm to establish the distance (content similarity) measuring process. Generally, the outcome of the clustering algorithm results from the order of objects used in the clustering or the other way the object patterns affect the intended outcomes of the algorithms [2]. In Bachelor and Wilkin's algorithm, to generate the first cluster, the object pattern is randomly selected, and the second cluster pattern derives from the first cluster [1].

Algorithm 3: Bachelor and Wilkins' algorithm

```
It is a sequential process
Number of features vectors (N)
$X_1, X_2 \ldots\ldots X_N$
$X_1$ and $X_2$are cluster center for i = 2
Find the distance to all other clusters from ($X_1$, $X_2$) clusters
$X_j \rightarrow dis\ (X_j, X_{1)},\ dis\ (X_j, X_2)$
$max_j\{\{min\ \{d\ (X_j, X_{1)},\ d\ (X_j, X_2)\}\}$
$> \tau d\ (X_1, X_2)$corresponding $X_j$will be consider for clustering & continue.
```

Similarly, new clusters are created from the pattern selected from the actual pattern and keep on assigning the pattern to the nearest cluster.

Table 8 describes the measured distances of the clusters using a matrix comparison of Table 5. Here, assuming a matrix of clusters with its center ‘A’ and measuring clusters’ distances from A. Here finding the distance from A, H is at a maximum distance, shifting to another cluster, i.e., dissimilar content. Similarly, finding the maximum distances from cluster H to the rest of the clusters and so on. Further, comparing the distances between two clusters A and H. Encircling the minimum distance value of the two clusters and similarly for K and so on. During this process, clusters should be predefined, and the measuring process keeps on. A study [15] has used the same technique for creating clusters, such as 1 (A, B, C, D), 2 (E, F, G, H), and3 (I, J, K).

Here, this study defines an NHC *k*-means clustering algorithm to show the document clustering process, establishing the proposed approach. It is one of the powerful clustering algorithms widely used [15] for creating clusters. Usually, this algorithm applies to applications, such as healthcare fraud detection [12], geostatic, anomaly detection [2], segmentation in news articles, and customer purchasing behaviors. However, it demonstrates some *limitations*, such as identifying *k*-parameter from an extensive data set and finding the optimal solution is computationally expensive. Indeed, the presented approach eliminates the initialization of the *k*-value [18]. Table 9 describes the process of measuring Square Euclidean (SE) distance using documents’ centroids values.

$$d^2(A,(A,B)) = (5-2)^2 + (3-2)^2 = 10 \text{ (SE)}$$

Table 8. Clusters’ distance matrix comparison of the measured distances.

First cluster A center	B	C	D	E	F	G	H	I	J	K
	1.0	2.2	2.8	7.0	7.8	7.8	8.4	~~6.0~~	7.0	8.0
H	7.8	6.4	5.6	1.41	1.0	1.0	X	6.0	~~5.0~~	6.3
K	8.0	6.0	6.32	5.8	6.7	5.3	X	2.0	1.41	X

Table 9. Documents and variables considered measuring square euclidean distance.

Set of documents	**(A, B)**	**(C, D)**	**Comments**
Documents’ centroids (X_1, X_2)	(2, 2)	(-1, -2)	
Square Euclidean distance	d^2 (A (A, B)) = 10 d^2 (A (C, D)) = 61		10<61, A is already in Group (A, B) No re- assignment of A is required.
	d^2 (B (A, B)) = 10 d^2 (B (C, D)) = 9		10>9, B is not a group of (C, D) B will enter the (C, D)
Two clusters (k=2) are obtained	I (A) II (B, C, D)		Further no partitioned requires

Table 10 describes an example to apply the *k*-means algorithm to the document’s corpus where A, B, C, D are the different sets of documents, and $X_1$and X_2 are the variables. The values for the variables assumed and explained by the following method:

Let *k*=2, is an arbitrary partition in the corpus. Initially, the corpus is partitioned into two sets of documents (A, B) and (C, D). To find a centroid of the partitioned documents, i.e.,$(\underline{X_1}, \underline{X_2})$ and the corresponding values (2, 2) and (-1, -2) are achieved using geometrical properties, shown in Table 10.

Table 10. Example that considers documents and its variables.

Documents	**A**	**B**	**C**	**D**
Different variables (X_1, X_2)	(5,3)	(-1, 1)	(1, -2)	(-3, -2)
Converse of (X_1, X_2)	(5-3)/2, (3+1)/2 = (2,2)		(1-3)/2, (-2-2)/2= (-1, -2)	

Similarly, A (5, 3) & (C, D) (-1, -2) => d2 (A (C, D)) = 61, B (-1, 1) and (A, B) (2, 2) => d2 (B (A, B)) = 10 and similarly d2 (B (C, D)) = 9

Importantly, the study’s approach discusses several potential clustering algorithms and methods [16] to demonstrate the documents’ clustering process and corelate to the study’s approach.

2.6. Graph Based Clustering

A graph-based algorithm that runs over on example 1, i.e., Table 4. A similarity matrix is created based on the *k*-nearest neighbor graph [7]. Generally, the graph-based algorithm takes more than one graph as input where the graph vertices represent the documents [2]. The transitive closure property applies to identify the possible paths (content similarity) between the documents. This method is used to find the most and least similarity contents in a document corpus by determining the possible paths in the graph - measuring the distances between the documents.

The presented approach also considers the AHC algorithm to merge the most similar sub-clusters of documents based on the cluster’s interconnectivity and closeness [2]. During the measuring process, the proposed algorithm experiences the same limitations as agglomerative hierarchical clustering algorithms.

Algorithm 4: Graph based clustering (Transitive closure property)

```
G (V, E)
N $\longrightarrow$ number of nodes in the graph
  N X N Matrix
```

$$i,j \begin{cases} 1, & v_i, v_j \; are\ connected\ (i,e\ feature \\ & vector\ i\ and\ j\ are\ similar \\ & 0, otherwise \end{cases}$$

$$k \rightarrow s_{ij} = 1\ (row\ with\ maximum\ number\ of\ 1)$$

j^{th} feature vector similar to i^{th} vector and k^{th} feature vector similar to j^{th} vector

According to transitive closure property k^{th} feature vector similar to i^{th} feature vector.

3. Substantial Resemblance Approach

The study's approach introduces a Substantial Resemblance (SR) technique to find the content similarity between documents [1]. The documents' content similarities are determined and measured based on binary values. Let considers a Venn diagram that represents the content similarity between two document clusters (C_X, C_Y) using logical mapping. Assume two documents' content is sufficiently similar, and *i* is the document of the cluster C_X maps with j of C_Y to determine the most content similarity. If it exists between the documents, it is represented by a binary value 0 or else by 1 shown in Equation (1). Again, *i* compares to other document *k* of cluster C_Y, if i > j and j>k, then i > k for all the documents in cluster C_Y. It is assumed that document cluster C contains a finite number of documents throughout the clustering process.

$$dis(d_i,d_j) + dis(d_j,d_k) - dis(d_i,d_k) \leq 1 \tag{1}$$

$$dis(d_i,d_j) \forall i,j = \begin{cases} 1\ if\ \rho(d_i,d_j) \leq \theta & \theta \in (0,1) \\ 0 & otherwise \end{cases} \tag{2}$$

$$\rho(d_i,d_j) = cos(\overrightarrow{d_\iota}, \overrightarrow{d_J}) \tag{3}$$

Where ρ is the similarity measure between documents d_i and d_j, i.e., $\rho(d_i,d_j)$. In Equation (2) θ considers the threshold value on the content similarity and restricts the lower similarity values. Let assume that d_i and d_j have cosine similarity 0.52, and d_j and d_k have cosine similarity 0.44, & Θ=0.1.

$if\ cos(\overrightarrow{d_\iota}, \overrightarrow{d_J}) = 1$, *documents are dissimilar, from (2).*
$else$

$if\ cos(\overrightarrow{d_\iota}, \overrightarrow{d_J}) > \theta$, *distance is* 0, *contents are similar*

$$Method\ of\ estimation\ of\ \theta \begin{cases} \rho > \theta, similarity\ exist \\ \rho \leq \theta, dissimilarity\ exist \end{cases}$$

[5]

$$These\ two\ distance\ have\ same\ values \begin{cases} dis(d_i,d_j) = 0 \\ dis(d_j,d_k) = 0 \end{cases}$$

Equation (4) is used to measure the content similarity of documents between two clusters (C_X, C_Y)

$$l_{ij} = \sum_{k=1}^{N} |\, dis(d_i,d_k) - dis(d_j,d_k)| \tag{4}$$

Here, 0 represents the document content similarity and "+ve" represents dissimilarity of documents between the clusters (C_X, C_Y). Here d_k (*k*= 1, 2…*N*) and *N* represents the set of documents in C*Y*.
Assigning *k=1*,

$l_{ij} = \left| dis(d_i,d_1) - dis(d_j,d_1) \right|$, *from eq. (4)*
$Contents\ are\ similar,\ dis(d_i,d_1) = dis(d_j,d_1) = 0$
$l_{ij} = 0\ (consider\ as\ binary\ value\ 0)$
Assigning *k=2*,
$l_{ij} = \left| dis(d_i,d_2) - dis(d_j,d_2) \right|$
$Let\ dis(d_i,d_2) \neq 0, i.e\ some + ve\ value\ x$
$dis(d_j,d_2) \neq 0, i.e\ some + ve\ value\ y$
$Contents\ are\ dissimilar,\ dis(d_i,d_2) \neq dis(d_j,d_2)$
$|x - y| = +ve\ (consider\ as\ binary\ value\ 1)$

For the substantial resemblance between the documents d_i and d_j, ∀i, *j* is defined by Equation (5)

$$SR(d_i,d_j) \begin{cases} N - l_{ij} & if\ dis(d_i,d_j) = 0 \\ -1 & otherwise \end{cases} \tag{5}$$

The approach describes three cases to measure, case 1 (maximum content similarity), case 2 (content dissimilarity), and case 3 (both content similarity and dissimilarity)

Consider case1 for maximum content similarity.

$$Case\ 1\text{:} \begin{cases} SR(d_i,d_j) = N - 0, i.e\ l_{ij} = 0 \\ SR(d_i,d_j) = N \end{cases}$$

$l_{ij} = 0, the\ contents\ of\ all\ the\ documents\ are\ exactly\ similar, i.e., maximum\ SR$

Let consider case 2 for content dissimilarity:

$$Case\ 2\text{:} \begin{cases} SR(d_i,d_j) = N - N, i.e\ l_{ij} = N \\ SR(d_i,d_j) = 0 \end{cases}$$

$l_{ij} = 1$, the contents are dissimilar

Let considers case 3 for both content similarity and dissimilarity:

$$Case\ 3\text{:} \begin{cases} SR(d_i,d_j) = N - (N > l_{ij}) \\ SR(d_i,d_j) = N - N' \end{cases}$$

Here the substantial resemblance defines using the distance between the two clusters C_x and C_y.

Let T_{XY} is a distance function to create a baseline function to measure the distance between C_X and C_Y.

$$T_{xy} = \{SR(d_i,d_j) \geq 0, \forall d_i \in C_x\ and\ d_j \in C_y\} \tag{6}$$

In Equation (6), *TXY* consists of all the occurrences of the substantial resemblance values for different parts of documents in the corpus shown in the Equation (7).

$$dis_{cluster(C_x,C_y)} = \begin{cases} \infty & if\ T_{xy} = 0 \\ N - avg(T_{xy}) & otherwise \end{cases} \tag{7}$$

$dis_{cluster(C_x,C_y)} = \infty, T_{xy} = SR(d_i,d_j) \geq 1\ \forall\ d_i \in C_x \& d_j \in C_y$
$SR(d_i,d_j) = 0, i.e\ N - l_{ij} => N = l_{ij}$, [1]

In [1], all documents are dissimilar, and no two documents have non-negative SR value. Therefore, with distance ∞ two clusters never be merged.

Here, the function $dis_{cluster(cx,cy)}$ finds the distance between two clusters, C_x and C_y as the average of the multi-set of non-negative SR values ($N > l_{ij}$)

$N - (N > l_{ij})$: $Non-negative\ SR\ values\ and\ N - (N < l_{ij})$: $Negative\ SR\ values$

3.1. Properties Measuring Clusters' Distance

The following properties 1 to 5 emphasize measuring the similarity and use to measure the distance between the clusters of the corpus. The distance between symmetric clusters is represented by properties 4 and 5 that are used for any pair of clusters.

1. $if\ Avg(T_{xy}) = N, then \min\left(dis(C_x, C_y)\right) = 0,$
   $i.e. SR(d_i, d_j) = N\ , Where\ \forall d_i \in\ C_x and d_j \in C_y$
   $otherwise, \max\left(dis(C_x, C_y)\right) = \infty$

Property 1 shows the distance between two clusters is minimum and represented by 0, i.e., the content similarity between the two set of documents d_i,d_j is N, otherwise *SR* is ∞.

2. $dis(C_x, C_y) = N - Avg(T_{xy}) = 0, then\ C_x = C_y\ , SR(d_i, d_j) = N, \forall d_i \in C_x and d_j \in C_y \Rightarrow dis(d_i, d_j) = 0,\ and\ l_{ij} = 0$
3. $dis(C_x, C_y) = dis(C_x, C_y), for\ symmetric,$ similarity measures for symmetric clusters [1].
4. $dis(C_x, C_y) \geq 0\ for\ anypair(C_x, C_y)$ of documents.
5. $for\ clusters\ C_x, C_y, and\ C_0,\ if\ 0 \leq dis(C_x, C_y) < N, 0 \leq dis(C_y, C_0) < N, and\ dis(C_x, C_0) = \infty$
   $then, dis(C_x, C_y) + dis(C_y, C_0) - dis(C_x, C_0) < 0$

Here, the content similarity of the document "o" compares the content similarity with documents "*x*" and "*y*." Similarly, the content similarity is measured for the rest of the documents. Likewise, the proposed algorithm (see Table 5) Travers the Hamiltonian graph to measure the content similarity.

4. Proposed Documents Grouping Approach

Section 3 describes illustrations that help understand the study's scope of measuring the documents 'content similarity and substantial resemblance [1].

The study's approach considers a Hamiltonian Graph-based clustering technique with no need to define clusters' initial values, unlike in the k-means approach [3] (see Tables 1 and 2), where an initial value (seed points) for creating the number of clusters needs to be assigned. The approach significantly deals with complex tasks, creates clusters, and produces accurate cluster results where the dataset is enormous. The substantial content resemblance between the documents [1] is searched during the process, and the content similarity state is defined [8] based on binary values. Then, Hamiltonian Graph properties are applied to a group of documents to find out the content similarity. A Hamiltonian path [23] is obtained by removing any one of the edges from the Hamiltonian cycle. This path traverses each vertex of the graph exactly once. The graph is connected if every pair of vertices must be a path of the graph. Generally, the length of a Hamiltonian path is n−1 in a connected graph where n is the number of vertices. Every Hamiltonian cycle has a path, but the converse does not exist. Every path in this cycle is a sub-graph, and each path can be directed or an undirected graph that traverses each vertex exactly once. Besides, the Hamiltonian graph demonstrates the complexity, i.e., $O\ (n * n!)$. Since the approach considers traversing the graph nodes [7] exactly once and ignoring the number of paths that usually appeared in a specific graph [2]. Remarkably, the approach-traverse all the nodes exactly once by tracing all the documents once and reducing complexity. Significantly, this method reduces the complexity of traversing the nodes multiple times and obtains linear complexity, i.e., O(n) by running the proposed algorithm over the newly constructed graph.

4.1. Hamiltonian Graph based Clustering Algorithm

Graph-theoretic methods should be one of the possible ways to improve computational performance [7]. Figure 3 (a graphical representation of the study's approach) describes vertices (A-K). These vertices follow example 1, from Table 6 pattern, and in the Figure 2, all vertices represent objects (document). Mark each object on the *XY* plane. Connect each object to all other objects either horizontally or vertically by dotted lines. Notably, while connecting the object, it should not intersect at any point of the coordinate either way. As shown in the figure, it connects (A-B), (B-C), and (C-D) by traversing only once. It should not proceed from D because the Y coordinates of D and Y coordinate of E intersect (3, 6). A similar approach applies to all the objects in the figure. This concept applies to making clusters, i.e., A, B, C, and D, form a cluster, say cluster 1. Similarly, the process forms two more clusters 2 (E, F, G, H) and 3 (I, J, K), from the graph. Here each cluster forms by following the Hamiltonian path on the *XY* plane.

Algorithm 5: Hamiltonian(k)

```
{
Do {
Nextvertex(k);
If (x[k]==0)
Return;
If(k==n)
Print (x [1: n]);
else
Hamiltonian(k+1);
} while(true);

}
```

Algorithm 6:Nextvertex(k)

```
{
Do {
X[k]=(x[k]+1) mod (n+1);
If(x[k]==0) return;
If(G[x[k-1], x[k]] ≠ 0)
{
```

```
For j=1 to k-1 do if(x[j]==x[k]) break;
If(j==k)
If (k<n or (k==n) && (G[x[n], x [1]] ≠0 || G[x[n], x [1]] =0)
Return.
} while (true);}
```

In this algorithm, $x\,[1{:}\,k-1]$ is a path of $(k-1)$ distance vertices. If $x[k]\ =\ 0$, then no vertex assigns to $x[k]$. After initial execution $x[k]$ is assigned to the next highest number vertex that doesn't appear in $x\,[1{:}\,k-1]$, if the graph is connected by an edge $x[k-1]$ else $x[k]=0$. If k=n then the graph connectivity of $x[k]$ is $x\,[1]$. Therefore, the linearity exists and minimizes the graph complexity.

4.2. Approach Significance

It can apply to any document corpus without bothering corpus's size, does not require an initial value assumption for clustering, minimizes the computation complexities, omits the randomization of the *k*-value, and produces quality clusters. Figure 4 shows another example where (B, C, D) forms a cluster. In this figure, vertex A (5, 3) on the *XY* plane, where the *Y* coordinate of A (*Y*=3) and *X* coordinate of B ($X\ =\ -1$) intersect at (-1, 3) of the *XY* plane. By following the approach, object A will not be part of cluster B. Similarly, the coordinate of vertex C and D intersect at (1, 3) and (-3, 3), respectively. Here, two clusters are created, 1 (A) and 2 (B, C, D). Table 11 shows that three clusters have created using Figure 2.

Table 11. Figure 2 produces three clusters.

Clusters		A	B	C	D	E	F	G	H	I	J	K
1	A	1	1	0	0	0	0	0	0	0	0	0
	B	1	1	1	0	0	0	0	0	0	0	0
	C	0	1	1	1	0	0	0	0	0	0	0
	D	0	0	1	1	0	0	0	0	0	0	0
2	E	0	0	0	0	1	1	1	0	0	0	0
	F	0	0	0	0	1	1	0	0	0	0	0
	G	0	0	0	0	1	0	1	1	0	0	0
	H	0	0	0	0	0	1	1	1	0	0	0
3	I	0	0	0	0	0	0	0	0	1	0	1
	J	0	0	0	0	0	0	0	0	0	1	1
	K	0	0	0	0	0	0	0	0	1	1	1

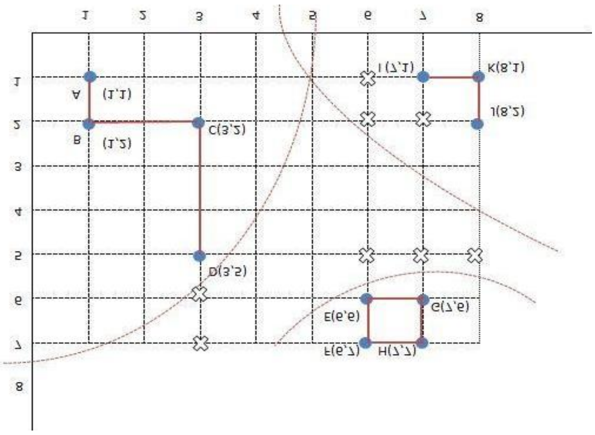

Figure 3.Graphical representation of the study's approach.

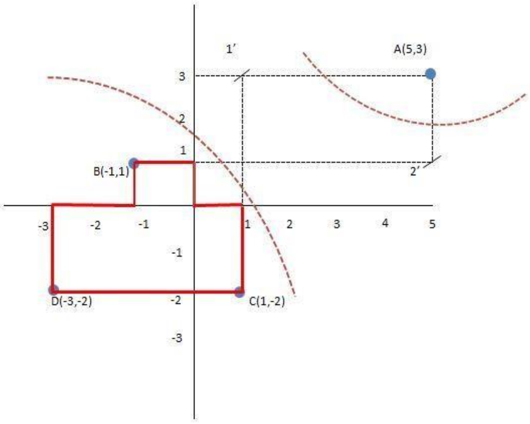

Figure 4. Graphical representation of the approach and Table 12 shows the achieved cluster of objects.

The approach considers all the graph vertices as documents, and the connectivity between the vertices represents documents' distances. If the documents' contents are most similar, then it corresponds to the least distance, and the least similar corresponds to maximum distance. Notably, all the documents with the least distance measure [8] form clusters, i.e., clusters with most content similarity or SR.

Table 12. Cluster of objects derived from Figure 4.

Items	A	B	C	D
Variables (X1,X2)	(5, 3)	(-1, 1)	(1,-2)	(-3, -2)

5. Result and Discussion

The presented study has considered several clustering methods to establish the study's idea of grouping documents by following Hamiltonian graph properties. Several existing methods, such as spanning tree, Bachelor and Wilkin's algorithm, distance matrix, graph-based clustering, substantial resemblance, and measuring distance properties, help establish the study's approach.

Significantly, these methods help to determine content similarity, clusters distances, and measuring substance resemblance between the documents in a corpus [1]. This approach applies to a real-time document corpus (see Table 13) of an academic program in a university environment.

5.1. Data Sets and Validity

Documents corpus is a collection of documents created and assembled with each source [22]. The approach applies to document corpus (see Table 13) of an academic program. The corpus contains more than 12000 documents generated by the seven sources of the program activities. The validity of document corpus is that tons of documents are generated during the program's academic cycle and need to be organized and stored for future uses.

Table 13. Overview of the documents data sets.

Documents' source	Value	Instances and description
Year	5x2 (S_1,S_2)	10- 5years with two semesters
Student registration sys	5x2x4	40 (5 years, 10 semesters, 2 enrolments, 2 retentions)
Learning management sys	5x2x40x2x2	1600 (Notices and activities, 2=Male and female sections)
Examination processing sys	5x2x40x2	800 (2=Male and female sections)
Academic course files	5x2x40x10x2	8000 (40=courses,10=documents in each course, 2=M&F)
SLOs measurement	5x2x10x3x3	900 (10=SLOs, 3=courses, 15= evidence)
Accreditation documents	9x15	135 (9= Criteria, 15=documents)
Additional documents	5x100	500 – the additional documents associated to the program

5.2. Limitations and Approach Impact

For effective document grouping, mostly used clustering algorithms and their characteristics investigated in section 1. The limitations of Table 2 show that prior knowledge of the number of clusters (seed points) is required, complexity is an issue with an extensive data set, and selecting k-value for an unknown data set are few limitations. The proposed approach overcomes these limitations by adopting a Hamiltonian graph-based clustering technique [11, 23], as discussed in section 4. For instance - no random initialization and prior knowledge of a number of clusters are required, i.e., k-value, which is unpredictable in other processes. Further, it significantly supports a high-density document corpus by traversing all the graph's nodes and edges exactly once. Besides, introduces the computational complexity overheads using the algorithm's iterative techniques.

Some clustering algorithms like AHC algorithms require termination criterion (see limitation 9, Table 2), which is difficult to estimate such conditions for a large dataset [1]. The proposed algorithm eliminates the halting condition using the Hamiltonian graph properties. Further, it reduces the computational expensiveness if, $x[k] = 0$, then no vertex assigns to $x[k]$ i.e., the maximum content similarity between the documents exists, and the control moves to the next highest number vertex. If $k = n$ then the graph connectivity of $x[k]$is $x\ [1]$, this shows that linearity exists and minimizes the graph complexity. The algorithm minimizes other graph-based algorithms' errors using the iterative techniques, i.e., the NeXT vertex algorithm. Notably, it generates clusters automatically despite the documents corpora's size and complexity.

5.3. Implications

The proposed approach can determine the number of clusters in advance to implement the algorithm. The number of clusters can obtain by applying threshold θ, i.e., the substantial similarity value between documents ((see Equation (2)) and restricts lower similarity values. Further, it determines the Cosine similarity value between documents, i.e., $\rho(\mathrm{d_i}, \mathrm{d_j})$. The approach may produce several very compact clusters and generate a vast number of small-sized clusters that could not be expected in practice. It is suggested that the Cosine value, i.e., ρ should be chosen logically when the proposed approach is applying to any intended application.

The proposed approach is based on Hamiltonian graph properties, and the path of the graph can obtain by removing any one of the edges from the Hamiltonian cycle. The proposed algorithm traverses each vertex of the graph exactly once, obtains linear complexity, i.e., $O(n)$, and helps to avoid the computational overhead. Indeed, the application developers should consider the complexity, i.e., $O\ (n\ *\ n!)$ very carefully, because it varies with the graph properties. Additionally, the proposed approach experiences similar limitations as agglomerative hierarchical clustering algorithms during the distance measuring process.

6. Conclusions and Future Work

The presented study initially investigates the commonly practicing document clustering algorithms. It determines the limitations of such algorithms and identifies the potential algorithms, such as AHC, spanning tree, and Bachelor and Wilkin's. It illustrates an undirected weighted graph, i.e., Figure 2, to measure the documents' content similarity and the distance matrix (Table 8) to measure the clusters' distances. Further, it demonstrates a SR approach and applies the distance measuring properties to measure the content similarities and distances between the clusters.

The proposed algorithm considers the graph-based clustering techniques and the distance measuring properties. The clusters are generated based on

1. Hamiltonian graph properties.
2. Substantial content similarity measures.

The cluster's quality measure depends on the substantial content resemblance between any two documents and their distances of every other document in the corpus. The documents with an SR are grouped into the clusters of maximum content similarity, whereas the documents with a low content similarity grouped in another group of clusters, i.e., singleton. The algorithm considers the group of singleton clusters and performs iterative operations to correlate to the baseline clusters. The Nextvertex algorithm's iteration technique reduces the computational complexity, unlike the other graph-based algorithms. The proposed algorithm automatically decides the number of clusters (seed points) and produces maximum and least content similarity clusters. It measures the content similarity using a threshold value θ (0.52 and 0.44) from the corpus (see section 3). Further, it omits the randomization of the *k*-value, whereas the other said clustering techniques are required *k*-value. For application developers, this approach should be an opportunity while developing document management applications.

In future, we shall consider the proposed approach on the university all types of document corpus and data. The approach can be applied to different types of other data, e.g., the social network and ongoing COVID-19 pandemic data. In such cases, the proposed approach reassesses and needs to determine the relations between different sets of nodes (i.e., social site and the regional COVID data). For instance, the application developers and data analysts should consider the proposed algorithm and its linear complexity for clustering the large data corpus. The study's authors are putting their efforts into applying the proposed approach to the regional COVID-19 data for classifying the more affected communities in the region.

Acknowledgement

The authors would like to express their gratitude to King Khalid University, Saudi Arabia for providing administrative and technical support. This work was supported by the Deanship of Scientific Research, King Khalid University, under Grant GRP/358/42.

References

[1] Basu T. and Murthy C., "A Similarity Assessment Technique for Effective Grouping of Documents," *Information Sciences*, vol. 311, pp. 149-162, 2015.

[2] Chandola V., Banerjee A., and Kumar V., "Anomaly Detection : A Survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 1-58, 2009.

[3] Chiang M., Tsai C., and Yang C., "Time-Efficient Pattern Reduction Algorithm for K-Means Clustering," *Information Sciences*, vol. 181, no. 4, pp. 716-731, 2011.

[4] Chouder M., Rizzi S., and Chalal R., "EXODUS: Exploratory OLAP over Document Stores," *Information Systems*, vol. 79, pp. 44-57, 2019.

[5] Conrad J., Al-Kofahi K., Zhao Y., and Karypis G., "Effective Document Clustering for Large Heterogeneous Law Firm Collections," *in Proceedings of the International Conference on Artificial Intelligence and Law*, Bologna Italy, pp. 177-187, 2005.

[6] Dasgupta S. and Ng V., "Towards Subjectifying Text Clustering," *in Proceedings 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Switzerland, pp. 483-490, 2010.

[7] Figueroa K. and Paredes R., "Approximate Direct and Reverse Nearest Neighbor Queries, and The K-Nearest Neighbor Graph," *in Proceedings 2nd International Workshop on Similarity Search and Applications*, Prague, pp. 91-98, 2009.

[8] Forsati R., Mahdavi M., Shamsfard M., and Meybodi M., "Efficient Stochastic Algorithms for Document Clustering," *Information Sciences*, vol. 220, 2013.

[9] Gallinucci E., Golfarelli M., and Rizzi S., "Schema Profiling of Document-Oriented Databases," *Information Systems*, vol. 75, pp. 13-25, 2018.

[10] Grygorash O., Zhou Y., and Jorgensen Z., "Minimum Spanning Tree Based Clustering Algorithms," *in ProceedingsInternational Conference on Tools with Artificial Intelligence*, Arlington, pp. 73-81, 2006.

[11] Hecht R. and Jablonski S., "NoSQL Evaluation A Use Case Oriented Survey," *in Proceedings of the International Conference on Cloud and Service Computing*, Hong Kong, 2011.

[12] Joudaki H., Rashidian A., Minaei-Bidgoli B., Mahmoodi M., Geraili B., Nasiri M., and Arab M., "Using Data Mining to Detect Health Care Fraud and Abuse: A Review of Literature," *Global Journal of Health Science*, vol. 7, no. 1, pp. 194-202, 2015.

[13] Jing L., Yu J., Zeng T., and Zhu Y., "Semi-Supervised Clustering via Constrained Symmetric Non-negative Matrix Factorization," *in Proceedings International Conference on Brain Informatics*, Athens, pp. 309-319, 2011.

[14] Langville A., Meyer C., Albright R., and Cox J., "Initializations for the Nonnegative Matrix Factorization," *in Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Philadelphia, pp. 1-8, 2006.

[15] Lasek P. and Gryz J., "Density-based Clustering with Constraints," *Computer Science and Information Systems*, vol. 16, no. 2, pp. 469-489, 2019.

[16] MacQueen J., "Some Methods for Classification and Analysis of Multivariate Observations," *in Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, no. 14, pp. 281-297, 1967.

[17] Mohamed M., Ghanem S., and Nagi M., "Privacy-Preserving for Distributed Data Streams: Towards l-Diversity," *The International Arab Journal of Information Technology*, vol. 17, no. 1, pp. 52-64, 2020.

[18] Ng A., Jordan M., and Weiss Y., "On Spectral

Clustering: Analysis and an Algorithm," *in Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, Vancouver, 2002.

[19] Pantel P. and Lin D., "Document Clustering with Committees," *in Proceedings of the 25th annual international ACM SIGIR Conference on Research and Development in Information Retrieval*, Tampere Finland, pp. 199-206, 2002.

[20] Steinbach M., Karypis G., and Kumar V., "A Comparison of Document Clustering Techniques," *in Proceedings of the IEEE International Conference on Computational Cybernetics*, Slovakia, pp. 1-2, 2000.

[21] Wang J., Wu S., Quan Vu H., and Li G., "Text Document Clustering with Metric Learning," *in Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Geneva, pp. 783-784, 2010.

[22] Wang Y., Choi I., and Liu H., "Generalized Ensemble Model for Document Ranking in Information Retrieval," *Computer Science and Information Systems*, vol.14, no. 1, pp. 123-151, 2017.

[23] Wierzchoń S. and Kłopotek M., *Studies in Big Data 34 Modern Algorithms of Cluster Analysis*, Springer, 2018.

[24] Xu W. and Gong Y., "Document Clustering by Concept Factorization," *in Proceedings of Sheffield SIGIR-27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Sheffield United Kingdom, pp. 202-209, 2004.

Mohammed Alghobiri is an experienced highly involved in information systems development and implementation, especially experimental and participative approaches. Part of his interest includes Management and Evaluation of Systems Development, including Software process improvement methods and ERP D&I. Databases, Data Mining, Decision Support Systems, and Electronic Government Concepts are also within his concern.

Khalid Mohiuddin recognizes as a researcher and an excellent teaching practitioner at the faculty of Information Systems at King Khalid University, Saudi Arabia. His interdisciplinary research involves Information Systems management, mobile cloud computing-IoT, edge, mobile edge, fog, AI, 5G, and quality development in higher education.

Mohammed Abdul Khaleel worked as a senior software developer in information systems management. Presently, he serves as a faculty member at the College of Computer Science, King Khalid University, Saudi Arabia. His research interest includes data mining, cloud data management, mobile cloud data management, and data management in server less computing.

Mohammad Islam is a research scholar at King Khalid University, Saudi Arabia. He has rich experience in research and teaching Information Systems. His interdisciplinary research interest includes business IS and business intelligence, and quality development in higher education.

Samreen Shahwar is a lecturer and research scholar at King Khalid University, Saudi Arabia. Her research interest involves information systems management, cloud computing, education learning, and higher education assessment.

Osman Nasr He is currently working as an Assistant Professor at the Department of Management Information Systems, King Khalid University in the Kingdom of Saudi Arabia. His research interests include Data mining, web-based systems, cloud computing, educational research, and quality development in higher education.