IADIS International Journal on Computer Science and Information Systems
Vol. 12, No. 1, pp. 88-108
ISSN: 1646-3692

A SOFT SIMILARITY MEASURE FOR K-MEANS BASED HIGH DIMENSIONAL DOCUMENT CLUSTERING

T. V. Rajinikanth[1] and G. Suresh Reddy[2]
[1]Department of Computer science and Engineering, Sreenidhi Institute of Science and Technology, Hyderabad, India
[2]Department of Information Technology, VNR VJIET, Hyderabad, India

ABSTRACT

Feature dimensionality has always been one of the key challenges in text mining as it increases complexity when mining documents with high dimensionality. High dimensionality introduces sparseness, noise, and boosts the computational and space complexities. Dimensionality reduction is usually addressed by implementing either feature reduction or feature selection techniques. In this work, the problem of dimensionality reduction is addressed using singular value decomposition and the results are compared to information gain approach through retaining top-k features. High dimensional clustering is carried by using k-means algorithm with gaussian function. The proposed dimensionality reduction and clustering approaches are compared to conventional approaches and results prove the importance of our approach.

KEYWORDS

Feature Selection, Feature Reduction, Clustering, Classification, Dimensionality

1. INTRODUCTION

Mining text documents using unsupervised learning has the prime challenge from the document dimensionality. In this regard, methods used to reduce the dimensionality have received wide attention from text mining researchers. Dimensionality is addressed in literature using two methods, Feature reduction and Feature selection (Hawashin, Mansour, & Shadi, 2013). In the feature extraction process also called feature reduction, the high dimensional text documents are projected onto their corresponding low dimensional representation in feature space using algebraic rules and transformations. The objective is to find an optimal transformation matrix corresponding to the input high dimensional document feature matrix (Lam Hong Lee et al., 2012).

The objective of present work is to apply basic gaussian function to k-means algorithm to cluster text documents and perform dimensionality reduction applying SVD. The SVD approach is extended to determine top-k important features. The application of gaussian membership function to k-means for document clustering is inspired from contribution of authors (Jung-Yi Jiang, Ren-Jia Liou, & Shie-Jue Lee, 2011).The membership function used retains the original distribution of words in documents. In the feature selection process, we reduce the feature set W, consisting of w features to a new reduced word set W', consisting of r features, where | r |<| w |. These set of features denoted by W', are later used when performing classification and clustering (Jing Gao & Jun Zhang, 2005; Hussein Hashimi, Alaaeldin Hafez, & Hassan Mathkour, 2015). The transformed view of document corpus using feature extraction process is depicted pictorially using figure 1. The paper is organized as follows. Related works are reviewed in Section 2, a review on feature selection using SVD for dimensionality reduction is carried out in Section-3. Section-4 introduces proposed approach. Results are discussed in Section-5. The paper is concluded in Section-6.

2. RELATED WORKS

In G. SureshReddy, Rajinikanth.T. V. & Ananda Rao (2014); Gangin Lee & Unil Yun (2017) frequent item sets are obtained from considered text corpus. These obtained frequent item sets are later used to form the feature set. This feature set is used to form the document-feature matrix and a document to document similarity is generated (Chintakindi Srinivas, Vangipuram Radhakrishna & C.V. Guru Rao, 2013). The similarity matrix is used to cluster documents. The design and analysis of similarity measure (Suresh Reddy, 2016 and Chintakindi Srinivas, Vangipuram Radhakrishna & C.V. Guru Rao, 2015) is based on considering word distribution. The similarity measure is based on feature function. To enhance clustering quality, (Shady Shehata, Fakhri Karray, & Mohamed Kamel, 2010) proposes concept based mining. Yanjun Li, Congnan Luo, Chung, & S. M. (2008), makes use of statistical data and uses this data to select suitable features from documents which to yield better clustering results. (Shuigeng Zhou & Jihong Guan, 2002) discuss approach for increasing efficiency of text classification.

Text clustering algorithms usually suit non-distributed environments, and usually are not that compatible to distributed scenario. In O. Papapetrou, W. Siberski & N. Fuhr (2012), authors propose a decentralized probabilistic clustering approach that suits distributed clustering. A fuzzy algorithm (Andrew Skabar & Khaled Abdalgader, 2013) is proposed that aims at generating sentence-level clusters. Authors (Shie-Jue Lee, Jung-Yi Jiang, 2014) introduces, multi-label text clustering using fuzzy logic. An extended and improved similarity measure is proposed in G. Suresh Reddy, A. Ananda Rao, & T. V. Rajinikanth (2015), which improves similarity values when compared to Suresh Reddy et al. (2014). SVD based clustering is adopted in Jing Gao et al. (2005); Suresh Reddy et al. (2015). Text classification such as rough sets based (Libiao Zhang, Yuefeng Li, Chao Sun, & Wanvimol Nadee, 2013), hybrid (Chin Heng, 2012), extracting group features by substring approach (Dell Zhang & Wee Sun Lee, 2006), SVM based (Jung Yi Jiang, 2011; Wen Zhang, Taketoshi Yoshida, & Xijin Tang, 2008) are some of the related works. Application to text mining w.r.t market prediction is discussed in Arman, Saeed, Ying Wah, & David (2014). In Sajid Mahmood, Muhammad Shahbaz, & Aziz Guergachi (2014), association rules (positive and negative) are extracted from text corpus and used to text clustering. A text clustering approach using generated similarity matrix is proposed in Wen Zhang (2010). Application of text clustering to pattern discovery is done in Ning Zhong, Yuefeng Li, & Sheng-Tang Wu (2012). Dimensionality and selection criteria to obtain, quality text classification and improved clustering are studied in Hussein, (2015); Fodor (2002); Christopher (2009); Sunghae Jun, Sang-Sung Park, & Dong-Sik Jang, (2014) and Jung (2011). A review on gene classification is carried in Shadi Aljawarneh & Bassam Al-shargabi (2013).

In the present work, our idea is to design a similarity measure overcoming dis-advantages in Euclidean, Cosine, Jaccard distance measures (Yung-Shen Lin, Jung-Yi Jiang & Shie-Jue Lee, 2014). The proposed measure considers distribution of features of the global feature set. Applications of similarity measure in software reuse are addressed by Chintakindi et al. (2013); Chintakindi et al. (2014); Chintakindi et al. (2015). Approach for clustering users based on transactions is given in M. S. B. Phridviraj, Vangipuram RadhaKrishna, Chintakindi Srinivas, & C.V. GuruRao (2015). Temporal pattern mining in time stamped temporal databases require similarity functions which can handle supports expressed as vectors. Novel similarity measures for temporal context are proposed in Vangipuram Radhakrishna, P. V. Kumar, & V. Janaki, (2015); Vangipuram Radhakrishna, P.V.Kumar, & V.Janaki (2016); Shadi A. Aljawarneh, Vangipuram Radhakrishna, P.V.Kumar, & V. Janaki (2017); Vangipuram Radhakrishna, Shadi A. Aljawarneh, Puligadda Veereswara Kumar, & Kim-Kwang Raymond Choo (2016) and V. Radhakrishna, P. V. Kumar, V. Janaki & S. Aljawarneh (2016). Applications of similarity measures to medical data mining are addressed in Shusaku Tsumoto, Haruko Iwata, Shoji Hirano, Yuko Tsumoto (2014). An application of text mining is discussed in the context of Arabic text in Nafaa Haffar et al. (2017). Semantic kernel for text classification is proposed in Berna Altınel, Murat Can Ganiz, & BanuDiri (2015) which is based on text corpus. Data mining and text mining principles are applied for intrusion detection in Gunupudi Rajesh Kumar, N. Mangathayaru, & G. Narsimha (2015); Shadi A. Aljawarneh, Raja A. Moftah, Abdelsalam M. Maatuk (2016); Gunupudi Rajesh Kumar et al. (2016); Shadi A. Aljawarneh et al. (2011). Reuse of resources in e-learning systems almost become impossible because of bad indexing. N. Haffar, M. Maraoui & S. Aljawarneh (2016) addresses this issue by proposing a dynamic system which uses natural language tools SAPA and AL-KHALIL. Ons Meddeb, Mohsen Maraoui, & Shadi Aljawarneh (2016) proposes a new model for AHRS (Arabic hand recognition system). The system is modeled considering preprocessing, segmentation, features extraction, classification and post-processing. Recognizing research trends by mining data obtained using search engines has been studied in Mohammed R Elkobaisi, Abdelsalam M Maatuk & Shadi Aljawarneh (2015).

3. FEATURE SELECTION USING SVD

Feature dimensionality has always been one of the key challenges in text mining as it increases complexity when mining documents with high dimensionality. High dimensionality introduces sparseness, noise, and boosts the computational and space complexities. Dimensionality reduction is usually addressed by implementing either feature selection or feature reduction techniques. For feature selection, SVD (singular value decomposition) and IG (Information gain) approaches are adopted through retaining top-k features to compare the dimension reduction achieved. Section 3.1 below outlines the dimensionality reduction using singular value decomposition.

3.1 Dimensionality Reduction Using SVD

The feature selection using singular value decomposition, is achieved using the procedure discussed below

1. Initialize the text corpus obtained after the preprocessing stage as a feature document matrix representation. This feature document matrix is denoted as [M]dxw.
2. Apply the method of SVD to transform the initial matrix,[M]dxw to its equivalent real valued matrix factorization consisting element matrices, document-document, feature-feature and singular value matrices.
3. In singular matrix obtained after decomposition, choose only elements of the first column. Each column in the singular matrix represents a column vector of dimension d, equal to the number of documents in the text corpus. ’w’ columns are available in the singular matrix.
4. To perform feature selection, first sort the obtained column vector in step-3. Feature selection is done by selecting only the top-k features from the sorted column vector. The top- k features are those with significant Eigen values which add up to 90% total energy. Also, features whose Eigen values are less than unity may be neglected as they do not affect the learning approaches.
5. Consider these top-k features and form the final global feature set.

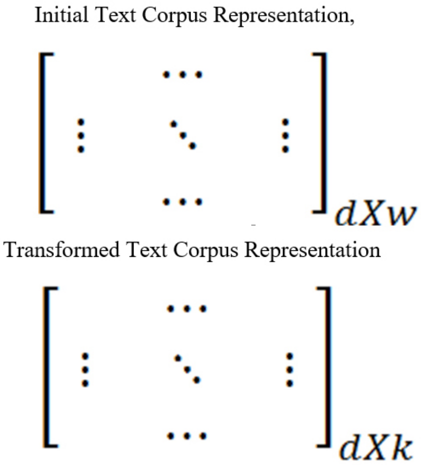
Initial Text Corpus Representation,
dXw
Transformed Text Corpus Representation
dXk

Figure 1. Feature Reduction

4. PROPOSED METHOD

In this section, proposed algorithm to cluster the corpus of text documents using fuzzy Gaussian membership function is discussed. Documents are expressed in terms of their feature probabilities. The total feature probability of each feature with respect to all documents is 1.

4.1 Algorithm

Algorithm for clustering using fuzzy similarity measure based K-means

Input : Text Corpus with documents expressed in terms of word probability
Output : K-Generated Clusters
Measure : Fuzzy Gaussian Membership Function

Step-1: Decide the number of clusters required. Call this value as k.

Choose the number of clusters required.

Step-2: Choose the k-appropriate cluster center.

Choose any random document, say document1 and compute similarities between document1 and all other documents in the corpus to be clustered and then choose the top k-1 least similar documents. The cluster center is now document vectors of the initial randomly chosen document say document1 and these top k-1 documents.

Step-3: Initialize mean and deviation for k-clusters. Initialize iteration I=0

Initialize k-cluster with document vectors as their cluster center and choose the standard deviation to some fuzzy value preferably not zero and lies between 0 and 1.

Step-4: Obtain similarity between each document to these k-cluster centers.

Find fuzzy similarity between each document from the text corpus and k-cluster centers. The document is moved to cluster to which it shows maximum fuzzy similarity value. This finishes one iteration, update I by incrementing it by 1 i.e. I= I+1

Step-5: Update cluster center for these k-clusters.

If clusters formed in the previous iteration are different to clusters obtained at the end of iteration, repeat step-5. Once the documents are assigned to clusters, at the end of iteration, compute the average of document vectors assigned and are grouped to a specific cluster. This value becomes mean of the corresponding cluster or clusters. If no change exist in the cluster configuration, than stop generating clusters.

Step-6: Compute similarity of each document from text corpus to updated k-cluster centers.

Find fuzzy similarity between each document from the text corpus and these updated k-cluster centers. The document is moved to cluster to which it shows maximum fuzzy similarity value. Update I by 1 i.e. I = I+1, go to step-5.

Step-7: Stop the clustering when clusters do not change between successive iterations.

Clusters generated are the final k-clusters.

The input to the fuzzy k-means algorithm is the text corpus with documents expressed in terms of feature probabilities and output is k-clusters generated. Since the measure computes similarity, choose the maximum value for deciding the choice.

4.2 Fuzzy Membership Function

The standard Gaussian function is adopted for clustering documents in incremental approach. The membership function defines the similarity degree of document doc_i to a given cluster say ‘g’. The Gaussian membership function is given by equation 1.

$$\mu_k{}^{i,g} = \prod_{k=1}^{k=m} e^{-(\frac{doc_i^k - mean_g^k}{deviation})^2} \qquad (1)$$

Here k represents, kth feature considered. doc_i^k refers to kth feature value in document i, $mean_g^k$ refers to mean of kth feature in document 'i'. We use the membership function which is standard Gaussian distribution function for computing similarity between cluster feature and new document corresponding feature. The notion, π in equation (1) represents product considered for all k features ($1 \leq k \leq m$) where 'm' denotes number of features.

5. CASE STUDY

5.1 Text Clustering

Consider the sample corpus obtained after dimensionality reduction as shown in Table 1.

Table 1. Sample Text Corpus

	Features			
Documents	w1	w2	w3	Class
Document 1	1	2	1	$C^{(1)}$
Document 2	0	1	3	$C^{(1)}$
Document 3	0	1	0	$C^{(1)}$
Document 4	1	5	2	$C^{(1)}$
Document 5	2	0	1	$C^{(2)}$
Document 6	5	0	1	$C^{(2)}$
Document 7	10	0	2	$C^{(2)}$
Document 8	3	0	1	$C^{(2)}$
Document 9	4	0	0	$C^{(2)}$

Table 2 shows documents expressed in terms of word probabilities. The total probability of each word with respect to all documents of the corpus is always equal to 1. This property of word probabilities is used for clustering using fuzzy measure. Table 3 shows computations of similarity between documents, document 1 to all other documents. i.e document 2 through document 9. The document 7 in Table 3 shows least similarity to document 1. Hence, this forms the best candidate for another cluster center. The similarity is computed using proposed fuzzy membership function.

Table 2. Modified Text Documents

	Features probabilities		
Documents	w1	w2	w3
Document 1	0.038462	0.222222	0.090909
Document 2	0	0.111111	0.272727
Document 3	0	0.111111	0
Document 4	0.038462	0.555556	0.181818
Document 5	0.076923	0	0.090909
Document 6	0.192308	0	0.090909
Document 7	0.384615	0	0.181818
Document 8	0.115385	0	0.090909
Document 9	0.153846	0	0

Table 3. Choosing Cluster Centers

Documents	Document 1
Document 2	0.829
Document 3	0.9154
Document 4	0.6203
Document 5	0.816
Document 6	0.7465
Document 7	0.4916
Document 8	0.801
Document 9	0.7528

Table 4. Initial clusters With Centers

Clusters	Documents	w1	w2	w3
Cluster 1	Document 1	0.038462	0.222222	0.090909
Cluster 2	Document 7	0.384615	0	0.181818

We get two clusters, cluster 1 and cluster 2 with document 1 belonging to cluster 1 and document 7 belonging to cluster 2. This is represented in Table 4. This finishes the initialization and become the starting point for clustering. Table 5, shows the computation of similarities of documents in corpus to both clusters. Now document 1, document2, document 3, document 4, document 8, document 9 are grouped to cluster 1 while document 6, document 7 are grouped to cluster 2. Call this one as first iteration, say iteration I=1.

Table 5. Similarity Computations to Initial Clusters, Iteration-1

	Cluster-1	Cluster-2	Decision
Document 1	1	0.4914	Cluster 1
Document 2	0.829	0.509	Cluster 1
Document 3	0.9154	0.4614	Cluster 1
Document 4	0.6202	0.1801	Cluster 1
Document 5	0.8159	0.6624	Cluster 1
Document 6	0.7465	0.8344	Cluster 2
Document 7	0.4916	1	Cluster 2
Document 8	0.801	0.723	Cluster 1
Document 9	0.7528	0.708	Cluster 1

Table 6. Similarity Computations to Initial Clusters, Iteration-2

	Cluster-1	Cluster-2	Decision
Document 1	0.9725	0.6339	Cluster 1
Document 2	0.8757	0.6334	Cluster 1
Document 3	0.94	0.6334	Cluster 1
Document 4	0.4928	0.2247	Cluster 1
Document 5	0.9199	0.8292	Cluster 1
Document 6	0.8591	0.9557	Cluster 2
Document 7	0.5907	0.9557	Cluster 2
Document 8	0.909	0.8797	Cluster 2
Document 9	0.8523	0.8634	Cluster 2

Since there is a change in configuration of clusters, proceed to next iteration. This is done to verify if clusters are consistent or not. At the end of second iteration, Iteration-2 has clusters updated and varying once again. So, proceed to one more iteration. At the end of second iteration, document 1, document 2, document 3, document 4, document 5, document 8 are grouped to cluster 1 while document 6, document 7 and document 9 are grouped to cluster 2. Call this as second iteration, say iteration I=2. This is represented in Table 6.

Table 7. Similarity Computations to Initial Clusters, Iteration-3

	Cluster-1	Cluster-2	Decision
Document 1	0.983	0.693	Cluster 1
Document 2	0.893	0.657	Cluster 1
Document 3	0.92	0.726	Cluster 1
Document 4	0.538	0.237	Cluster 1
Document 5	0.887	0.894	Cluster 2
Document 6	0.817	0.989	Cluster 2
Document 7	0.5557	0.8934	Cluster 2
Document 8	0.874	0.936	Cluster 2
Document 9	0.804	0.936	Cluster 2

Table 6 shows computation of similarities of documents in corpus to both updated clusters at the end of iteration-2. Now documents document1, document 2, document 3, document 4 are grouped to cluster 1 while document 5, document 6, document 7, document 8, document 9 is grouped to cluster 2. Call this as third iteration, say iteration I=3. This is shown in table 7. Since clusters are changed once again, proceed for next iteration. Since clusters generated at Iteration-4 also remain same, stop here and declare clusters formed as final clusters. Table 8 show the final updated clusters using proposed method. Initially had two classes with document 1, document 2,document 3, document 4 categorized as class-1 and document 5, document 6, document 7, document 8,document 9 are categorized to class-2. The obtained clusters have the same documents, which justifies the efficiency and accuracy of the approach.

Table 8. Initial Clusters with Centers

Clusters	Documents
Cluster 1	1,2,3,4
Cluster 2	5,6,7,8,9

5.2 Dimensionality Reduction

Let M be the sample document feature matrix which is considered for dimensionality reduction. The order of the matrix is 9 X10. Here, consider the same input document set used in the Table 9 to demonstrate the dimensionality reduction process using singular value decomposition. The sample input is given below for the sake of convenience.

The word document matrix in Table 9 consist of 9 documents, 10 features and 2 classes. The initial feature set contains ten terms W = {campus, building, lane, floor, relax, food, colony, web, WC, fridge}. The feature set size is 10. This is treated as an initial global vector. The initial feature set is obtained after preprocessing the text corpus set consisting these nine documents. The objective is to see the possibility of dimensionality reduction and at the same time, also retain the dominant and significant features eliminating least dominating features called outliers.

Table 9. Word Document Matrix

	Features or words										
Doc	**$w^{(1)}$ campus**	**$w^{(2)}$ building**	**$w^{(3)}$ lane**	**$w^{(4)}$ floor**	**$w^{(5)}$ relax**	**$w^{(6)}$ food**	**$w^{(7)}$ colony**	**$w^{(8)}$ web**	**$w^{(9)}$ wc**	**$w^{(10)}$ fridge**	**Class**
$D^{(1)}$	0	1	0	0	1	1	0	0	0	1	$C^{(1)}$
$D^{(2)}$	0	0	0	0	0	2	1	1	0	0	$C^{(1)}$
$D^{(3)}$	0	0	0	0	0	0	1	0	0	0	$C^{(1)}$
$D^{(4)}$	0	0	1	0	2	1	2	1	0	1	$C^{(1)}$
$D^{(5)}$	0	0	0	1	0	1	0	0	1	0	$C^{(2)}$
$D^{(6)}$	2	1	1	0	0	1	0	0	1	0	$C^{(2)}$
$D^{(7)}$	3	2	1	3	0	1	0	1	1	0	$C^{(2)}$
$D^{(8)}$	1	0	1	1	0	1	0	0	0	0	$C^{(2)}$
$D^{(9)}$	1	1	1	1	0	0	0	0	0	0	$C^{(2)}$

Stage-1: Apply SVD

On applying SVD, three matrices are obtained, left singular matrix called document x document matrix, Eigen value matrix which is a diagonal matrix and the third matrix called as right singular matrix which provide the real valued matrix factorization denoted as M = [document matrix] X [eigen values] X [word matrix]^{T}. Table 10, Table 11 and Table 12 gives document to document, Eigen value matrix and word to word matrices. Table 13 and Table 14 gives the Eigen values before and after sorting.

Table 10. Matrix U (Document X Document Matrix)

-0.1349	-0.2729	-0.1618	-0.4403	0.7357	-0.0484	-0.2167	-0.2213	-0.2186
-0.1525	-0.4215	0.6911	-0.0219	-0.1221	-0.3667	0.3545	-0.0364	0.2114
-0.0137	-0.1428	-0.0284	0.1846	-0.2607	-0.236	-0.2333	-0.7336	-0.4778
-0.1920	-0.7976	-0.3913	0.1966	-0.1377	0.1438	-0.1904	0.2228	0.0982
-0.1642	-0.0298	0.4653	0.1179	0.332	0.4638	-0.5669	-0.2121	0.2238
-0.3748	0.0532	-0.0049	-0.7739	-0.4096	-0.0031	-0.2974	0.0224	0.0319
-0.7796	0.2643	-0.0444	0.3361	0.1631	-0.3178	-0.0662	0.221	-0.1632
-0.2689	-0.0040	0.1456	0.0412	-0.2158	0.6781	0.4732	-0.0094	-0.4176
-0.2720	0.1330	-0.3204	0.0738	-0.076	0.1047	0.3139	-0.5171	0.6462

Table 11. Eigen Value Matrix S

6.33	0	0	0	0	0	0	0	0	0
0	3.88	0	0	0	0	0	0	0	0
0	0	2.05	0	0	0	0	0	0	0
0	0	0	1.73	0	0	0	0	0	0
0	0	0	0	1.58	0	0	0	0	0
0	0	0	0	0	1.15	0	0	0	0
0	0	0	0	0	0	1.05	0	0	0
0	0	0	0	0	0	0	0.66	0	0
0	0	0	0	0	0	0	0	0.38	0

Table 12. Matrix V (Feature X Feature matrix)

-0.5732	0.2644	-0.1547	-0.2447	-0.3912	-0.153	-0.0059	0.2726	-0.5145	-0.0462
-0.3697	0.1136	-0.2804	-0.2699	0.3627	-0.504	0.0956	-0.4121	0.348	-0.1155
-0.2981	-0.0903	-0.2996	-0.0728	-0.4254	0.5242	0.221	-0.0907	0.5108	0.1849
-0.4808	0.2294	0.0765	0.7154	0.3332	0.2537	0.0205	-0.1136	-0.0968	-0.0231
-0.0819	-0.4804	-0.4597	-0.0272	0.2896	0.207	-0.1555	0.3372	-0.058	-0.5314
-0.3505	-0.4191	0.6769	-0.3264	0.1406	0.1585	0.2636	-0.0745	-0.0593	-0.1155
-0.0869	-0.5553	-0.0584	0.3203	-0.4143	-0.2726	-0.2461	-0.4879	-0.1829	0
-0.1775	-0.2456	0.1243	0.2943	-0.0609	-0.4679	0.0928	0.6126	0.3824	0.2311
-0.2083	0.074	0.2025	-0.1844	0.0538	0.1237	-0.8821	0.047	0.2418	0.1386
-0.0516	-0.2753	-0.2692	-0.1405	0.3763	0.0826	0.025	0.0022	-0.3144	0.7625

Table 13. First column Elements of Right Singular Feature Matrix

Features before sorting	
$w^{(1)}$	-0.5732
$w^{(2)}$	-0.3697
$w^{(3)}$	-0.2981
$w^{(4)}$	-0.4808
$w^{(5)}$	-0.0819
$w^{(6)}$	-0.3505
$w^{(7)}$	-0.0869
$w^{(8)}$	-0.1775
$w^{(9)}$	-0.2083
$w^{(10)}$	-0.0516

Table 14. Sorted features

Features before sorting		**Features after sorting**	
$w^{(1)}$	0.5732	$w^{(1)}$	0.5732
$w^{(2)}$	0.3697	$w^{(4)}$	0.4808
$w^{(3)}$	0.2981	$w^{(2)}$	0.3697
$w^{(4)}$	0.4808	$w^{(6)}$	0.3505
$w^{(5)}$	0.0819	$w^{(3)}$	0.2981
$w^{(6)}$	0.3505	$w^{(9)}$	0.2083
$w^{(7)}$	0.0869	$w^{(8)}$	0.1775
$w^{(8)}$	0.1775	$w^{(7)}$	0.0869
$w^{(9)}$	0.2083	$w^{(5)}$	0.0819
$w^{(10)}$	0.0516	$w^{(10)}$	0.0516

Table 15. Sorted features and sorted Eigen values

Features after sorting		**Features after sorting**	
$w^{(1)}$	0.5732	$EV^{(1)}$	6.3313
$w^{(4)}$	0.4808	$EV^{(4)}$	3.8887
$w^{(2)}$	0.3697	$EV^{(2)}$	2.0543
$w^{(6)}$	0.3505	$EV^{(6)}$	1.7350
$w^{(3)}$	0.2981	$EV^{(3)}$	1.5891
$w^{(9)}$	0.2083	$EV^{(9)}$	1.1554
$w^{(8)}$	0.1775	$EV^{(8)}$	1.0549
$w^{(7)}$	0.0869	$EV^{(7)}$	0.6650
$w^{(5)}$	0.0819	$EV^{(5)}$	0.3829
$w^{(10)}$	0.0516	$EV^{(10)}$	0.0000

Table 16. Sorted features with Eigen values

features	$\mathbf{w^{(1)}}$	$\mathbf{w^{(4)}}$	$\mathbf{w^{(2)}}$	$\mathbf{w^{(6)}}$	$\mathbf{w^{(3)}}$	$\mathbf{w^{(9)}}$	$\mathbf{w^{(8)}}$
singular values	6.3313	3.8887	2.0543	1.7350	1.5891	1.1554	1.0549

Table 17. Reduced document feature matrix using top-7 features

	$\mathbf{w^{(1)}}$ **campus**	$\mathbf{w^{(2)}}$ **building**	$\mathbf{w^{(3)}}$ **lane**	$\mathbf{w^{(4)}}$ **floor**	$\mathbf{w^{(6)}}$ **food**	$\mathbf{w^{(8)}}$ **web**	$\mathbf{w^{(9)}}$ **wc**
$\mathbf{D^{(1)}}$	0	1	0	0	1	0	0
$\mathbf{D^{(2)}}$	0	0	0	0	2	1	0
$\mathbf{D^{(3)}}$	0	0	0	0	0	0	0
$\mathbf{D^{(4)}}$	0	0	1	0	1	1	0
$\mathbf{D^{(5)}}$	0	0	0	1	1	0	1
$\mathbf{D^{(6)}}$	2	1	1	0	1	0	1
$\mathbf{D^{(7)}}$	3	2	1	3	1	1	1
$\mathbf{D^{(8)}}$	1	0	1	1	1	0	0
$\mathbf{D^{(9)}}$	1	1	1	1	0	0	0

6. RESULTS

Figure 2 shows dimensionality of documents after initial preprocessing phase, after applying SVD and after computing feature IG then applying SVD on the resultant document matrix. The dimensionality reduction of documents is not significant through computing IG and SVD, compared to applying only SVD. This can be depicted from the Figure 2.

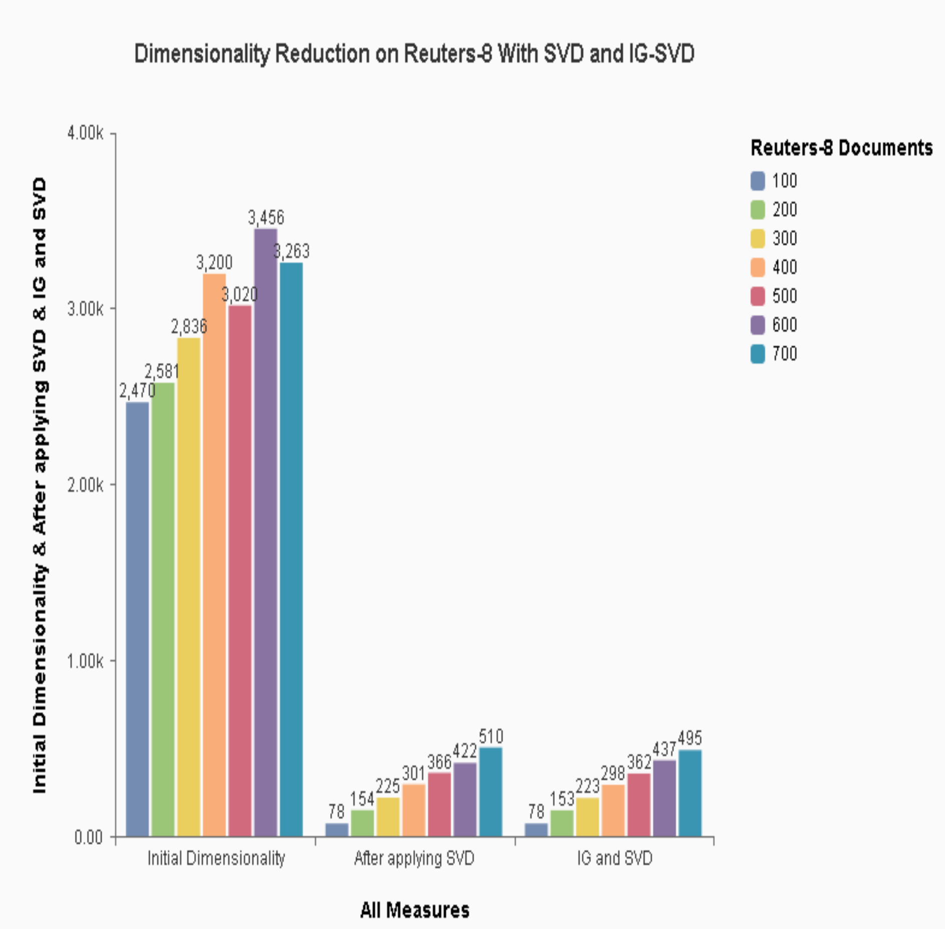

Figure 2. Dimensionality reduction obtained on Reuters-8 dataset with SVD, IG-SVD

The graph of figure 2 , compares the dimensionality reduction achieved using both the approaches for a randomly chosen 100, 200, 300, 400, 500, 600 and 700 text documents from Reuters, R8 of R21578 text corpus after retaining top-k features by retaining 90% feature energy. From this it may be concluded, that SVD alone is sufficient and no need to compute IG.

Figure 3 shows dimensions of documents before and after applying SVD. For example, the global vector of features for 700 documents chosen randomly from Reuters, R8 text corpus consists of 3263 features. So the dimensionality of each of these documents obtained after the preprocessing phase i.e. stop word and stemming elimination before applying dimensionality reduction is 3263. After applying SVD, through retaining 90% feature energy, the reduced dimensionality is 510. This means that it achieved almost 85% dimensionality reduction.

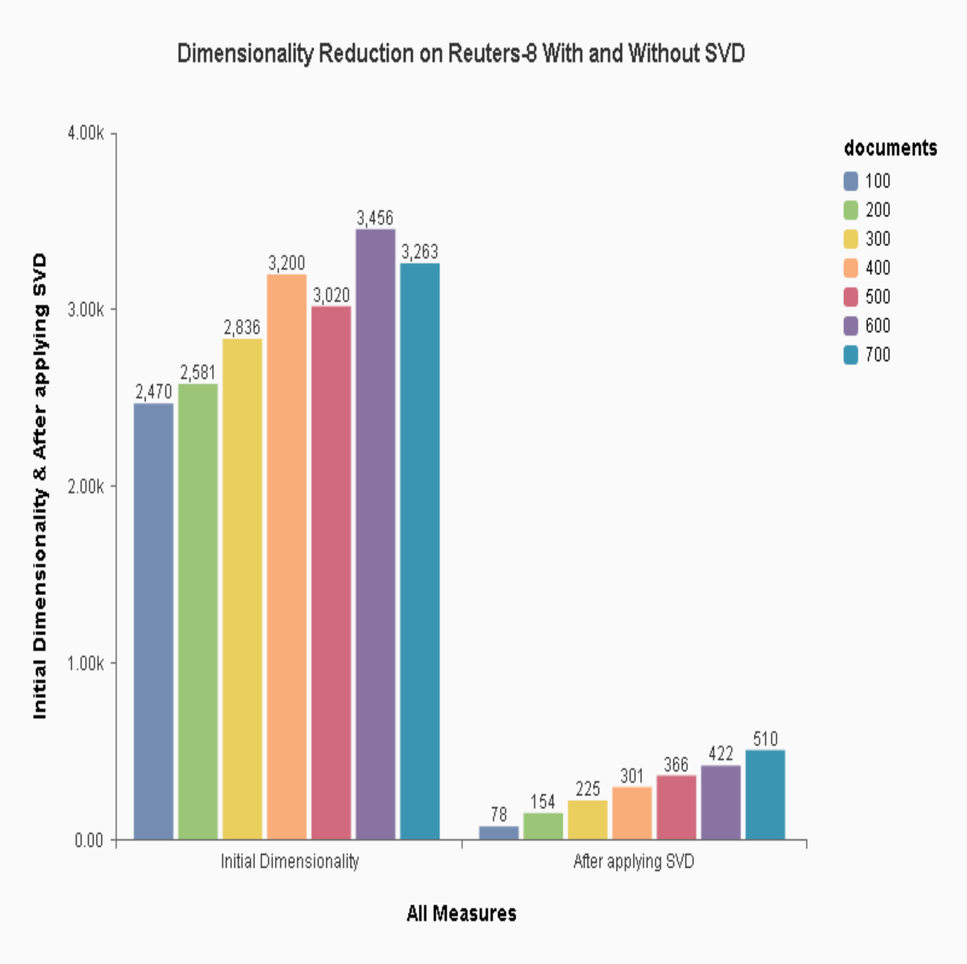

Figure 3. Dimensionality reduction obtained on Reuters-8 dataset with / without SVD

One of the evaluation approaches for clustering is through the use of a Silhouette plot. The Silhouette value ranges from -1 to +1. A high Silhouette value, usually 1denotes that it is well-matched to its own cluster and poorly matched to neighboring clusters. Conversely, if many points show a low or negative Silhouette value, then the clustering solution may have either too many or too few clusters. For the working example in section 5.1, the average silhouette value 0.4622, for k=2, after dimensionality reduction using fuzzy membership function is shown in figure 4.

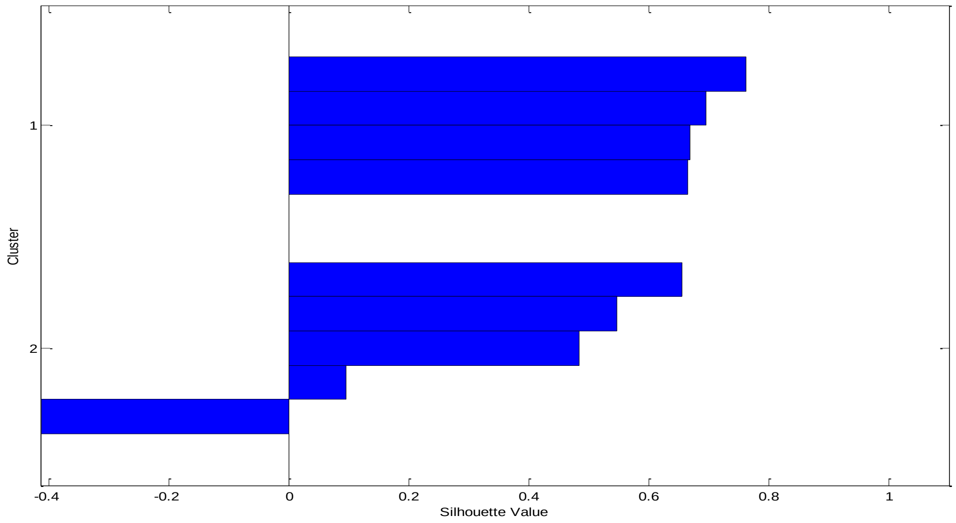

Figure 4. Average Silhouette Value, 0.4622, for k=2 after Dimensionality Reduction Using Fuzzy Membership Function

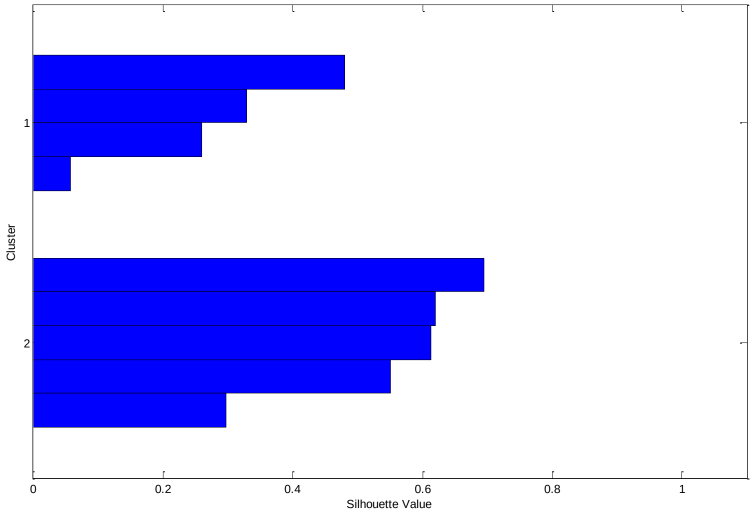

Figure 5. Average Silhouette Value, 0.4337, for k=2, Cosine, Before Dimensionality Reduction

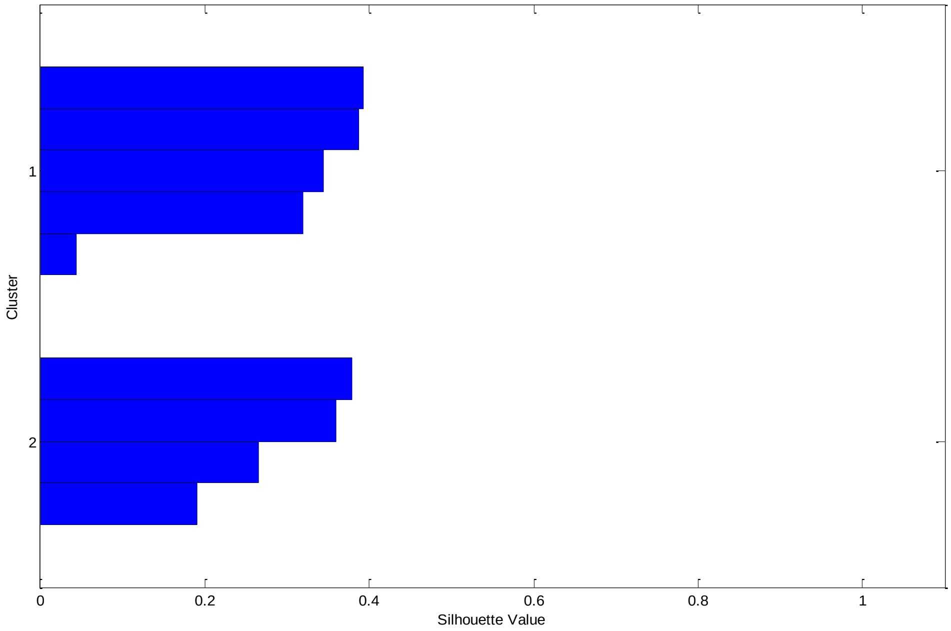

Figure 6. Average Silhouette Value, 0.2979, for k=2, City Block Distance

The average silhouette value 0.4337, for k=2, before dimensionality reduction using Cosine measure is shown in figure 5. The average silhouette value 0.2979, for k=2, for City block distance measure is shown in figure 6. This shows that the proposed approach for clustering better compared to existing K-means approach and this is obtained because of the fuzzy measure used for clustering.

7. CONCLUSIONS

This paper discusses an approach for clustering high dimensional text documents by applying K-Means algorithm using novel fuzzy Gaussian membership function which uses word probabilities with respect to documents in the text corpus. For clustering, the reduced document-feature matrix obtained using feature selection and feature extraction techniques is used. The results of dimensionality reduction using information gain and SVD are also compared. The conventional SVD approach is supported by defining a procedure to obtain top-k important features. The effectiveness of clustering using membership function is addressed using silhouette plots. The results show the performance of K-means clustering is improved when adopted the proposed measure and approach.

ACKNOWLEDGEMENTS

This work has been funded under UGC minor research project with File no: 4-4/2014-15(MRP-SEM/UGC-SERO). The principal investigator T. V. Rajinikanth and Co-Principal Investigator, G. Suresh Reddy thank UGC for providing the research project grant.

REFERENCES

Andrew Skabar, & Khaled Abdalgader. (2013). Clustering sentence-level text using a novel fuzzy relational clustering algorithm. *IEEE Transactions on Knowledge and Data Engineering, 25*(1), 62-75.

Arman Khadjeh Nassirtoussi, Saeed Aghabozorgi, Teh Ying Wah, and David Chek Ling Ngo. (2014). Text mining for market prediction: A systematic review. *Expert Systems with Applications, 41*, 7653–7670.

Berna Altınel, Murat Can Ganiz, & BanuDiri. (2015). A corpus-based semantic kernel for text classification by using meaning values of terms. *Engineering Applications of Artificial Intelligence, 43*, 54-66. http://dx.doi.org/10.1016/j.engappai.2015.03.015

Chintakindi Srinivas, Vangipuram Radhakrishna & C.V. Guru Rao. (2013). Clustering software components for program restructuring and component reuse using hybrid XOR similarity function. *AASRI Procedia, 4*, 319-328.

Chintakindi Srinivas, Vangipuram Radhakrishna, & C.V. Guru Rao. (2014). Clustering software components for program restructuring and component reuse using hybrid XNOR similarity function. *Procedia Technology,12*, 246-254.

Chintakindi Srinivas, Vangipuram Radhakrishna & C.V. Guru Rao. (2014). Clustering and classification of software component for efficient component retrieval and building component reuse libraries. *Procedia Computer Science, 31*, 1044-1050.

Chintakindi Srinivas, Vangipuram Radhakrishna & C.V. Guru Rao. (2015). Software component clustering and classification using novel similarity measure. *Procedia Technology, 19*, 866-873.

Christopher J. C. Burges. (2009). Dimension Reduction: A Guided Tour. *Foundations and trends in machine learning, 2*(4), 275–365.

Dell Zhang & Wee Sun Lee. (2006). Extracting key-substring-group features for text classification. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and data* mining (KDD '06), 474-483. doi:10.1145/1150402.1150455

Fodor, I.K. (2002) A Survey of Dimension Reduction Techniques. Technical Report, UCRL-ID-148494, Lawrence Livermore National Laboratory, Livermore. Retrieved Dec 21, 2016 from https://computation.llnl.gov/casc/sapphire/pubs/148494.pdf.

G. Suresh Reddy. (2016). Clustering and classification of high dimensional text documents using improved similarity measures (Unpublished doctoral dissertation). JNTUA University, Anantapur, INDIA.

G. SureshReddy, Rajinikanth.T.V. & Ananda Rao. (2014). A frequent term based text clustering approach using novel similarity measure. *Proceedings of the IEEE International Advance Computing Conference (IACC), 495-499.*

G. SureshReddy, Rajinikanth.T. V., & Ananda Rao. A. (2014). Design and analysis of novel similarity measure for clustering and classification of high dimensional text documents. *Proceedings of the 15th International Conference on Computer Systems and Technologies (CompSysTech '14), 883,* 194-201. doi : 10.1145/2659532.2659615

G.Suresh Reddy, A.Ananda Rao, T.V.Rajinikanth. (2015). An improved Similarity Measure for Text Clustering and Classification. *Advanced Science Letters, 21* (11), 3583-3590.

Gangin Lee & Unil Yun. (2017). A new efficient approach for mining uncertain frequent patterns using minimum data structure without false positives. *Future Generation Computer Systems, 68,* 89-110.

Gunupudi Rajesh Kumar, N. Mangathayaru, and G. Narasimha. (2015). An improved k-Means clustering algorithm for intrusion detection using Gaussian function. *Proceedings of the International Conference on Engineering & MIS 2015 (ICEMIS '15). Article 69, 7 pages.*

Gunupudi Rajesh Kumar, N. Mangathayaru, and G. Narsimha, 2015. Intrusion Detection Using Text Processing Techniques: A Recent Survey. *Proceedings of the International Conference on Engineering & MIS 2015 (ICEMIS '15),* Article 55, 6 pages.

Gunupudi Rajesh Kumar, N.Mangathayaru, G.Narsimha. (2015). A Novel Similarity Measure for Intrusion Detection using Gaussian Function. *Revista Tecnica De La Facultad de Ingeneria Universidad Del Zulia*, *39*(2), 173-183.

Gunupudi Rajesh Kumar, N.Mangathayaru, G.Narsimha. (2016). An Approach for Intrusion Detection Using Novel Gaussian Based Kernel Function. *Journal of Universal Computer Science, 22*(4), 589-604.

Gunupudi, R. K., Mangathayaru, N., & Narsimha, G. (2016). Intrusion detection a text mining based approach. *Special issue on Computing Applications and Data Mining International Journal of Computer Science and Information Security (IJCSIS), 14*, 76-88.

Hawashin, Bilal; Mansour, Ayman; Aljawarneh, Shadi. (2013). An efficient feature selection method for arabic text classification. *International Journal of Computer Applications, 83*(17), 1-6.

Hussein Hashimi, Alaaeldin Hafez, Hassan Mathkour. (2015) Selection criteria for text mining approaches, *Computers in Human Behavior, 51*, 729-733.

Jing Gao & Jun Zhang. (2005). Clustered SVD strategies in latent semantic indexing. *Information Processing & Management, 41*(5), 1051-1063.

Jung-Yi Jiang, Ren-Jia Liou, Shie-Jue Lee. (2011). A Fuzzy Self-Constructing Feature Clustering Algorithm for Text Classification. *IEEE Transactions on Knowledge and Data Engineering, 23*(3), 335-349.

Lam Hong Lee, Chin Heng Wan, Rajprasad Rajkumar, & Dino Isa. (2012). An enhanced Support Vector Machine classification framework by using Euclidean distance function for text document categorization. *Applied Intelligence, 37*(1), 80-99.

Libiao Zhang, Yuefeng Li, Chao Sun, & Wanvimol Nadee. (2013). Rough set based approach to text classification. *Proceedings of the IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT) - 3*, 245-252.

M.S.B. Phridviraj, Vangipuram RadhaKrishna, Chintakindi Srinivas, C.V. GuruRao. (2015). A novel gaussian based similarity measure for clustering customer transactions using transaction sequence vector, *Procedia Technology, 19,* 880-887.

Nafaa Haffar, Mohsen Maraoui, Shadi Aljawarneh, Mohammed Bouhorma, Abdallah Altahan Alnuaimi, Bilal Hawashin. (2017). Pedagogical indexed arabic text in cloud e-learning system. *International Journal of Cloud Applications and Computing, 7*(1), 32-46.

Ning Zhong, Yuefeng Li, & Sheng-Tang Wu. (2012). Effective Pattern Discovery for Text Mining. *IEEE Transactions on Knowledge and Data Engineering, 24*(1), 30-44.

O. Papapetrou, W. Siberski and N. Fuhr. (2012). Decentralized Probabilistic Text Clustering. *IEEE Transactions on Knowledge and Data Engineering, 24*(10), 1848-1861.

S. Aljawarneh, V. Radhakrishna, P. V. Kumar and V. Janaki. (2016). A similarity measure for temporal pattern discovery in time series data generated by IoT. *Proceedings of International Conference on Engineering & MIS (ICEMIS),* 1-4.doi: 10.1109/ICEMIS.2016.7745355.

Sajid Mahmood, Muhammad Shahbaz, & Aziz Guergachi. (2014). Negative and positive association rules mining from text using frequent and infrequent itemsets. *The Scientific World Journal*, *2014*, Article ID 973750, 11 pages. doi:10.1155/2014/973750

Shadi A Aljawarneh, Mohammed R Elkobaisi, Abdelsalam M Maatuk. (2016). A new agent approach for recognizing research trends in wearable systems. *Computers & Electrical Engineering*. Retrieved from http://www.sciencedirect.com/science/article/pii/S0045790616309995.

Shadi A. Aljawarneh, Vangipuram Radhakrishna, P.V.Kumar, V. Janaki. (2017). G-SPAMINE: An approach to discover temporal association patterns and trends in internet of things. *Future Generation Computer Systems*, http://dx.doi.org/10.1016/j.future.2017.01.013.

Shadi A. Aljawarneh, Raja A. Moftah, Abdelsalam M. Maatuk. (2016). Investigations of automatic methods for detecting the polymorphic worms signatures. *Future Generation Computer Systems*, *60*, 67-77.

Shadi Aljawarneh. (2011). Cloud Security Engineering: Avoiding security threats the right way. *International Journal of Cloud Applications and Computing (IJCAC), 1*(2), 64-70.

Shady Shehata, Fakhri Karray; &Mohamed Kamel. (2010). An efficient concept-based mining model for enhancing text clustering. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1360-1371. doi : 10.1109/TKDE.2009.174

Shie-Jue Lee, Jung-Yi Jiang. (2014). Multilabel Text Categorization Based on Fuzzy Relevance Clustering. *IEEE Transactions on Fuzzy Systems, 22* (6), 1457-1471.

Shuigeng Zhou & Jihong Guan. (2002). An approach to improve text classification efficiency. *Proceedings of the 6th East European Conference on Advances in Databases and Information Systems (ADBIS '02),* 65-79.

Shusaku Tsumoto, Haruko Iwata, Shoji Hirano, Yuko Tsumoto. (2014). Similarity-based behavior and process mining of medical practices. *Future Generation Computer Systems, 33*, 21-31. http://dx.doi.org/10.1016/j.future.2013.10.014

Sunghae Jun, Sang-Sung Park, & Dong-Sik Jang. (2014). Document clustering method using dimension reduction and support vector clustering to overcome sparseness. *Expert Systems with Applications*, *41*(7), 3204-3212. doi: 10.1016/j.eswa.2013.11.018

V. Radhakrishna, P. V. Kumar & V. Janaki. (2016). Mining of outlier temporal patterns. *Proceedings of the International Conference on Engineering & MIS*, 1-6. doi: 10.1109/ICEMIS.2016.7745343

V. Radhakrishna, P. V. Kumar, V. Janaki & S. Aljawarneh. (2016). A computationally efficient approach for temporal pattern mining in IoT. *Proceedings of the International Conference on Engineering & MIS*, 1-4. doi: 10.1109/ICEMIS.2016.7745354.

V. Radhakrishna, P. V. Kumar, V. Janaki & S. Aljawarneh. (2016). A similarity measure for outlier detection in time stamped temporal databases. *Proceedings of the International Conference on Engineering & MIS*, 1-5.doi: 10.1109/ICEMIS.2016.7745347

Vangipuram Radhakrishna, P.V.Kumar, &V.Janaki. (2015). A novel approach for mining similarity profiled temporal association patterns. *Revista Tecnica De La Facultad de Ingeneria Universidad Del Zulia*, *38*(3), 80-93.

Vangipuram Radhakrishna, P.V.Kumar, &V.Janaki. (2015). A survey on temporal databases and data mining. *Proceedings of the International Conference on Engineering & MIS*. doi:10.1145/2832987.2833064

Vangipuram Radhakrishna, P.V.Kumar, &V.Janaki. (2015). An approach for mining similarity profiled temporal association patterns using Gaussian based dissimilarity measure. *Proceedings of the International Conference on Engineering & MIS*. doi:10.1145/2832987.2833069

Vangipuram Radhakrishna, P.V.Kumar, &V.Janaki. (2016). A novel similar temporal system call pattern mining for efficient intrusion detection. *Journal of Universal Computer Science, 22*(4), 475-493.

Vangipuram Radhakrishna, P.V.Kumar, &V.Janaki. (2016). An efficient approach to find similar temporal association patterns performing only single database scan. *Revista Tecnica De La Facultad de Ingeneria Universidad Del Zulia, 39*(1), 241-255. doi:10.21311/001.39.1.25

Vangipuram Radhakrishna, P.V.Kumar, &V.Janaki. (2016). Looking into the possibility of novel dissimilarity measure to discover similarity profiled temporal association patterns in IoT. *Proceedings of the International Conference on Engineering & MIS*, 1-5.

Vangipuram Radhakrishna, P.V.Kumar, &V.Janaki. (2016). Mining Outlier Temporal Association Patterns. *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies (ICTCS '16)*. doi: 10.1145/2905055.2905320

Vangipuram Radhakrishna, P.V.Kumar, &V.Janaki. (2016). Normal Distribution Based Similarity Profiled Temporal Association Pattern Mining (N-SPAMINE). *Database Systems Journal*, *7*(3), 22-33. http://www.dbjournal.ro/archive/25/25_3.pdf

Vangipuram Radhakrishna, Shadi A. Aljawarneh, Puligadda Veereswara Kumar, Kim-Kwang Raymond Choo. (2016). A novel fuzzy gaussian-based dissimilarity measure for discovering similarity temporal association patterns. *Soft Computing*. First Online: 18 November 2016. doi:10.1007/s00500-016-2445-y

Wen Zhang, Taketoshi Yoshida, & Xijin Tang. (2008). Text classification based on multi-word with support vector machine. *Knowledge-Based Systems, 21*(8), 879-886.

Wen Zhang, Taketoshi Yoshida, Xijin Tang, & Qing Wang. (2010). Text clustering using frequent item sets. *Knowledge-Based System, 23*(5), 379–388. http://dx.doi.org/10.1016/j.knosys.2010.01.011

Yanjun Li, Congnan Luo, Chung, & S.M. (2008). Text clustering with feature selection by using statistical data. *IEEE Transactions on Knowledge and Data Engineering*, *20*(5), 641-652.

Yung-Shen Lin, Jung-Yi Jiang & Shie-Jue Lee. (2014). A similarity measure for text classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, *26(*7), 1575-1590.

N. Haffar, M. Maraoui & S. Aljawarneh. (2016). Use of indexed Arabic text in e-learning system. Paper presented at the International Conference on Engineering & MIS (ICEMIS), Agadir. Abstract retrieved from http://ieeexplore.ieee.org/abstract/document/7745321/

Ons Meddeb, Mohsen Maraoui, & Shadi Aljawarneh. (2016, Sep). Hybrid modeling of an OffLine Arabic Handwriting Recognition System AHRS. Paper presented at the International Conference on Engineering & MIS, Agadir. Retrieved from http://ieeexplore.ieee.org/abstract/document/7745319/

Mohammed R Elkobaisi, Abdelsalam M Maatuk, Shadi Aljawarneh. (2015, Aug). A Proposed Method to Recognize the Research Trends using Web-based Search Engines. Paper presented at the International Conference on Engineering & MIS, Istanbul, Turkey. Retrieved from http://dl.acm.org/citation.cfm?id=2833012

Shadi Aljawarneh & Bassam Al-shargabi. (2013). Gene Classification: A Review. Paper presented at the 6th International Conference on Information Technology. Retrieved from http://sce.zuj.edu.jo/ICIT13/images/Camera%20Ready/Sorftware%20Engineering/650_shadi.pdf