



## RESEARCH ARTICLE

# A Novel Fragmentation Scheme for Textual Data Using Similarity-Based Threshold Segmentation Method in Distributed Network Environment

Sashi Tarun

School of Computer Science and Engineering, Lovely Professional University, Phagwara, India.  
sashitarun79@gmail.com

Ranbir Singh Batth

School of Computer Science and Engineering, Lovely Professional University, Phagwara, India.  
ranbir.21123@lpu.co.in

Sukhpreet Kaur

Department of Computer Science and Engineering, Chandigarh Engineering College, Mohali, India  
sukhpreet.4479@cgc.edu.in

Published online: 25 December 2020

**Abstract** – Data distribution is one of the most essential architectures of any serving network. Data storage and its retrieval depend a lot on how the data is organized in the distributed environment. With the fast development of technology, the requirements of users have also changed. A user who was stationary earlier has become mobile now and requires access to the data from anywhere in the world. An unorganized data structure will result in output delay in the network and may further result in user migration from one service provider to another service provider. Data fragmentation is one of the most essential parts when it comes to data storage. Organized data always gives convenience to others to use it conveniently. Due to the vast collection of data extraction of information in a fast manner is very complicated. So, to achieve performance in a distributed system an optimal strategy is required to overcome previous lapses and serves the maximum number of users in a wide geographical network. This research paper proposes a novel relative based fragmentation method that analyses the attributes of the data in relative architecture and is helpful to achieve query performance with better speed and accuracy. To assess the current proposed work a comparison has been drawn between k-means dependent cosine similarity measurement and hybridization of cosine and soft-cosine partition methods for data partitioning. Mentioned results in the article shows that the proposed similarity-based threshold segmentation method outperforms the existing in terms of partitioning strategy, precision, and recall parameters to achieve performance.

**Index Terms** – Fragmentation, K-Means, Similarity, Data Partitioning, Threshold, Segmentation, Precision, Recall.

## 1. INTRODUCTION

A distributed system refers to the use of independent computers engaged to share various resources in the

connected networks. A good distributed design is having the capability to cover each data-items requirement raised by the users. Users looking for desire data or information, using different query angles. Some of them focus on data-items belonging to a single table or a combination of more than one. From different angles, user queries are classified as fine and coarse-grained, single database or multi-database, and follow a collective and selective approach to reach required data [1]. If knowledge grows at a rapid rate day by day, design architecture should be scalable to handle vast amounts of information in the future.

The mechanism of fragmentation, allocation, and deduplication of data is associated with improving data in a distributed network. Proper utilization of available storage space is achieved by dividing the large global data into small independent parts called fragments or segments [2]. These independent segments help to reduce the load in a widely distributed environment as compared to accessing data from a single large data schema. With distributed systems, issues such as scalability, data availability, security, searching speed, and inconsistency of data can be easily managed. It has become difficult to work with a single large data system and to entertain millions of users simultaneously with high data accuracy.

The growth of cloud computing, VANET, OPPNETs is the product of parallel technology, software technology, and network infrastructure innovations [3-4]. This is a new form of a computer model that provides users with the data, applications, and various IT resources through the network as a service without delay in the exchange of information [5]. Cloud computing can be considered as a kind of infrastructure

**RESEARCH ARTICLE**

management tool, resource management through virtualization technology, which consists of a large capacity resource pool. Cloud users will send requests through the network and then receive the service. In which dynamically deploy, modify, and reconfigure the resource pool, and cancel the operation, etc are included [6].

The availability of numerous services over the Internet is cloud computing. These assets include data processing software and apps for records, servers, computers, networks, and tablets. You can store files in cloud-based storage in an external database rather than storing them on your hard drive or local storage unit. Tools and data resources can be used as long as an electronic device has access to the Internet. Cloud storage is referred to as such because it is possible to remotely recover the stored information in the data or a virtual space. On remote servers, cloud service providers allow users to store files and programmes and then access all data online. This means that the user does not have to be in a certain position to use it in order to be able to function remotely.

Cloud storage is a popular choice for individuals and businesses for many reasons, including cost savings, increased efficiency, quality and efficiency, reliability and security [7].

The key aim of this study is to implement a new fragmentation architecture suitable for scalable question addressing in the distributed network world in terms of textual data. Earlier data fragmentation was based on empirical data and far-reaching to get the desired result. This approach introduces a novel relative-based fragmentation architecture where there is no ground reality and similarity calculations are carried out on the textual data to reach the conclusive result using vector calculations. This paper uses a mixture of similarities of cosine, soft cosine, and hybrid similarity as a differentiation between the entities of data for partitioning.

Existing design not proved to be effective on diverse data trends include textual data context. Earlier techniques focus on finding similarities between more than one documents but this technique is responsible to find the similarity in the relation itself by comparing each row to one another and apply similarity calculation on vector values. So, there is a need to depend on the required strategy suitable for a diverse environment with adaptive nature.

As follows, the paper is structured. The classification of data is in the form of a category based on related attributes is addressed in Section Segmentation. Section Similarity Measures helps to discover the relationship between the rows in relation or two data-items using Cosine, Soft Cosine similarity, and hybrid similarity. In the proposed methodology section steps are evaluated one by one i.e. selection of dataset, stop word removal, word-to-vector conversion, implementation of cosine, soft-cosine, and hybrid similarity,

determination of initial centroid of the dataset used, Euclid distance calculation, finding centroid positions for each cluster, creating fragmentation, and validating the fragments using machine learning and neural network are included. The outcomes and discussion of the research work section is included to illustrate the feasibility of the work proposed. Comparative analysis is done at the end of the section to equate the new model with the current scheme.

### 1.1. Segmentation

Segmentation is the method of collecting data with similar properties or separating cloud data into smaller, coherent, and interconnected areas. Text segmentation is a process by which a document is split into smaller parts, typically called segments. It is used extensively in word-processing. These segments are classified as word, phrase, subject, sentence, or any unit of information, depending on the task of the text analysis. The method of removing coherent blocks of text is Text segmentation. The section is called the boundary section or passage.

There are several explanations of why a split document could be useful for the analysis of the text. One explanation for this is that it is smaller and more coherent than whole documents. Segmentation of text is a big problem when it comes to obtaining information. It aims to divide a text into homogeneous segments i.e. segments with the following characteristics: (a) each segment has a particular topic and (b) adjacent segments deal with different topics. These segments can be traced as appropriate to a query from a broad base of unformatted or loose text [8].

### 1.2. Similarity Measures

When there is available ground truth for the clustering then the similarity value will be evaluated through the ground truth of the cluster or region but when there is no ground truth of the region or cluster then the ground truth becomes radical and hence similarity measures are calculated through vector calculation. Discourse is the measurement of the equivalence of two pieces of evidence. In the sense of data mining, an agreement is commonly defined as a gap along with the dimensions describing the objects' properties. The degree of similarity will be high if this distance is small; if a distance is large, the degree of similarity will below. The similarity measure is used in many ways, including plagiarism, asking for a similar question previously asked about Quora, collaborative filtering in recommendation systems, etc. A similar measure may be described as the distance between various points of data. While the similarity is a quantity that represents the strength of the relationship between two data items, the difference is between the two data items measuring divergence. Three similarity measures are used in conjunction in this study, namely Cosine, Soft Cosine similarity, and hybrid similarity [9].



**RESEARCH ARTICLE**

The resemblance is determined in the 0 to 1 [0, 1] scale.

- Relationship equal to 1 such that if X = Y
- Similarity is equal to 0, unless X is equal to Y

Where, X, Y are two different vector lists.

Similarity is arbitrary and relies extensively on the domain and its use. Two fruits, for instance, are similar because of color or height, or taste.

**1.3. Cosine Similarity**

Similarity is in general, a measure of similarity; that is, how similar things are compared to similar things. One was with the use of vectors, an equation for the computer. A vector is literally a quantity which has both size and direction. A vector is considered a 1-dimensional sequence in Computer Science. The resemblance of cosine is a method used to calculate the angle of cosine between them. The point product of the two vectors is required for finding the angle between the two vectors as shown in Figure 1. Measurement of cosine equality assumes the uniform point sum of the two objects. By determining the cosine relation, we will effectively attempt to find the cosine of the angle between the two lines. The 0° cosine is 1, and it is less than 1 for every other variable. Therefore, it is an orientation and not a magnitude judgment: two vectors with the same direction have a cosine similarity of 1, two vectors with a 90° similarity of 0, and two vectors with the same direction have a similarity of -1, irrespective of their magnitude [10].

$$\text{Cos } \theta = \frac{\vec{A} \cdot \vec{B}}{\|\vec{A}\| \|\vec{B}\|}$$

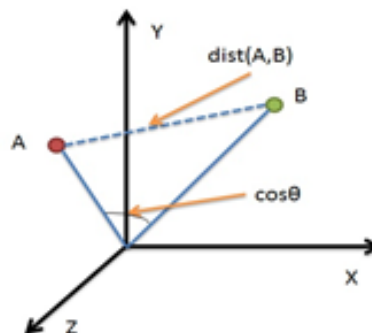


Figure 1 Similarity of Cosine

**1.4. Soft Cosine Similarity**

A soft cosine agreement tests attribute to the agreement. The conventional criterion of Cosine conformity determined similarity based on features determined by the model of vector space (VSM), which are completely different from each other. On the other hand, it can be a great advantage of soft cosine similarity if one needs to use a criterion of agreement that can help with the grouping or classification of documents [11].

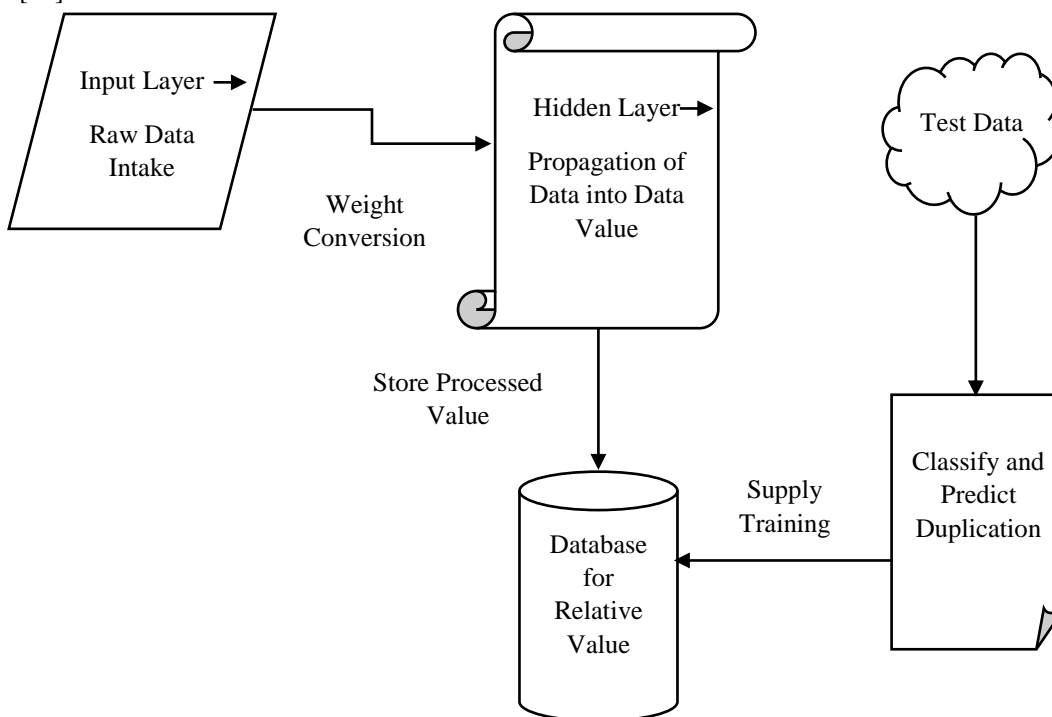


Figure 2 Learning Computer Architecture

**RESEARCH ARTICLE**

## 1.5. Hybrid Similarity

In this similarity measures, the features of cosine similarity and soft cosine similarity are combined.

## 1.6. Machine Learning

Machine learning (ML) is an artificial intelligence (AI) technique that without specific programming provides devices with the ability to learn and strengthen automatically. Machine learning focuses on the development of computer systems capable of viewing and learning knowledge themselves. Based on the explanations we present, the learning process begins with insights or data, such as examples, direct knowledge or input, looking for data patterns, and making informed future decisions. The key goal is to allow computers learn automatically and adjust behavior appropriately without human intervention or support [12]. ML categorization is always undertaken as given below;

- Machine learning algorithm supervised
- Unsupervised Algorithm for machine learning
- Semi-supervised algorithm for deep learning
- Machine learning algorithm for strengthening [13]

The learning computer architecture (shown in Figure 2) works completely on the relative information model. It learns from what is provided to it. It is separated into the input layer, the hidden layer, and the output layer into three parts. The input layer takes the data as raw data and transforms it against the specified goal label into a more understandable form. The hidden layer propagates the meaning of the data and generates the cross-validation training platform.

**2. RELATED WORK**

Several contributions were given by the researchers to build the distributed system robust. It was noticed that the researchers did not stress the efficacy of the proposed data partitioning work and that it was important to strengthen it. This issue arises due to methods of unsupervised data partitioning. Some researchers have used clustering methods, but the accuracy is lower because of the presence of unsupervised algorithms, and it is important to use the theory of similarity for clustering. K-methods are considered to be a less popular clustering algorithm since clustering is too difficult and costly to use than k-mean. The suggested algorithm uses a standard deviation that decreases the overall time for formulating the cluster by a simple k-mean. The proposed solution splits the gap with the standard deviation of the square root. This modified k-mean does even better than k-mean and k-methods. Less time is needed to formulate the clusters. But neither k-means nor k-methods also work on very large-scale outcomes. The authors have not used any kind of similarity measurement technique for distance

calculation between different types of data that results in poor clustering performance, and it must be integrated with k-means using optimization approaches [14]. Researchers suggest a deep neural network such as CODEnnn (Code-Description Embedding Neural Network). CODE does not fit the textual resemblance, but includes code fragments and high-dimensional vector field examples in natural language, as well as associated vectors in the code fragment and the accompanying definition. Code fragments associated with natural language questions can be obtained by the associated vector representation in accordance with their vectors. In the queries that must be handled, the task could even be semantically identified with the keywords. The researchers did not include source code management structures in this study to help symbolize, and the deep neural network is used and limited for the basic benefit of information engineering issues [15]. For image co-segmentation tasks, a new clustering algorithm called salience-guided limited cosine similarity clustering method (SGC3) has been proposed, where a one-step clustering technique extracts the usual foreground. In the method, the unsupervised significant prior is used to direct the clustering mechanism's auxiliary partition-level information. To ensure the robustness of the noise and outliers in a given previous one the similarity between the instance level and the partition level is used for joint estimation. Eventually, the optimization of associated K-means aims to successfully solve the objective function. Experimental outcomes from two widely used data sets show that the proposed solution has achieved successful performance against the most mature distribution methods [16]. A systematic experimental analysis of twenty-four benchmark functions in a test suite. ABC (Artificial colony of bees) is a very common and effective tool for optimization. ABC still does however have a lack of convergence. In order to further increase ABC convergence velocity, a new form of ABC (CosABC) is proposed to pick better neighbors based on cosine similarity. Under the direction of chosen neighbors, a new solution search equation was applied to reduce the constraint of ABC undirected search. There is a further contrast with some of the most sophisticated algorithms to check Cos-ABC supremacy. The related results of the comparison show that Cos-ABC is efficient and competitive [17]. To find similar knowledge for a user whose original question cannot be addressed precisely, clustering-based fragmentation is suggested. Approximation algorithms and lookup tables are used to give a better shape to the distributed system for supporting flexible query answering [18]. Work for optimized fragmentation approach on each attribute was conducted to know about their retrieval and update frequencies in each site. And proposed a synchronized horizontal fragmentation approach to reduce data locality issues and total cost. In this work, if query (Q) is initiated from multiple (M) locations, this query will be interpreted as a separate query for each position with a different radio

**RESEARCH ARTICLE**

frequency [19]. To achieve fragmentation an algorithm based on agglomerative hierarchical clustering was introduced to reduce the number of iterations. It mainly focused on the minimization of transmission cost [20]. It is proposed to render fragments vertically with an Updated Bond Energy Algorithm (BEA). This algorithm utilizes attribute affinity and seeks to create clusters of attributes and attributes that are individually evaluated by the same query [21]. A hybrid fragmentation approach for deductive database systems is proposed as HFA for horizontal fragmentation and, RCA and DVF for vertical fragmentation. This is a two-phase process deductive database is fragmented using variable bindings and dependency relationships represented by dependency graph [22]. For the efficient partitioning of large datasets without query statistics, MCRUD and MMF algorithms have been suggested. It is suggested here that earlier partitioning methods were not acceptable because there were no usable statistics at the initial stage of the implementation of distributed database query statistics. [23]. Work on frequent access patterns (FAP) is given to reflect the behavior workload to ensure the data integrity and ratio of approximation. It presented a data structure that was based on trees by utilizing the depth-first search (DFS) coding for maintaining them as well as to manage newly entered queries [24]. The fragmentation mechanism has recently been shown to have a detrimental effect on the performance of negatively exported processes. Finally, by merging Process Mining (PM), Social Network Analysis (SNA) and Text Mining, the fragmentation process and improved knowledge sharing among port Community System (PCS) actors have been improved so that process efficiency can be achieved [25]. Study on data fragmentation in the public and private sectors is being carried out in order to hold information in a structural archive in order to attain data security [26].

It is concluded that, continuous efforts were given to improve partitioning strategies for distributed network environment. Earlier, partitioning of data was not using query statistics but later on the basis of the behavior of users' frequent access pattern has used to make partitions. Due to lack in picking neighbors for the clustering and as a result query get affected during response and suffered delay. So for effective partitioning, information sharing between network nodes, maintaining efficiency, controlling delay and balancing data load a new scheme is required work for diverse environment.

**3. PROPOSED METHODOLOGY**

The steps are as follows:

- i) Begin by uploading unlabeled Twitter data.
- ii) Apply filtering of data by removing English Stop Words.
- iii) Apply word to vector operations.

- iv) Apply similarity calculations on the set of row list.
- v) Finding of initial centroid of data.
- vi) Calculation of Euclid distance of each tweet.
- vii) Finding the number of centroids.
- viii) Creating fragments using K-Means.
- ix) Validate the fragments using ML Training and classify using a neural network.

To achieve this, we have followed further processes in sequence. The step description of each process is provided in the next sections. Each step is individually implemented in Matlab R2016a.

**3.1. Dataset Used**

The dataset used for the proposed work is from the sample of data and is accessible from [27] on 27.12.20.

The above-defined dataset consists of tweets in unlabelled form, which needs to be segmented, and for segmentation. In this dataset "text" column having 1048576 instances of data out of that 93 instances are used for further implementation of the relative fragmentation experiment.

**3.2. Stop Word Removal**

The foremost step is to filter English stop words. The list of those stop words can be accessed from stop list [28] on 27.12.20.

The process of removing English stop words (as shown in Figure 3) from the tweets helps to decrease the dimension of the available data.

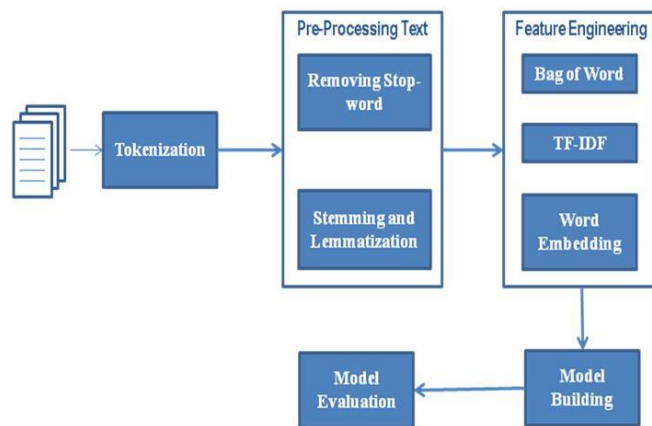


Figure 3 Stop Words Removal Process

Prepositions, articles, nouns, seem to be the most common words in text documents, etc. These words do not provide meaningful information about the text. Stop list words were omitted from the text because certain words in information retrieval (IR) software are not called keywords. For E.g. By

**RESEARCH ARTICLE**

maintaining an English stop word dictionary, English stop words are deleted from each text file in the data set [29].

For the removal of stop words, the code snippet is given below in Figure 4.

```

for i=1:r
    currentdata=rawdata{i,1};

    words=getwords(currentdata);
    allwords(i)=words;
    filteredwords=filterdata(words, stopwords);
    allfiltered(i)=filteredwords;
    for wd=1:numel(filteredwords)
        w2v(i, wd)=sum(double(filteredwords(wd)));
    end
end

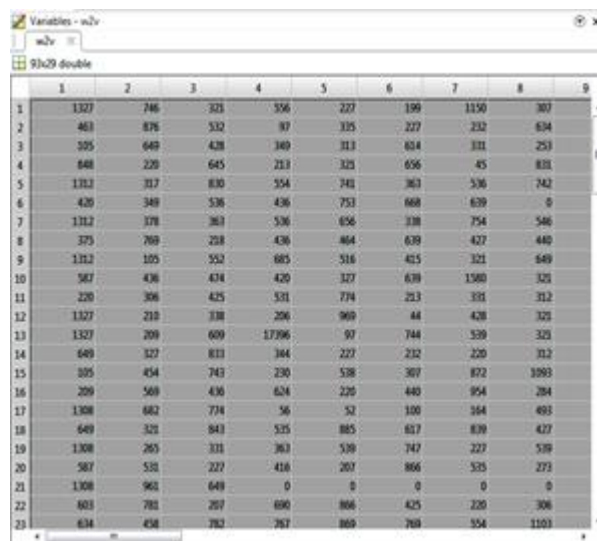
```

Figure 4 Stop Word Removals and W2V Conversion

3.3. Word to Vector

Similarity is usually a test of similarity, i.e. how similar or similar objects are compared. By using vectors, one method of calculating similarity is. A vector is essentially a number that has both direction and magnitude. A 1-dimensional sequence is considered a vector in Computer Science. A way to build a compact space of word vectors is Word2vec. It takes as input a broad text corpus (tweets after stop words have been removed) and assigns each word in the tweet a vector. First a dictionary is generated and then the vector representation of the terms is computed. Contextual proximity based vector representation: the words in the text adjacent to the same words (and thus have a similar meaning) in the vector representation have indexes of high similarity. The values of the vector after approximation (as seen in Figure 5) are represented in the dataset for each row of words in column 5. Therefore, the next step is to evaluate the similarity index of the data using three similarity measures named Cosine, Soft Cosine, and hybrid similarity index. A detailed description of these three measures is provided in the upcoming sections.

Skip-gram and Continuous Bag of Words are the two primary models in word2vec. In the Skip-gram model, terms are predicted from a word in their context, and the most possible word is chosen on the basis of the context in the CBOW model. To get the output of the probability distribution of each term, the output layer uses a softmax feature or a combination of it. Input and output words are given in one-hot encoding in these models, such that a single row is chosen  $W$  when multiplied by the matrix  $W$  connecting the input and hidden layers. Dimension  $N$  is the algorithm hyper parameter and the qualified  $W$ -output matrix, since its lines contain vector representations of terms. [30].



	1	2	3	4	5	6	7	8	9
1	1327	746	321	556	227	199	1150	307	
2	461	876	532	87	335	227	232	634	
3	505	649	438	349	313	654	331	253	
4	648	228	645	213	325	656	45	831	
5	1312	317	830	554	741	363	536	742	
6	426	349	536	436	753	668	639	0	
7	1312	378	363	536	656	338	754	546	
8	375	769	258	436	464	639	427	440	
9	1312	105	552	685	516	415	321	649	
10	567	436	474	420	327	639	1580	321	
11	220	306	425	531	774	213	331	312	
12	1327	210	338	206	969	44	428	321	
13	1327	209	609	1796	97	744	539	321	
14	649	327	811	344	227	232	220	312	
15	105	454	743	230	538	367	872	1093	
16	209	569	436	624	226	440	954	284	
17	1308	682	774	56	52	100	364	493	
18	649	321	943	525	865	617	839	427	
19	1308	265	311	363	539	747	227	539	
20	567	531	227	418	267	866	535	273	
21	1308	961	649	0	0	0	0	0	
22	603	781	297	690	866	425	220	306	
23	634	438	782	387	869	769	554	1103	

Figure 5 Vector Representation of the Word

To speed up the training of Skip-gram and CBOW models, modifications are used softmax, such as hierarchical softmax and negative sampling, which allow calculating the probability distribution faster than in linear time from the size of dictionary [31-32].

3.4. Cosine/ Soft Cosine / Hybrid Similarity Index

A document in the vector model is considered as an unordered set of terms. Terms to retrieve information are the words that make up the text to obtained essential or useful information. Here Cosine similarity is applied to calculate the similarity index for the uploaded document to the rest of the text in the set. For example, if we have considered 100 rows in the tweet, then the similarity (either cosine or soft cosine) is determined by comparing the word vector to the rest of the 99 rows. In this way for row 2, row3, row4.....row 100, have to be determined.

Input: Word to Vector data

Output: simvalue=Calculatecossim(v1, v2)

1. Calculatecossim(v1, v2) = [ ];
2. nume = 0; //numerator
3. deno=0;//denominator
4. deno1=0;
5. deno2=0;
6. for I = 1 → v1.length
7. nume=nume+v1(I)\*v2(I);
8. End for
9. for J = 1 → v1.length

**RESEARCH ARTICLE**

10.  $deno1=deno1+v1(J)^2;$
11.  $deno2=deno2+v2(J)^2;$
12. End for
13.  $deno=sqrt(deno1)*sqrt(deno2);$
14.  $simvalue=nume/deno;$
15. Return: *simvalue* as output
16. End function;

---

Algorithm 1 Cosine Similarity between Vectors

Algorithm 1 showing the functioning of cosine similarity, it is represented by calculating the cosine angles between two vectors *v1* and *v2*. To do this, in relation each row is compared with other rows using a vector list and use numerator and denominator as a variable. It is calculated by multiplying each vector with one another row-wise sequentially and stores their result in *nume* variable. And *deno* is calculated by multiplying by the square root of *deno1* and *deno2*. *deno1* and *deno2* is the square of *v1* and *v2* respectively. In the end, *simvalue* is calculated by the division of *nume* and *deno*.

Input: Word to Vector data

Output:  $sc=Calculatesoftcosine(v1, v2)$

1.  $Calculatesoftcosine(v1,v2)=[]$
2.  $sc=0;$
3.  $num=0;$
4. for  $I = 1 \rightarrow v1.length$
5. for  $J = 1 \rightarrow v2.length$
6.  $num=num+ v1(I)*v2(J);$
7. End for
8. End for
9.  $avalue=0;$
10.  $bvalue=0;$
11. for  $I = 1 \rightarrow v1.length$
12. for  $J = 1 \rightarrow v1.length$
13.  $avalue=avalue+v1(I)*v1(J);$
14.  $bvalue=bvalue+v2(I)*v2(J);$
15. End for
16. End for
17.  $avalue=sqrt(avalue);$
18.  $bvalue=sqrt(bvalue);$

19.  $deno=avalue*bvalue;$
20.  $sc=num/deno;$
21.  $sc=sc/(max(v1)/max(v2));$
22. End function

---

Algorithm 2 Soft Cosine Similarity

To achieve accuracy in the result of cosine similarity an improved algorithm as soft cosine similarity is proposed. Algorithm 2 is used to calculate soft cosine by a division of numerator and denominator. The numerator is the multiplication of *v1* and *v2* for each row with other available rows in the relations. The denominator on the other side is the square-root of *v1* and *v2* for each row and column.

Input: calculated results of cosine similarity and soft cosine similarity

Output:  $allhybrid=hybridsim()$

1.  $[r,c]=size(w2v);$
2.  $allcossim=[];$
3.  $allsoftcossim=[];$
4.  $allhybrid=[];$
5. for  $i=1 \rightarrow r$
6.  $v1=w2v(i,:);$
7.  $simvalue=0;$
8.  $softvalue=0;$
9.  $counter=0;$
10. for  $j=i+1 \rightarrow r$
11.  $v2=w2v(j,:);$
12.  $simvalue=simvalue+Calculatecossim(v1,v2);$
13.  $softvalue=softvalue+Calculatesoftcosine(v1,v2);$
14.  $counter=counter+1;$
15. End for
16.  $allcossim(i)=simvalue/counter;$
17.  $allsoftcossim(i)=softvalue/counter;$
18.  $allhybrid(i)=allcossim(i)+allsoftcossim(i);$
19. End for
20. End function

---

Algorithm 3 Hybrid Similarity

Algorithm 3 calculation is based on the results of cosine and soft-cosine similarity. To calculate firstly compute the size of the vector to find out the number of rows and columns. Next

**RESEARCH ARTICLE**

is to fetch the vectors  $v1$  and  $v2$  values from each word present in the rows using  $w2v()$ . Calculate similarity value by calling  $Calculatecossim()$  on  $v1$  and  $v2$  values. Calculate Soft Cosine similarity value by calling  $Calculatesoftcosine()$  on  $v1$  and  $v2$  values. And at last, calculate the hybrid similarity by adding the average of cosine and soft cosine similarity for each row.

In Figure 6 columns 1, 2, 3 represent cosine, soft-cosine, and hybrid similarity calculation outcomes respectively. After applying Cosine, Soft Cosine, and hybrid similarity measures, the similarity index graphical representation is as shown in Figure 7.

dtp	1	2	3
1	0.71144	1.9851	2.6965
2	0.58426	2.5277	3.112
3	0.54093	4.3554	4.8963
4	0.5818	2.1891	2.7709
5	0.73933	2.0626	2.8019
6	0.47753	4.0862	4.5637
7	0.69349	2.093	2.7865
8	0.68527	2.5232	3.2085
9	0.7215	2.121	2.8425
10	0.62197	1.7704	2.3924
11	0.63983	3.2016	3.8414
12	0.58562	2.1394	2.7251
13	0.3668	0.15274	0.51954
14	0.63537	3.2175	3.8529
15	0.7287	3.1002	3.8289
16	0.68648	2.8575	3.544
17	0.46367	2.0984	2.5621
18	0.64726	2.5198	3.167
19	0.6095	2.1303	2.7398
20	0.29064	0.15057	0.44121

Figure 6 Outcomes after Similarity Measures

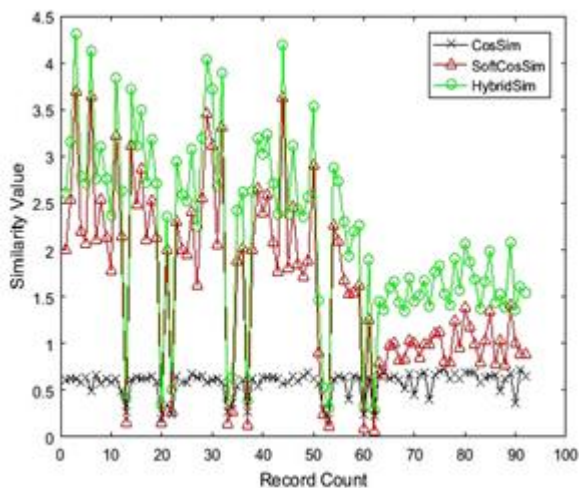


Figure 7 Similarity Index Graphical Representation

The resultant output for all tweets in the row is evaluated to obtain an average similarity index is then measured. The

formula used for cosine, soft cosine, and hybrid similarity index is represented by equations mentioned below:

$$\text{Cosine Similarity} = \sum_{i=1}^n \text{cosinesimilarity} / n \quad (1.1)$$

$$\text{Soft Cosine Similarity} = \sum_{i=1}^n \text{Softcosinesimilarity} / n \quad (1.2)$$

$$\text{Hybrid Similarity} = \text{Cosine} + \text{Soft Cosine} \quad (1.3)$$

$$\text{Set} = [\text{Cos}_i, \text{soft}_i, \text{hybrid}_i] \quad (1.4)$$

3.5. To Find Initial Centroid (IC) of Data

The next step is to determine, the Initial Centroid (IC) of the tweet data, which is obtained as the average of each similarity measure obtained from each similarity index (Cosine, soft cosine, and hybrid) individually. This value is taken as IC for the tweet. The formula used to determine the IC is given by equation (1.5).

$$\text{Initial Centroid (IC)} = \left[ \sum_{i=1}^k \frac{\text{All Cosine}}{K}, \sum_{i=1}^k \frac{\text{All softCosine}}{K}, \sum_{i=1}^k \frac{\text{All hybrid}}{K} \right] \quad (1.5)$$

Where, k is the total number of tweets in a given document.

As shown in Figure 8, the centroid of each cluster is represented by the "X" sign. The available data are grouped into three clusters named cluster1, cluster2, and cluster3, each represented by different colors the blue, red, and orange colors respectively.

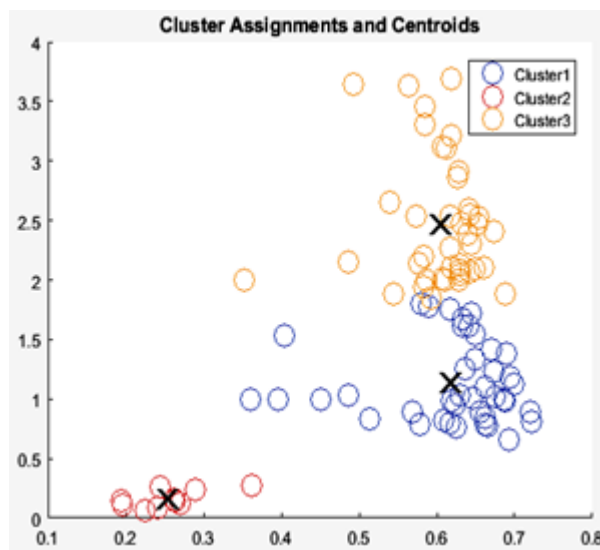


Figure 8 Cluster Assignments for Centroid

3.6. To Find Euclidian Distance of Each Tweet

Euclidean distance is the geometric distance in multidimensional space. The Euclidean distance between points T1 and T2 in n-dimensional space is calculated using the following formula (1.6). The formula used to calculate the



**RESEARCH ARTICLE**

Euclidian distance of each tweet from the set (st), which is calculated using equation (1.4) for cosine data, soft cosine, and hybrid data is represented by equations shown below:

For all St in sets calculate,

$$D1 = ECU1(IC, ST) \tag{1.6}$$

$$D2 = Squared ECU1 (IC, ST) \tag{1.7}$$

$$D3 = (D1 + D2)/2 \tag{1.8}$$

Note that the Euclidean distance (and its square) is calculated from the Tweet data obtained from the previous step.

After this, to determine the number of fragments, the total number of centroids in the given data has been calculated by measuring the average of D1, D2, and D3. i.e.

D=Total number of centroids

$$Calculate D = \frac{D1+D2+D3}{3} \tag{1.9}$$

An example of distance measured from st that is d1 is represented by Figure 9.

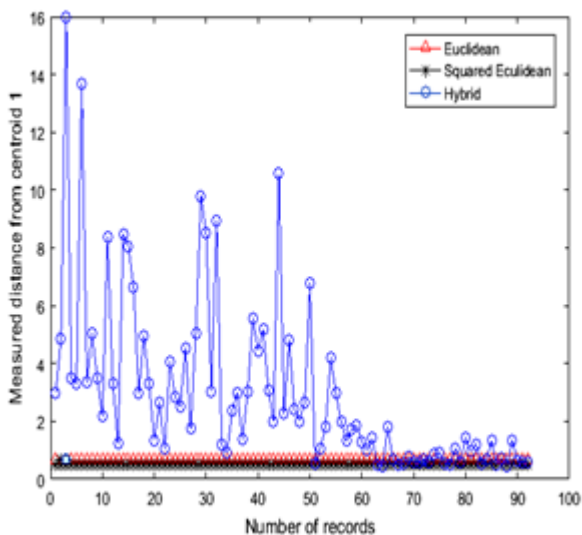


Figure 9 Measured Distances from Centroid 1

**3.7. Fragmentation**

Data fragmentation refers to dividing the data into segments so that the storage becomes easy. To determine the number of fragments from the available data, the formula used is written by equation (1.10).

$$p = \frac{\sqrt{D \times K}}{c} \tag{1.10}$$

Where K is the total number of rows.

**3.8. Neural Network**

The data obtained after fragmentation is passed along with the original word 2 vector data as input to the Artificial Neural Network (ANN). The used structure of ANN is given below (see Figure 10).

The classified fragments for 100, 200, 300, 400, and 500 rows are listed in Table 1 below.

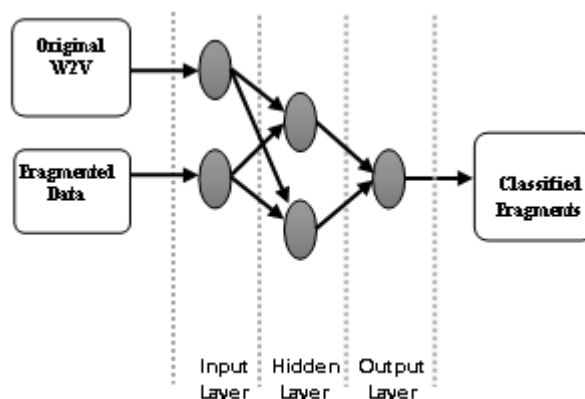


Figure 10 ANN Structure to Classify Fragmented Data

For 100 Rows	For 200 Rows	For 300 Rows	For 400 Rows	For 500 Rows
1	1	2	4	3
1	1	2	2	2
2	1	1	1	1
1	2	3	3	2
1	2	1	1	3
1	1	2	2	2
2	1	1	2	4
1	3	2	3	4
1	2	3	3	3
3	1	4	4	3
2	2	3	2	2

Table 1 Classification of Fragments for Each Row

**4. RESULTS AND DISCUSSIONS**

Fragmented architecture has been designed using MATLAB simulator with 4 GB RAM, 64-bit operating system, and a processor of 2.30 GHz. The performance has been analyzed in terms of the classified accuracy. The results have been evaluated individually, for 100, 200, 300, 400, and 500 rows. Experiments have been performed five times to determine the detection accuracy as depicted in Table 2. Figure 11 represents the classification accuracy of the designed fragmented structure. The simulations have been performed five times so that the exact accuracy for the uploaded data that might contain rows (100, 200, 300, 400, and 500). From the figure, it is observed that with the increase in rows, the classification accuracy increases. This is because with the

**RESEARCH ARTICLE**

increase in the data the ability to train ANN structure is increases that result in improved classification of the fragmented data. The average of the classification accuracy obtained for 100, 200, 300, 400, and 500 rows are 63.81, 76.28, 81.52, 83.58, and 92.078 respectively.

Iterations	100	200	300	400	500
1	62.45	75.89	82.15	84.57	91.04
2	63.57	76.28	81.27	83.57	92.57
3	62.78	76.18	79.68	84.25	93.17
4	64.28	75.12	80.15	82.37	92.14
5	65.97	77.94	84.36	83.14	91.47

Table 2 Classification Accuracy for Different Rows

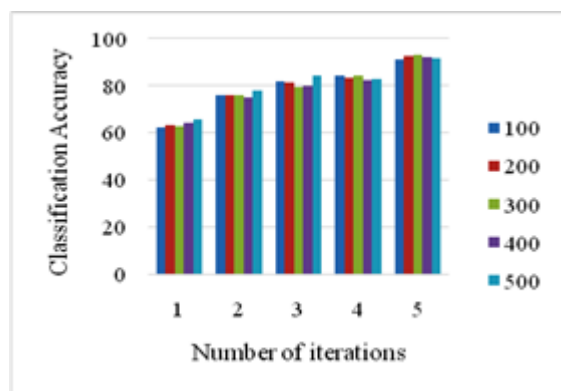


Figure 11 Classification Accuracy

To evaluate precision and recall average percentage of iterations is calculated as shown in Table 3. The fragments architecture created after the evaluation of the proposed work is further evaluated by parameters such as True Positive, True Negative, False Negative, and False Positive are shown in Table 4. Precision and Recall are shown in Table 5.

True Positive (Tp) =  
 $\text{Total true selected elements} / \text{Total sample size}$  (1.11)

False Positive (Fp) =  
 $\text{False selected elements} / \text{Total sample size}$  (1.12)

True Negative (Tn) =  
 $\text{True left samples} / \text{Total Sample Size}$  (1.13)

False Negative (Fn) =  
 $\text{False left sample} / \text{Total Sample Size}$  (1.14)

Table 5 showing the precision and recall and found that they are almost the same for every row passed and Figure 12 shows the graphical representation of the precision and recall. The Fp value reveals that the components are not put in fitting clusters. The Fp outcomes in this work are lower. Tn showing

bad samples that are left. If it is high it indicates a good search response. Depends on the value calculated for T<sub>p</sub>, T<sub>n</sub>, F<sub>p</sub>, and F<sub>n</sub> precision and recall values are calculated.

No. of Rows/Records	Average of all five iteration in % (approx)	Remaining average (%)
100	64	36
200	76	24
300	82	18
400	84	16
500	92	8

Table 3 Average Percentage of Iterations

Rows	100	200	300	400	500
Total true selected samples (T <sub>p</sub> )	0.45	0.53	0.57	0.59	0.64
True left samples (T <sub>n</sub> )	0.25	0.17	0.13	0.11	0.11
False left samples (F <sub>n</sub> )	0.44	0.53	0.57	0.58	0.64
false selected sample (F <sub>p</sub> )	0.25	0.08	0.04	0.03	0.01

Table 4 Evaluation of T<sub>p</sub>, F<sub>p</sub>, T<sub>n</sub>, and F<sub>n</sub>

Total Passed Rows	100	200	300	400	500
Precision	0.64	0.87	0.93	0.95	0.98
Recall	0.64	0.76	0.82	0.84	0.85

Table 5 Evaluation of Precision and Recall

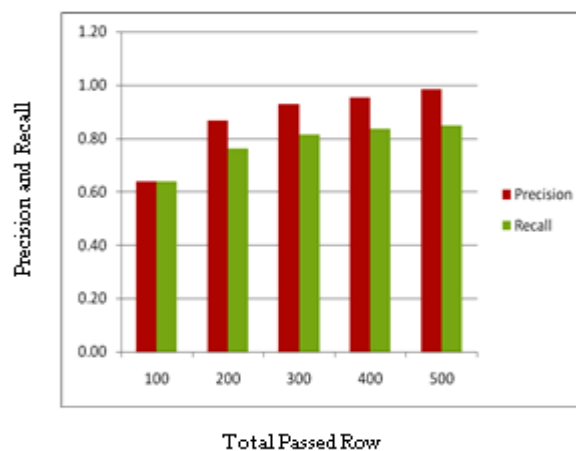


Figure 12 Precision and Recall

**RESEARCH ARTICLE**

**5. COMPARATIVE ANALYSIS OF PROPOSED AND EXISTING WORK**

The maximum attained precision is 0.98 for total passed rows of 500. The recall value for every row passed is 0.85. It is found that an enhancement in proposed work in the case of precision and recall is seen. Earlier researchers used the k-means dependent cosine similarity measurement method to determine the feature similarity between the cluster centroids and the data points to quantify the similarity between the outcome of the clustering and the side details. A clustering algorithm for data partitioning using the principle of a learning method has been proposed. The main drawback of this proposed work is that only cosine similarity based k-means have been used for partitioning large data sets [33]. An effort on the hybridization of cosine and soft-cosine is also carried out to improve precision and recall parameters during partitioning [34]. In this research, we have proposed cosine, soft cosine, and hybrid similarity as an enhanced mechanism and achieved an increase of 0.98 precision and 0.85 recall values as outcome. So, we rely on this technique for fragmenting the text data for a diverse system. Table 6 shows the comparison of calculated precision and recall. Figure 13 shows the Graphical Views of Precision and Recall

Parameters	Huaping Guo et al. [33]	Kiranjeet Kaur et al. [34]	Proposed Work
Precision	0.47	0.69	0.98
Recall	0.54	0.67	0.85

Table 6 Comparison of calculated Precision and Recall

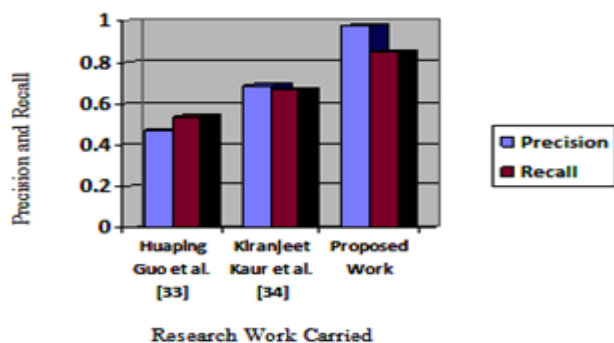


Figure 13 Graphical Views of Precision and Recall

Proposed works has many advantages over existing methodologies and are following as:

- It helps to improve the classification of fragmented data with high precision and recall and indicates maximum coverage, accuracy, and reduce overall computational time.

- Earlier techniques depends on cosine based k-means, cosine and soft cosine hybridization for clustering were not effective due to lack of balance in the quality and efficiency of clustering in categorical data sets.
- The principles of existing approaches are fine for some case, but not applicable. So algorithm hybridization is the best approach. Cosine and soft cosine similarity notions are used to compute hybrid similarity in this research work to ensure improved efficiency and can easily dealing with large data sets.

**6. CONCLUSION**

In this paper, novel relative data fragmentation architecture is proposed to divide the large dataset into different fragments. Here, twitter data is being applied for experiment purposes and converted into vectors to use it for fragmentation purposes. Cosine, soft cosine and hybrid similarity calculation is calculated and centroid positions are discovered. K-Mean algorithm is used to calculate the distance between data points with each centroid to discover clusters. At last, validation and performance is performed by checking its accuracy using ANN. In this research, efforts are given to introduce novel similarity based data fragmentation architecture in an unsupervised learning environment. Comparison is performed and attained high precision and recall as compared to the existing proposed methods.

Future work is stress on the allocation and deduplication of data to strengthen the wide distributed network environment. So, that most of the user’s requirements can be satisfied and also work for the enhancement of various parameters such as cost, delay factors, duplication of data.

**REFERENCES**

- [1] Tarun S., Batth R. S. (2019). Distributed Database Design Challenges and its Countermeasures-A Study. Journal of the Gujarat Research Society 21 (6), pp. 875-886
- [2] S. Tarun, R. S. Batth and S. Kaur, "A Review on Fragmentation, Allocation and Replication in Distributed Database Systems," 2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE), Dubai, United Arab Emirates, 2019, pp. 538-544, doi: 10.1109/ICCIKE47802.2019.9004233
- [3] R. Singh and K. S. Mann, "Improved TDMA Protocol for Channel Sensing in Vehicular Ad Hoc Network Using Time Lay," Proceedings of 2nd International Conference on Communication, Computing and Networking Lecture Notes in Networks and Systems, pp. 303-311, 2018.
- [4] A. Nayar, R. S. Batth, D. B. Ha, and G. Sussendran, G. "Opportunistic networks: Present scenario-A mirror review" International Journal of Communication Networks and Information Security," 10 (1), pp. 223-241, 2018.
- [5] G.S Shahi, R.S Batth, S. Egerton, 2020 "MRGM: An Adaptive Mechanism for Congestion Control in Smart Vehicular Network", International Journal of Communication Networks and Information Security 12 (2).
- [6] Qi, H., & Gani, A. (2012, May). Research on mobile cloud computing: Review, trend and perspectives. In 2012 Second International

**RESEARCH ARTICLE**

Conference on Digital Information and Communication Technology and its Applications (DICTAP), IEEE, pp. 195-202.

[7] Venters, W., & Whitley, E. A. (2012). A critical review of cloud computing: researching desires and realities. *Journal of Information Technology*, 27(3), pp. 179-197.

[8] Borkar, V., Deshmukh, K., & Sarawagi, S. (2001, May). Automatic Segmentation of text into structured records. In *Proceedings of the 2001 ACM SIGMOD international conference on Management of data*, pp. 175-186.

[9] Santini, S., & Jain, R. (1999). Similarity measures. *IEEE Transactions on pattern analysis and machine Intelligence*, 21(9), pp. 871-883.

[10] Huang, A. (2008, April). Similarity measures for text document clustering. In *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008)*, Christchurch, New Zealand ,Vol. 4, pp. 9-56.

[11] Sidorov, G., Gelbukh, A., Gómez-Adorno, H., & Pinto, D. (2014). Soft similarity and soft cosine measure: Similarity of features in vector space model. *Computación y Sistemas*, 18(3), pp. 491-504.

[12] Michie, D., Spiegelhalter, D. J., & Taylor, C. C. (1994). *Machine learning. Neural and Statistical Classification*, 13(1994), pp. 1-298.

[13] Ikonomakis, M., Kotsiantis, S., & Tampakas, V. (2005). Text classification using machine learning techniques. *WSEAS transactions on computers*, 4(8), pp. 966-974.

[14] Verma and A. Kumar, "Performance Enhancement of K-Means Clustering Algorithms for High Dimensional Data sets", *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol. 4, No. 1, pp.5-9, 2014.

[15] Z.Tao, H. Liu, H. Fu and Y.Fu, "Image Co-segmentation via Saliency-Guided Constrained Clustering with Cosine Similarity", *AAAI*, pp. 4285-4291, 2017

[16] X. Gu, H. Zhang and S. Kim, "Deep code search", In *Proceedings of the 40th International Conference on Software Engineering*, ACM, pp. 933-944, 2018.

[17] W. L. Xiang, Y. Z. Li, R. C. He, M.X. Gao, M.Q. An, "A novel artificial bee colony algorithm based on the cosine similarity", *Computers & Industrial Engineering*, Vol. 115, pp.54-68, 2018.

[18] Wiese, L. (2014). Clustering-based fragmentation and data replication for flexible query answering in distributed databases. *Journal of Cloud Computing* 3, 18. <https://doi.org/10.1186/s13677-014-0018-0>

[19] Ali A. Amer, Adel A. Sewisy, Taha M.A. Elgandy. (2017). An optimized approach for simultaneous horizontal data fragmentation and allocation in Distributed Database Systems (DDBSs). *Heliyon* 3 e00487. doi: 10.1016/j.heliyon.2017. e00487

[20] Abdalla, H., & Artoli, A. M. (2019). Towards an efficient data fragmentation, allocation, and clustering approach in a distributed environment. *Information*, 10(3), 112. <https://doi.org/10.3390/info10030112>

[21] Rahimi, H., Parand, F. A., & Riahi, D. (2018). Hierarchical simultaneous vertical fragmentation and allocation using modified Bond Energy Algorithm in distributed databases. *Applied computing and informatics*, 14(2), pp. 127-133. <https://doi.org/10.1016/j.aci.2015.03.001>

[22] Lim, S., Ng, Y. (2001). A Hybrid Fragmentation Approach for Distributed Deductive Database Systems. *Knowledge and Information Systems* 3, pp. 198–224. <https://doi.org/10.1007/PL00011666>

[23] Khan S. I., (2016). Efficient Partitioning of Large Databases without Query Statistics", *Database System Journal*, pp. 34-53.

[24] Peng, P., Zou, L., Chen, L., & Zhao, D. (2019). Adaptive distributed RDF graph fragmentation and allocation based on query workload. *IEEE Transactions on Knowledge and Data Engineering*, 31(4), pp.670-685. <https://doi.org/10.1109/TKDE.2018.2841389>

[25] Aloini, D., Benevento, E., Stefanini, A., & Zerbino, P. (2020). Process fragmentation and port performance: Merging SNA and text mining. *International Journal of Information Management*, 51, 101925. <https://doi.org/10.1016/j.ijinfomgt.2019.03.012>

[26] Memmi, G., Kapusta, K., & Qiu, H. (2015, August). Data protection: Combining fragmentation, encryption, and dispersion. In *2015 International Conference on Cyber Security of Smart Cities, Industrial Control System and Communications (SSIC)* (pp.1-9). IEEE. <https://doi.org/10.1109/SSIC.2015.7245680>

[27] Links: <https://www.kaggle.com/soaxelbrooke/first-inbound-and-response-tweets/data?select=sample.csv>

[28] Links: <https://gist.github.com/larsyencken/1440509>

[29] Lende, S. P., & Raghuvanshi, M. M. (2016, February). Question answering system on education acts using NLP techniques. In *2016 world conference on futuristic trends in research and innovation for social welfare (Startup Conclave)* (pp. 1-6). IEEE.

[30] Zeyu, X., Qiangqian, S., Yijie, W., & Chenyang, Z. (2018). Paragraph vector representation based on word to vector and CNN learning. *Computers, Materials & Continua*, 55(2), pp. 213-227.

[31] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[32] Bartunov, S., Kondrashkin, D., Osokin, A., & Vetrov, D. (2016, May). Breaking sticks and ambiguities with adaptive skip-gram. In *artificial intelligence and statistics*, pp. 130-138.

[33] H. Guo, J. Zhou and C.A. Wu (2018), "Imbalanced Learning Based on Data-Partition and SMOTE", *Information*, Vol. 9, No. 9, pp. 238.

[34] Kaur K., Laxmi V. (2019), "Hierarchical Clustering Based Improved Data Partitioning using Hybrid Similarity Measurement Approach", *International Journal of Innovative Technology and Exploring Engineering*, Volume-8 Issue-8, pp. 3008-2014.

**Authors**



**Mr. Sashi Tarun** is a PhD. Research Scholar in the School of Computer Science And Engineering at Lovely Professional University, Punjab, India. He has completed M.Tech. Computer Science from Jamia Hamdard University, New Delhi. His research interests are Distributed Systems, Cloud Systems, Database System, Computer Networks, AI, and Machine Learning. He has number of papers in his credit. He has 7 years of teaching experience as

Assistant Professor.



**Dr. Ranbir Singh Batth** is working as an Associate Professor in the School of Computer Science and Engineering and he also serves as an International coordinator for at Lovely Professional University, Punjab, India. He has received his Ph.D. from IKG Punjab Technical University, Kapurthala, Punjab, India in 2018 and the Master degree in Computer Engineering from Punjabi University, Patiala. His research interests include Wireless Sensor Networks, Cloud Computing, Network Security, Ad Hoc Networks, IoT, Machine Learning, Deep Learning, Wireless Communications and Mobile computing. He also serves as an editorial member, guest editor, and reviewer for various reputed International journals. He has been the organizing chair, session chair and advisory member for various reputed International conferences. He is an active member of ACM and IEEE computer Society.



**Dr. Sukhpreet Kaur** is working as Associate Professor in CSE department at Chandigarh Engineering College, Landran, Mohali. She has in total of 15 years of vast experience in teaching and research. She has done Ph.D in CSE from I K Gujral Punjab Technical University, Jalandhar and has done her Masters in Technology in CSE from GNDEC, Ludhiana. The various research areas in which she worked includes Image Processing, Artificial Intelligence and Computer Vision.