International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-3, September 2019

AEAO: Auto Encoder with Adam Optimizer Method for Efficient Document Indexing of Big Data

Y.Krishna Bhargavi, Y.S.S.R.Murthy, O.SRINIVASA RAO

Abstract: ***In the big data era, the document classification became an active research area due to the explosive nature in the volumes of data. Document Indexing is one of the important tasks under text classification. The objective of this research is to increase the performance of the document indexing by proposing Adam optimizer in the auto-encoder. Due to the larger dimension and multi-class classification problem, the accuracy of document indexing is reduced. In this paper, an enhanced auto encoder is used based on the objective function of the Adam optimization (AEAO), which improves the learning rate and accuracy of indexing. The documents from the 20-newsgroup data set are converted into vector representation, and then the cosine similarity and Pearson correlation have been measured from the vector. The word to vector representation has words in the vector form and the frequency of words in the document increases their value. The Adam optimization technique selects the features by using similarity values and improves the learning rate. The auto encoder classifier classifies the document based on the objective function of the Adam optimizer. The experiment is conducted using python and the result infers that the classification performance of AEAO is better than that of Similarity-based classification framework for Multiple-Instance Learning and Self-Adaptive LSH encoding for multi-instance Learning techniques in terms of parameters like precision, recall and f-score.***

Keywords: ***Adam Optimizer, Auto Encoder, Big Data, Cosine Similarity, Document Indexing, Pearson Correlation, Word to Vector Representation.***

I. INTRODUCTION

In Big data, text classification is one of the major factors that reduce the burden of large data maintenance in many sectors such as IT, education institutions, medical field etc. The classification technique can be used to classify the document with content-based effectively. The representation of text documents are to be easily interpreted by the classifiers.

Manuscript published on 30 September 2019
* Correspondence Author
Y.Krishna Bhargavi*, Department of CSE, Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad, India. Email: kittu.bhargavi@gmail.com
Dr.Y.S.S.R.Murthy, Department of IT, SRKR Engineering College, Bhimavaram, India. Email: yssrmurthy@gmail.com
Dr.O.Srinivasa Rao, Department of CSE, University College of Engineering, JNTUK, Kakinada, India. Email: osr_phd@yahoo.com

Bag-of-words method and graph based representation are the usual ways of representing the documents. The technique of bag-of-words is one of the simplest ways of representing document based on the words and its occurrence in the document [1]. The category of data has to be predefined in order to classify the relevant document [2]. There is a need to rank the document based on the popularity because there are a number of documents available online for the single query that makes it difficult for the user to retrieve relevant information [3]. The web provides enormous information easily than searching for relevant information in local libraries [4]. The most common technique used in automated document classification is long short-term memory technique that analyses the long dependencies sequentially using a memory unit and a gate mechanism [5]. Automated document indexing is highly in need to reduce time and high recognition of required information. The document indexing method as well as traditional method requires pre treating the data to improve the performance of classification. In the classification, it is difficult to identify the relevant features due to its high dimensionality paving way to feature selection [6].

II. LITERATURE REVIEW

The most common problem occurs with document indexing involves in feature selection and high dimensionality of data. The general techniques involve in representing the features are Bag-of-words and graph based technique. The graph-based techniques clear the issues of high-dimension and have low efficiency. Centroid based classifier has better performance than KNN, Naive Bayes and decision tree, but decreases the accuracy as data is present as a class-imbalance distribution. Naive Bayes [7–11] technique classifies the document based on the keywords and joint probability of document category. Some of the methods applied to the document indexing include KNN, decision tree, Naive Bayes, genetic algorithm and Support Vector Machine (SVM). KNN technique stores the similarity between the documents and the certain amount of classification data, from that the documents have been classified [12–16]. Naive Bayes has a better outcome in the document indexing but if the training documents are low, the performance degrades. KNN has the lower computational time and it has some issues like high storage requirement, and low efficiency. SVM is a very slow process and decision tree techniques don't perform well with high features [17–21].

The learning method of Back Propagation Neural Network in document indexing struck in the local minima and has the disadvantage of slow learning [22–26].

Author	Methodology	Advantages	Limitations
Zongda Wu, et al. [27]	Heuristic selection and semantic measure	Semantic similarity helps to achieve higher accuracy	It increases the execution time of the method as this doesn't follow the feature reduction.
Younghoon Lee, et al. [28]	The documents are converted into matrix and features are reduced by the filtering technique then semantic distances are preserved.	Filtering technique is used in this method reduces the execution time of the word clustering.	This technique needs to select more relevant features for the efficient performance. It has poor efficiency.
Ghulam Mujtaba, et al. [29]	Conceptual graph based document representation is used instead of Bag-of-words method	The autopsy documents are effectively classified in this method with its similarity measure.	Six similarity and graph based techniques used for the document indexing, increases the computational time.
Shoaib Jameel, et al. [30]	The hybrid techniques of Bag-of-words, Word Net based approach and SVM is used as a classifier.	This technique has high efficiency compared to state-of-art method.	The computational time need to be reduced.
Chuan Liu, et al. [31]	This method follows class label information and word order structure.	This technique solves the problem of misfit in the data that usually occurs in the centroid based classifier.	The high precision values are not attainable for every category.
Bo Liu, et al. [32]	Multiple instances learning technique based on similarities is used.	The feature selection helps to minimize the execution time by learning the similarity.	The multiple instances produce many features and tend to classify more documents and it suffers from increase in error rate.
Wei Zong, et al. [33]	The semantic features have been measured in the documents and SVM is used to classify the document.	For the different number of features, F-measure is high.	Higher selection of features increases the execution time. Data distribution is not considered in this method and causes the error in classification.
Haytham Elghazel, et al. [34]	The document vocabulary is categorized and Latent semantic Indexing is measured, Boos- Texter is used to classify the document.	High dimensions problem is solved using latent semantic method.	For efficient indexing of documents, the learning rate is to be improved.
Abdullah Saeed Ghareb, et al. [35]	An enhanced genetic algorithm with various hybrid methods solves the problem of high dimensionality.	This method provides better classification results.	This technique attempted to solve the problem of wrapper technique (higher execution time), still lagging in efficiency.
Kijung Park, et al. [36]	Latent topics are used to classify the documents.	The technique is delivered the reliability.	It finds difficulties in identifying the relation between the documents and the accuracy of the method need to be improved.

Table 1: Literature Review

III. PROBLEM DEFNITION & SOLUTIONS

The major problems with the document indexing are discussed in the following:

- Most of the state-of-art techniques involve in the single-label multi-class classification problem, but the text documents naturally belong to the multiple categories. So efficient multi-class technique is required in the classification.
- The common issue in the document indexing is a higher dimension in the documents and many techniques attempted to solve these problems by filtering method. The filtering techniques reduce the features in the document at the cost of efficiency. Wrapper technique has the highest accuracy and this requires more execution time for the large number of documents.
- KNN techniques and other clustering techniques were also proposed for the document indexing. These techniques face some problem of complexity in creating the sample of the document and it is sensitive to the single training sample, i.e., single noisy sample can affect the performance of the clustering method.

By combining Adam Optimizer and auto encoder technique, the problems identified above are solved. This method tackles this issue by using the extended stochastic approach in the learning process of the classifier. The feature selection techniques clear the problem with higher dimension, which is a basic issue of the document data. This will select the relevant features by learning the similarities between the documents.

This method increases the performance by classifying the data in the multi-class. Generally, the feature selection techniques deteriorate the performance of the classification, but in this research, this is solved by selecting the relevant features. Adam optimizer is the extended version of the stochastic method, which increases the learning rate in selecting relevant features.

The objective function produced by the Adam optimizer is utilized by the auto encoder to classify the document. This solves the problem with high computational time.

Usually the documents are represented in two ways, namely Bag-of-words and graph- based representation. The Bag-of-words convert the documents into vector representation and the whole document is provided as vector to the learning method. The graph-based method plotted the document as the graph and regarding values are used in the learning method. The graph based method has less efficiency. The Bag-of-words are used in the current technique to increase the performance. Filter and wrapper techniques have the issues of high computation time and low efficiency, so here embedded technique is used. The Adam optimization technique is used instead of the traditional auto encoder learning method, which is embedded in the network. This increases the learning rate and efficiency of the document indexing. Another problem with the document indexing involves in the misfit of the classification due to the complex data distribution. The AdaGrad and RMSProp used in the Adam optimizer solves the data distribution issues by analyzing the data through finding the objective function. Adam optimization and auto encoder techniques solve the problem of high dimension, large computational time and misfit problem.

IV. PROPOSED METHODOLOGY

The combination of the Adam optimizer and auto encoder is used in this research to classify the documents from the 20-Newsgroup data set. The two similarities are measured in the input document from the dataset. In this method, the extended stochastic method finds the features in the given document while improving the learning rate. The auto encoder classifier classifies the document based on the objective function of the Adam optimizer. The word to vector technique represents the document in the vector value, and then the similarities are measured between the documents. To find relevant features from the document, Adam optimizer uses the similarities. The auto encoder classifies the test data, after the feature is selected. Figure 1 shows the proposed technique's architecture.

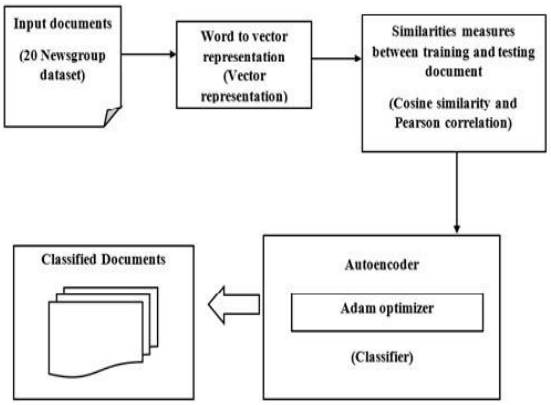


Figure 1: Architecture of Proposed Method

A. Word to Vector Representation

Word to Vector representation is the word vector model [37–41], which defines the words in the word space vector representation. This is the simple representation as those used in the text categorization. As the similar words in the different documents are represented in the vector and these words are identified using the machine learning technique. Assume V is a document vector representation; the values are like as given in the Equation (1).

$$V = (v_1, v_2, v_3, ..., v_4) \quad (1)$$

Each word in the document considers as an article and vector of the words collectively represent the document. The vector is represented by word space vector and it is measured for each article. The number of words in total for the document is given as n and as shown in Equation (2), the representation of document contains v_1 *to* v_n. The vector representation for each single word is given below:

$$v_1 = (1, 0, 0, \ldots 0)$$
$$v_2 = (0, 1, 0, \ldots 0)$$
$$v_3 = (0, 0, 1, \ldots 0)$$
$$v_n = (0, 0, 0, \ldots n) \quad (2)$$

The visual and intuitive characteristic words are represented in the vector. The document vector representation is the sum of individual word vectors in the given word. The vector representation is similar to the special word space i.e., most similar words resemble each other, and otherwise they appear irrelevant. Vector representation is generally said to be correlation value and this finds the similarity between the training and testing correlation. For instance, if the two words like "movie" and "Hollywood" appear more in the document, then the correlation value gets increased as the word combinations repeat. So, these two words have higher correlation in their documents. The similarity between the values are higher that helps to retrieve similar documents.

This method mostly focused on the repetition of the words after the word once appeared in the document, which considers as contextual information of the word. The window length is denoted as c and words in the same window belong to the same context.

Different words have different contextual properties and used for representing words, which is basic idea followed by the word to vector representation.

The two training models are followed by the word to vector representation, one is Bag of words and skip-gram is another. In this method, the bag-of-words gives the occurrence of words in the document that is used for document indexing by the auto encoder.

B. Similarity Measures

The cosine and Pearson coefficients are measured to find the distance value of the document that is used further to identify the similar document. The similarity value obtained is used by the Adam optimizer to find the relevant features between the documents, helps to improve efficiency of the technique. Cosine and Pearson coefficient similarities help to find the similarity values, which are common similarities used for the text classification.

Cosine Similarities

Once vector values are measured for the documents, then the correlation between the documents are measured as the cosine angle between the vectors, so-called cosine similarity [42], [43]. The cosine similarities are mostly used to obtain the similarity between the documents and also for the clustering process for information retrieval.

The two documents are considered $\overrightarrow{V_1}$ and $\overrightarrow{V_2}$, then cosine similarity is measured using the Equation(3).

$$C_{sim}\left(\overrightarrow{V_1},\overrightarrow{V_2}\right)=\frac{\left(\left(\overrightarrow{V_1}\right).\left(\overrightarrow{V_2}\right)\right)}{|\overrightarrow{V_1}|\times|\overrightarrow{V_2}|} \quad (3)$$

Where $\overrightarrow{V_1}$ and $\overrightarrow{V_2}$ are the vector representations of the documents with n dimension is given in the Eq.(1). The dimension represents the term weight in the document and it is present as whole number. Cosine similarity values occur as non-negative and present between 0 and 1. Cosine similarity values are independent of the length of the document and help to find the similarity between the two given documents. If the cosine similarity value is obtained as 1, then the documents are identical. If the value is obtained as 0, then the two documents are irrelevant and not to be classified for the given document. The documents with the same composition but different total are considered as identical.

Pearson Correlation Coefficient

The Pearson correlation is another measure that finds the relation between the two vectors [44], [45]. The dimension of the vector is denoted as n and Pearson correlation is measured using the Eq.(4).

$$P_{sim}\left(\overrightarrow{V_1},\overrightarrow{V_2}\right)=\frac{n\sum_{d=0}^{n}w_{d,1}\times w_{d,1}-TF_1\times TF_2}{\sqrt{\left[n\sum_{td=1}^{n}w_{d,1}^2-TF_1^2\right]\left[n\sum_{d=1}^{m}w_{d,2}^2-TF_2^2\right]}} \quad (4)$$

Where $TF_1=\sum_{d=1}^{m}w_{d,1}$ and $TF_2=\sum_{d=1}^{m}w_{d,2}$

This similarity measure ranges from -1 to +1 and this Pearson correlation gives the distance measures from the documents. The combination of these measures is used by the Adam optimizer to find features.

C. Auto Encoder Technique

The Bag-of-words represent as x, which is mentioned in the word to vector representation section and x_i is each word in the document.The order of vector in bag-of-word is not similar to the word order in the document and bag-of-words are measured for the D-dimensional vector representation from the training set$\{x^t\}_{t=1}^{T}$. The W matrix is constructed from $D\times V$, where summing of column of W is used by the encoder. The encoder function is represented as $\phi(x)$ and loss function is optimized in the decoder. The parameters are carefully chosen for optimizing the loss function otherwise it will affect the performance in spare and high-dimensional data.

Auto encoder Training of Bag-of-words

Technique followed in this method is to compute the architecture of the auto encoder with direct representation of the vector. Auto encoder computes a binary vector observation with decoder reconstruction that reduces the cross-entropy loss [46]. The bag-of-words are converted into the fixed size and word present in the document is $\mathrm{v(X)}_{x_i}=1$ otherwise zero [47], [48]. The encoder is represented by multiplying $\mathrm{v(x)}$ with word representation matrix W, given in Eq. (5).

$$\mathrm{a(X)=c+Wv(x),\phi(X)=h(a(x)} \quad (5)$$

Where $\mathrm{h(.)}$ is an element-wise non-linearity i.e. sigmoid or hyperbolic tangent, and c is a D-dimensional bias vector. The next step is involved in the sum of representing all words with its frequency of each word. The decoder has been parameterized to produce a reconstruction from the non-linear form as shown in the Eq. (6).

$$\mathrm{\hat{v}(X)=sigm(V\phi(X)+b)} \quad (6)$$

Where transpose of matrix W is V, b is the bias vector of the reconstruction layer and $\mathrm{sigm(a)=1/(1+\exp(-a))}$ is the sigmoid non-linearity in the Eq. (7). Then the reconstruction is compared to the original bag-of-words.

$$\mathrm{l(v(X))=-\sum_{i=1}^{V}v(X)_i\log(\hat{v}(X)_i)+(1-v(X)_i)\log(1-\hat{v}(X)_i)} \quad (7)$$

Adam optimizer is used instead of sum of reconstruction cross-entropies across the training set. Usually, this follows the stochastic or mini-batch gradient descent function, whereas the extended version of the stochastic function is Adam optimizer. The high-dimensional data makes the reconstruction process to consume more time and attempt to reconstruct the whole bag-of-words. Adam optimization increases the learning rate and solves this issue, which has computational time less than that of stochastic reconstruction. The input is given as mini-batch to perform the function and updates the value. Each update improves the function of the Adam optimizer.

Adam Optimization technique

The Adam optimization technique is used in this research to update network weight iteratively depend on the training data. The objective function is $f(\theta)$, a stochastic scalar function that is differentiable w.r.t θ. This feature is reduced w.r.t θ and focus to obtain the minimum features $E[f(\theta)]$ with most relevant. The stochastic function realizations are denoted as $f_1(\theta),\ldots,f_t(\theta)$ with the timesteps of $1,\ldots t$. The stochastic values may be selected from the random subsamples of data points, or evolve from inherent function noise. The stochastic function gradient represents as $g_t=\nabla_\theta f_t(\theta)$, the partial derivatives vector of f_t w.r.t θ measured at time step t. The moving averages of gradient (m_t) and the squared gradient (v_t) where hyper-parameter is defined as $\beta_1,\beta_2\in[0,1)$ controls the exponential decay rates of these moving averages.

The moving averages are assigned as the gradients first moment (mean) and second (the non-centered variance) raw moment. During the initial time steps the decay rates are low (β are near to 1), then the moving averages are set as 0's that results in the moment estimates which are initialized as zero. So, decay rate is set as 0.9 ($\beta = 0.9$), which help to improve the learning rate. The moving averages are easily counteracted and it helps to achieve bias-corrected value $\widehat{m_t}$ and $\widehat{v_t}$ and shown in the initializing bias condition.

Table 2: Pseudo Code for Adam Optimizer

Set : α : step size
Set : $\beta_1, \beta_2 \in [0,1)$:Moment estimates- Exponential decay rates
Set :$f(\theta)$: Stochastic objective function with θ as parameter
Set : θ_0 : Parameter vector initialization
$m_0 \leftarrow 0$ (First moment vector initialization)
$v_0 \leftarrow 0$ (Second moment vector initialization)
$t \leftarrow 0$ (Initialization of time step)
While (θ_t not converged) do
$t \leftarrow t + 1$
$g_t = \nabla_\theta f_t(\theta_{t-1})$ Gradient
$m_t \leftarrow \beta_1 . m_{t-1} + (1 - \beta_1) . g_t$
$v_t \leftarrow \beta_2 . v_{t-1} + (1 - \beta_2) . g_t^2$
$\widehat{m_t} \leftarrow m_t / (1 - \beta_1^t)$
$\widehat{v_t} \leftarrow v_t / (1 - \beta_2^t)$
$\theta_t \leftarrow \theta_{t-1} - \theta . \widehat{m_t} / (\sqrt{\widehat{v_t}} + \epsilon)$
end while
return θ_t

The change in computation order can improve the efficiency of the algorithm. For example, the last three lines in the loop can be replaced using following Eq. (8).

$\alpha_t = \alpha . \sqrt{1 - \beta_2^t} / (1 - \beta_1^t)$ and $\theta_t \leftarrow \theta_{t-1} - \alpha . m_t / (\sqrt{v_t} + \hat{\epsilon})$(8)

Update Rule

Carefully selecting the choice of step sizes is the important property of the Adam update rule. Consider $\epsilon = 0$, effective step followed in parameter space at timestep $t$is $\Delta_t = \alpha . \widehat{m_t} / \sqrt{\widehat{v_t}}$. The two bounds of stepsize is $|\Delta_t| \leq \alpha . \frac{(1-\beta_1)}{\sqrt{1-\beta_2}}$ in the case $(1 - \beta_1) > \sqrt{1 - \beta_2}$, and $|\Delta_t| \leq \alpha$ otherwise. The first condition follows high sparsity value, when gradient is zero at all time steps except current time step. The step size should be smaller for less sparse cases and if$(1 - \beta_1 = \sqrt{1 - \beta_2}$, and then it gets as $|\widehat{m_t} / \sqrt{\widehat{v_t}}| < 1$therefore$|\Delta_t| < \alpha$. The step size approximately bounded by step size setting to α, i.e., $|\Delta_t| \preccurlyeq \alpha$. Around the parameter α trusted region has been created, which helps to measure the gradient and beyond the region gradient values are not calculated. The good optimum value for many machine learning algorithms is prior known. The α is measured in right scale in advance andα is a set in the parameter space, then the optimum value of θ has been found with number of iteration.

Initialization Bias Correlation

First moment vector in the algorithm is analogous and Adam optimizer helps to derive the second moment vector. The second vector is measured with the help of the exponential moving average of the squared gradient with its decay rate β_2. The second moment vector is non-centered variance and consider g gradient of the stochastic objectives$f$. The gradient value of the each time step is measured as $g_1, g_2, \ldots, g_t$ from the gradient distribution function$g_t \approx p(g_t)$. Let the gradient values are initialized as $v_o = 0$ and exponential moving average with timestep t is calculated as $v_t = \beta_2 . v_{t-1} + (1 - \beta_2) . g_t^2$ can be written as a function of gradient at all previous time step in Eq. (9):

$$v_t = (1 - \beta_2) \sum_{i=1}^{t} \beta_2^{t-i} . g_i^2 \quad (9)$$

The moving average of exponential value at time step of t is $\mathbb{E}[v_t]$, which highly depend on the gradient of the second moment value $\mathbb{E}[g_t^2]$. The exponential moving average is measured in Eq. (10).

$$\begin{aligned} \mathbb{E}[v_t] &= \mathbb{E}[(1 - \beta_2) \textstyle\sum_{i=1}^{t} \beta_2^{t-i} . g_i^2] \\ &= \mathbb{E}[g_t^2 . (1 - \beta_2) \textstyle\sum_{i=1}^{t} \beta_2^{t-i} + \zeta] \\ &= \mathbb{E}[g_t^2] . (1 - \beta_2^t) + \zeta \end{aligned} \quad (10)$$

The second moment gradient $\mathbb{E}[g_t^2]$does not vary if the ζ is zero. The value of ζ has to be small and decay value is chosen for this condition, such that exponential moving average assigns small weights to gradient. The average gradient values are calculated from the many gradient values to estimate reliable value of second moment with small decay rate. The prior step in initialization is insufficient due to lower decay rate and lead to the initial step much larger.

Auto encoder Function

Consider the bag-of-words is X for the training document x, and that is compared to the bag-of-words Y in the testing document y and it helps to classify the related document from the datasets. Let training set of such pairs(x, y), representation has been learned and compared to the other document to find the relevant features. If the vector representation of the training and testing documents is more similar, then the document is classified in the category. The regular auto encoder technique is used along with Adam optimizer, which reconstructs the vector representation and compared with other document [49]. Now, the matrix is defined as W^x and W^y that they are related to the vector representation of both document $V^x$and V^y[50-52]. The vector representation of the document may vary in size, but the dimension is same for the document. Then the Adam optimizer helps to reconstruct the representation.

$\theta_t(X) \leftarrow \theta_{t-1}(X) - \alpha . \widehat{m_t}(X) / (\sqrt{\widehat{v_t}} + \epsilon \theta_t(Y)\ \theta_{t-1}(Y) - \alpha . \widehat{m_t}(Y) / (\sqrt{\widehat{v_t}} + \epsilon$

(11)

The objective function has been found for both the documents and the relevance has been compared for both documents, in the Eq. (11). This technique decodes the reconstruction of one document x and another decodes the other document y. The encoding and decoding techniques help to mapping the document from the given pair (x, y), and the model is trained to construct y from x (loss $l(x, y)$), then construct x from y (loss $l(y, x)$), and reconstruct x and y from itself.

Reconstruction Process

To find the relevant document related to the input document, two more terms are included in the loss function for more efficient representation. The joint reconstruction technique has been followed in this method $l([x, y], [x, y])$, where both documents are simultaneously given as input and reconstructed in the Eq. (12).

$$l(x,y) + l(y,x) + l(x) + l(y) + \beta l([x,y],[x,y]) - \lambda.\mathrm{cor}(a(x),a(y)) \quad (12)$$

The term $\mathrm{cor}(a(x),a(y))$ is the sum of the scalar correlation between each pair $a(x)_D, a(y)_D$, from the dimension of vectors $a(x), a(y)$. Adam optimization techniques are used to obtain the objective function of the system.

D. Document Representations

The construction of matrix W^x and W^y from the given two documents and vector representation has been provided in column of the matrix.

The document has been retrieved as $\mathbb{Z} \in \{X,Y\}$ and contain n words, $z_1, z_2, z_3, \ldots, z_m$. Some parameters are evaluated to analyze the performance of the proposed technique after the document has been classified.

V. EXPERIMENTAL DESIGN

The most important objective of the research is to present the effective document indexing technique. The input documents are represented in the bag-of-words and the similarity methods namely cosine similarity and Pearson correlation are applied. The Adam optimizer technique learns the features from the document. Auto encoder classifier uses the features to classify the document in its respective category. The data collection and the metric measures are discussed in this section.

Benchmark Collection

20-News group data sets [53–58] (http://qwone.com/~jason/20Newsgroups/) is used in this research for evaluation of the proposed method. This data set consists of approximately 20,000 documents and it is one of the common data set used for the document indexing technique. This data set consists of 20 different newsgroups and openly available data set for download in the format of ".tar.gz".

Parameter metrics

The performance of the document indexing is evaluated using parameters like f-score, precision and recall. The execution time and F-score are calculated, then compared with another method. The number of instances that are classified correctly is denoted as a and the number of instances that are classified incorrectly is denoted as b. Total class is C_m and the number of instances in C_m is classified into another class as c. The formula for precision p, recall r and f-score f_1 are given in this section. The precision and recall have the inverse relation, the cost of one increase and other decreases. The f-score is introduced to measure the average value based on the precision and recall value, is shown in Eq. (13), (14)& (15). The time taken by the method to classify the data is the execution time. The Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) are measured in this method from the Eq. (16) & (17).

$$p = \frac{a}{a+b} \quad (13)$$

$$r = \frac{a}{a+c} \quad (14)$$

$$F_1(r,p) = \frac{2pr}{p+r} \quad (15)$$

$$RMSE = [n^{-1}\textstyle\sum_{i=1}^{n}|e_i|^2]^{\frac{1}{2}} \quad (16)$$

$$MAE = [n^{-1}\textstyle\sum_{i=1}^{n}|e_i|] \quad (17)$$

VI. RESULTS & DISCUSSION

This research used Python with the Sci-kit Learn package to work on the standard 20-Newsgroup data sets that consist of documents related to different categories. The text in the documents is converted into vector representation and repeated words in the document increases their specific representation. Then the cosine similarity and Pearson's correlation values are measured that finds the similarity between the documents. Adam optimizer increases the learning rate due to its extended techniques of stochastic method. The auto encoder effectively classifies the document based on the Adam optimizer's objective function. There are numerous parameters that has been measured to analyze the performance of the proposed method. The parameters are compared with existing methods to investigate the efficiency of document indexing. The value of major parameters like F-score, precision and recall are measured in the experiment.

Quantitative Analysis

The proposed technique is analyzed in the quantitative manner to measure the effectiveness in document classification. The basic measures of the document retrieval include precision, recall and F-score are calculated.

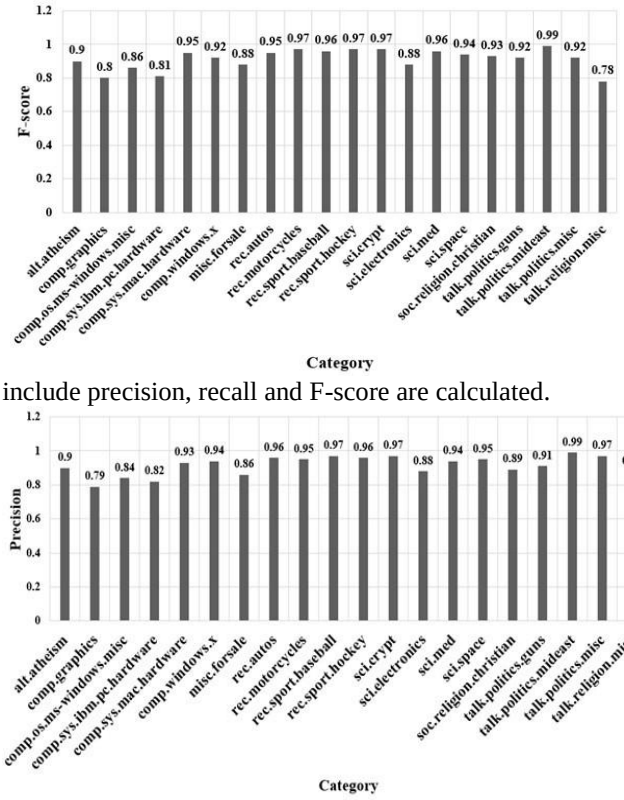


Figure 2: Precision Measure of Document Indexing using Adam Optimizer and Auto Encoder

There are 20 categories present in the newsgroup data sets and precision values are measured for each category. The precision value of the

outcomes for each category is plotted as a graph in the Figure 2. From the Figure 2 it can be inferred that the precision values are approximately present between 80% to 99%. The document related to the "talk.politics.mideast" has higher precision value and "comp.graphics" document has the lowest precision value. The precision values attained by the proposed method are more suitable for the information retrieval or document indexing.

Another measure to identify the relevance of the outcomes is recall. The outcomes of the proposed method measured in terms of the recall to analysis its performance. The recall values are measured for each category as shown in graphical representation in Figure 3. The best recall value is achieved in the "rec.motorcycles" and the lowest is in the category of "talk.religion.misc". The proposed method has achieved a considerable recall measure for the document indexing.

Figure 3: Recall Measure of Document Indexing using Adam Optimizer and Auto Encoder

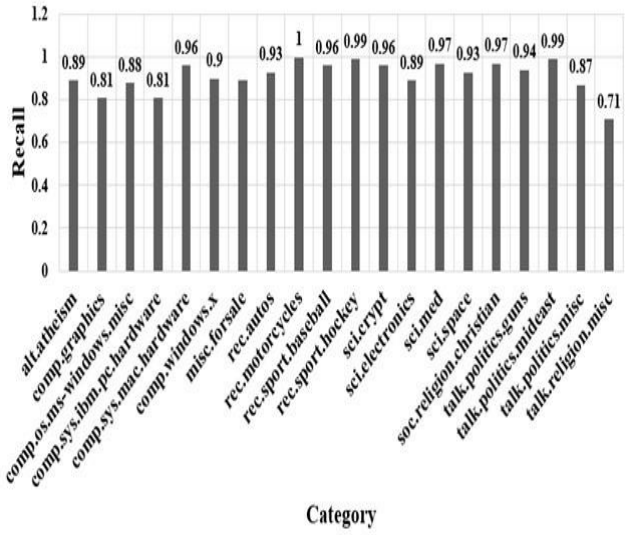
The F-score has been measured from the output of the proposed method and this value gives the harmonic mean of the precision and recall value. For each category, the F-score is calculated based on their precision and recall value. The measured F-score from the output is shown as a graph in the Figure 4. The achieved F-score is more suitable for the document indexing based on its precision and recall value. The higher F-score has been attained in the "talk.politics.mideast" category due to the higher precision and higher recall. The value of F-score is considerable for all categories in the 20 newsgroup data set.

Figure 4: F1-score Measure of Document Indexing using Adam Optimizer and Auto Encoder

The precision, recall, F-score and support values achieved by the proposed technique are offered in the Table 4. As observed, the value of f-score for each category is considerable for the document indexing. The average value and total support value are mentioned in the last record of the Table 4. The values of support are the number of samples from the response for each class. Approximately there are more than a hundred samples of supports are provided by the proposed method. The average value of the various parameters is presented in the Table 5.

Category	Precision	Recall	F-score	Support
alt.atheism	0.90	0.89	0.90	126
comp.graphics	0.79	0.81	0.80	134
comp.os.ms-windows.misc	0.84	0.88	0.86	129
comp.sys.ibm.pc.hardware	0.82	0.81	0.81	139
comp.sys.mac.hardware	0.93	0.96	0.95	129
comp.windows.x	0.94	0.90	0.92	164
misc.forsale	0.86	0.89	0.88	129
rec.autos	0.96	0.93	0.95	159
rec.motorcycles	0.95	1.00	0.97	145
rec.sport.baseball	0.97	0.96	0.96	152
rec.sport.hockey	0.96	0.99	0.97	153
sci.crypt	0.97	0.96	0.97	140
sci.electronics	0.88	0.89	0.88	158
sci.med	0.94	0.97	0.96	159
sci.space	0.95	0.93	0.94	160
soc.religion.christian	0.89	0.97	0.93	147
talk.politics.guns	0.91	0.94	0.92	126
talk.politics.mideast	0.99	0.99	0.99	152
talk.politics.misc	0.97	0.87	0.92	126
talk.religion.misc	0.88	0.71	0.78	100
avg / total	0.92	0.92	0.92	2827

Table 3: Parameter Measures for the Proposed Method in 20-Newsgroup data set



The precision, recall, F-score and support values achieved by the proposed technique are offered in the Table 3. As observed, the value of F-score for each category is considerable for the document indexing. The average value and total support value are mentioned in the last record of the Table 3. The values of support are the number of samples from the response for each class. Approximately there are more than a hundred samples of supports are provided by the proposed method. The average value of the various parameters is presented in the Table 4.

Table 4: Average Value of Different Parameters

Metrics	Value
Testing Accuracy	91.64999999999999%
Precision	91.69196465417384%
Recall	91.65192783869827%
f1 score	91.60267044267853%
Mean Squared Error	5.409621506897771
Root Mean Squared Error	2.325859305052172

Comparative Analysis

The accuracy of the proposed method is compared with other document indexing techniques.

20-Newsgroup datasets are used to measure the performance of the proposed method.

The accuracy is measured for the estimated method and compared it with the existing method in document indexing. The sparse positive bag with multiple-instance learning is used in the sparse multi instance learning technique. There are two types of methods followed in this research [59] that is structure based and weight based for document indexing. The structure based method is denoted as miStruct and weight based is denoted as miDoc, as shown in Table 5. The accuracy of the proposed method is high compared to clustering technique.

In the SALE method [61], the self-adaptive locality sensitive hashing technique is used in multi-instance learning. The proposed method has obtained higher accuracy compared with other techniques as well provided higher performance in the various categories.

Table 5: Accuracy Comparison of Each Category in 20-Newsgroup Datasets

Category	Accuracy of the existing, k=20 [59]		Accuracy of Proposed method
	miDoc	mistruct	
alt.atheism	92.2	91.1	100
comp.graphics	85	84.1	98.05447
comp.os.ms-windows.misc	78.9	78.2	98.69119
comp.sys.ibm.pc.hardware	72.7	71.9	98.16059
comp.sys.mac.hardware	78.9	81.4	99.50478
comp.windows.x	90.8	83.2	99.04492
misc.forsale	61.9	56.7	98.86806
rec.autos	93.9	88.9	99.39866
rec.motorcycles	93.3	91.7	99.71701
rec.sport.baseball	93.9	92.2	99.61089
rec.sport.hockey	71.8	71.5	99.68164
sci.crypt	83.4	85.4	99.68164
sci.electronics	92.4	86.9	98.65582
sci.med	87.9	84	99.50478
sci.space	84.9	84.2	99.29254
soc.religion.christian	78.6	76.7	99.22179
talk.politics.guns	84.7	80.1	99.29254
talk.politics.mideast	81.8	82.3	99.89388
talk.politics.misc	90.4	87.1	99.32791
talk.religion.misc	67.6	66.9	98.62045
avg / total	83.3	81.2	100

Table 6: Accuracy Comparison with State-of-Art Method

Category	SMILE [60]	SALE [61]	AEAO [Proposed]
alt.atheism	66.3	76.7	100
comp.graphics	78.5	76.3	98.05447
comp.os.ms-windows.misc	61.7	74.3	98.69119
comp.sys.ibm.pc.hardware	62.5	75.3	98.16059
comp.sys.mac.hardware	60.8	76.2	99.50478
comp.windows.x	72.6	78.1	99.04492
misc.forsale	56.7	75.6	98.86806
rec.autos	70.8	79.1	99.39866
rec.motorcycles	65.4	80.2	99.71701
rec.sport.baseball	66.9	77.8	99.61089
rec.sport.hockey	88.9	82	99.68164
sci.crypt	70.8	72.2	99.68164
sci.electronics	82.6	80	98.65582
sci.med	70.3	75.5	99.50478
sci.space	80.1	76.3	99.29254
soc.religion.christian	50.4	75.7	99.22179
talk.politics.guns	53.9	76.5	99.29254
talk.politics.mideast	69.8	79.2	99.89388
talk.politics.misc	62.6	77.6	99.32791
talk.religion.misc	56.4	75.8	98.62045
avg / total	67.4	77.02	100

VII. CONTRIBUTION OF THE RESEARCH

The growing number of documents in internet increases the need of effective methods to classify the documents. Many existing methods on classification of documents suffer from the high dimensional problem and inaccurate issues. These techniques improve the performance of the document classification, and require more computational time for their function due to processing of various feature selection from the document, which helps to classify the document accurately. The proposed method solves the problem of high dimensional data by using feature selection techniques, which eliminate the irrelevant features from the document. The extended stochastic technique (Adam optimizer) helps to improve the learning rate in the auto encoder technique. The outcome of experiment in different metrics show cased the effectiveness of the proposed method. Therefore, the proposed technique has high efficiency and can be applicable to the document indexing.

VIII. CONCLUSION

The aim of the proposed method is to increase the performance of the document indexing and can be used in various fields. 20-Newsgroup data sets consist of documents related to the various categories, used as input in this research. The training and testing documents are converted into the vector representation, then similarity such as cosine similarity and Pearson correlation is measured. The Adam optimization technique is introduced for the learning process from the large number of data. Adam optimizer calculates the objective function and increases the learning rate with help of AdaGrad and RMSProp. The auto encoder utilizes the objective function to classify the document belonging to their categories. The Adam optimizer and auto encoder technique solve some problems, which are described below.

- Adam optimizer measures the objective function and increase the learning rate, this solves the problem of the large computational time possessed by the other techniques like KNN, and centroid based method.
- The higher dimension problem is usually faced by Bag-of words method, which contains the vector representation of the document. The extensive stochastic method solves this problem by finding the relevant features.

The experimental result proved the effectiveness of the proposed technique and obtained a reliable outcome. This identifies that the proposed method classifies the document more effectively.

ACKNOWLEDGEMENT

We gratefully acknowledge the comments and suggestions from the reviewers. We would also like to thank the mentors for helpful discussions.

REFERENCES

1. García, M. A. M., Rodríguez, R. R., and L.A. (2018) Leveraging Wikipedia knowl- edge to classify multilingual biomedical documents. Artificial intelligence in medicine, 88, 37–57.
2. Du, Y., Liu, J., Ke, W. G., and X. (2018) Hierarchy construction and text classifica- tion based on the relaxation strategy and least information model. Expert Systems with. Applications, 100,157–164.
3. Zhao, Q., Kang, Y., Li, J. W., and D. (2018) Exploiting the semantic graph for the representation and retrieval of medical documents. Computers in Biology and Medicine, 101, 39–50.
4. Barbary, O. S. E. and A.S. (2018) Feature selection for document classification based on topology. Egyptian Informatics Journal, 19, 129–132.
5. Rao, G., Huang, W., Feng, Z. C., and Q. (2018) LSTM with sentence representations for document-level sentiment classification. Neurocomputing, 308, 49–57.
6. Uysal, A. K. (2016) An improved global feature selection scheme for text classifi- cation. Expert systems with. Applications, 43, 82–92.
7. Viegas, F., Rocha, L., Resende, E., Salles, T., Martins, W., Freitas, M. G., and M.A. (2018). Exploiting efficient and effective lazy Semi-Bayesian strategies for text classification.
8. Jiang, L., Li, C., Wang, S. Z., and L. (2016) Deep feature weighting for naive Bayes and its application to text classification. Engineering Applications of Artificial Intel- ligence, 52, 26–39.
9. Chen, J., Huang, H., Tian, S. Q., and Y. (2009) Feature selection for text classifica- tion with Naïve Bayes. Expert Systems with. Applications, 36, 5432–5435.
10. Zhang, L., Jiang, L., Li, C. K., and G. (2016) Two feature weighting approaches for naive Bayes text classifiers. Knowledge-Based Systems, 100, 137–144.
11. Diab, D. E. H. and K.M. (2017) Using differential evolution for fine tuning naïve Bayesian classifiers and its application for text classification. Applied Soft Comput- ing, 54, 183–199.
12. Gallego, A. J., Calvo-Zaragoza, J., Valero-Mas, J. R.-J., and J.R. (2018) Clustering- based k-nearest neighbor classification for large-scale data with neural codes repre- sentation. Pattern Recognition, 74,531–543.
13. Zhou, K., Zeng, J., Liu, Y. Z., and F. (2018) Deep sentiment hashing for text retrieval in social CIoT. Future Generation Computer Systems, 86, 362–371.
14. Bilal, M., Israr, H., Shahid, M. K., and A. (2016) Sentiment classification of Roman- Urdu opinions using Naïve Bayesian, Decision Tree and KNN classification tech- niques. Journal of King Saud University-Computer and Information Sciences, 28, 330–344.
15. Tomašev, N. B. and K. (2015) Hubness-aware kNN classification of high- dimensional data in presence of label noise. Neurocomputing, 160, 157–172.
16. Myhre, J. N., Mikalsen, K. Ø., Løkse, S. J., and R. (2018) Robust clustering using a kNN mode seeking ensemble. Pattern Recognition, 76,491–505.
17. Jung, C., Liu, Q. K., and J. (2009) Accurate text localization in images based on SVM output scores. Image and vision computing, 27, 1295–1301.
18. Yan, J., Li, J. G., and X. (2011) Chinese text location under complex background using Gabor filter and. SVM. Neurocomputing, 74, 2998–3008.
19. Manek, A. S., Shenoy, P. D., Mohan, M. V., and K.R. (2017) Aspect term extraction for sentiment analysis in large movie reviews using Gini Index feature selection method and SVM classifier. World wide, 20,135–154.
20. Liu, Y., Bi, J. F., and Z.P. (2017) A method for multi-class sentiment classification based on an improved one-vs-one (OVO) strategy and the support vector machine (SVM) algorithm. Information Sciences , ?, 394–395
21. Chen, P., Yuan, L., He, Y. L., and S. (2016) An improved SVM classifier based on double chains quantum genetic algorithm and its application in analogue circuit diagnosis. Neurocomputing, 211, 202–211.
22. Li, C. P. and S.C. (2009) Combination of modified BPNN algorithms and an ef- ficient feature selection method for text categorization. Information Processing & Management, 45, 329–340.
23. Li, C. H. and J.X. (2012) Spam filtering using semantic similarity approach and adaptive BPNN. Neurocomputing, 92,88–97.
24. Zhang, W., Tang, X. Y., and T. (2015) Tesc: An approach to text classification using semi-supervised clustering. Knowledge-Based Systems, 75, 152–160.
25. Yu, B., Xu, Z. L., and C.H. (2008) Latent semantic analysis for text categorization using neural network. Knowledge-Based Systems, 21, 900–904.
26. Yuan, H., Lau, R. X., and W. (2016) The determinants of crowdfunding success: A semantic text analytics approach. Decision Support Systems, 91, 67–76.
27. Wu, Z., Zhu, H., Li, G., Cui, Z., Huang, H., Li, J., Chen, E. X., and G. (2017) An efficient Wikipedia semantic matching approach to text document classification. Information Sciences, 393, 15–28.
28. Lee, Y., Im, J., Cho, S. C., and J. (2018). Applying convolution filter to matrix of word-clustering based document representation.
29. Mujtaba, G., Shuib, L., Raj, R. G., Rajandram, R., Shaikh, K. A.-G., and M.A. (2018) Classification of forensic autopsy reports through conceptual graph-based document representation model. Journal of biomedical informatics, 82, 88–105.
30. García, M. A. M., Rodríguez, R. P., Ferro, M. R., and L.A. (2016) Wikipedia-based hybrid document representation for textual news classification. Soft Computing,pp, 6047.
31. Liu, C., Wang, W., Tu, G., Xiang, Y., Wang, S. L., and F. (2017) A new Centroid- Based Classification model for text categorization. Knowledge-Based, 136, 15–26.
32. Liu, B., Xiao, Y. H., and Z. (2018) A selective multiple instance transfer learning method for text categorization problems. Knowledge-Based Systems, 141, 178–187.
33. Zong, W., Wu, F., Chu, L. S., and D. (2015) A discriminative and semantic fea- ture selection method for text categorization. International Journal of Production Economics, 165, 215–222.
34. Elghazel, H., Aussem, A., Gharroudi, O. S., and W. (2016) Ensemble multi-label text categorization based on rotation forest and latent semantic indexing. Expert Sys- tems with. Applications, 57,1–11.
35. Ghareb, A. S., Bakar, A. H., and A.R. (2016) Hybrid feature selection based on en- hanced genetic algorithm for text categorization. Expert Systems with. Applications, 49, 31–47.
36. Park, K. K. and G.E.O. (2017) Text mining-based categorization and user perspec- tive analysis of environmental sustainability indicators for manufacturing and ser- vice systems. Ecological, 72,803–820.
37. Xiong, S., Lv, H., Zhao, W. J., and D. (2018) Towards Twitter sentiment classifi- cation by multi-level sentiment-enriched word embeddings. Neurocomputing, 275, 2459–2466.
38. Papagiannopoulou, E. T. and G. (2018) Local word vectors guiding keyphrase ex- traction. Information Processing & Management, 54, 888–902.
39. Henry, S., Cuffy, C. M., and B.T. (2018) Vector representations of multi-word terms for semantic relatedness. Journal of biomedical informatics, 77, 111–119.
40. Lee, G., Jeong, J., Seo, S., Kim, C. K., and P. (2018) Sentiment classification with word localization based on weakly supervised learning with a convolutional neural network. Knowledge-Based Systems, 152,70–82.
41. M., Q., H., H., Lu, L., and L. (2018) Bag of meta-words: A novel method to repre- sent document for the sentiment classification. Expert Systems with. Applications, 113, 33–43.
42. Xia, P., Zhang, L. L., and F. (2015) Learning similarity with cosine similarity en- semble. Information Sciences, 307,39–52.
43. Kalhori, H., Alamdari, M. Y., and L. (2018) Automated algorithm for impact force identification using cosine similarity searching. Measurement, 122, 648–657.
44. Bermudez-Edo, M., Barnaghi, P. M., and K. (2018) Analysing real world data streams with spatio-temporal correlations: Entropy vs. Pearson correlation.
45. Mu, Y., Liu, X. W., and L. (2018) A Pearson's correlation coefficient based decision tree and its parallel implementation. Information Sciences, 435, 40–58.
46. Kingma, D. B. and J. (2014). Adam: A method for stochastic optimization. arXivp.
47. Zuo, Y., Feng, M. L., and G. (2014) The effect of temperature gradients on the parameters of Adam–Gibbs model. Journal of Non-Crystalline Solids, 387, 86–93.

48. Enríquez, F., Troyano, J. L.-S., and T. (2016) An approach to the use of word em- beddings in an opinion classification task. Expert Systems with. Applications, 66, 1–6.
49. Su, J., Wu, S., Zhang, B., Wu, C., Qin, Y. X., and D. (2018) A neural generative autoencoder for bilingual word embeddings. Information Sciences, 424, 287–300.
50. Wang, X., Ma, Y., Cheng, Y., Zou, L. R., and J.J. (2018) Heterogeneous domain adaptation network based on autoencoder. Journal of Parallel and Distributed Com- puting, 117, 281–291.
51. Xu, J., Xiang, L., Liu, Q., Gilmore, H., Wu, J., Tang, J. M., and A. (2016) Stacked sparse autoencoder (SSAE) for nuclei detection on breast cancer histopathology im- ages. IEEE transactions on medical, 35, 119–130.
52. Irsoy, O. A. and E. (2017) Unsupervised feature extraction with autoencoder trees.
53. Neurocomputing, 258, 63–73.
54. Cheng, W., Zhang, X., Pan, F. W., and W. (2016) HICC: an entropy splitting-based framework for hierarchical co-clustering. Knowledge and Information Systems, 46, 343–367.
55. Venkatesh, G. A. and K. (2018) Map Reduce for big data processing based on traffic aware partition and aggregation. Cluster Computing , 1–7.
56. Singhal, V. M. and A. (2018) Majorization Minimization Technique for Optimally Solving Deep Dictionary Learning.
57. Soleimani, H. M. and D.J. (2017) Exploiting the value of class labels on high- dimensional feature spaces: topic models for semi-supervised document classification. Pattern Analysis and Applications , 1–11.
58. Ranjan, N. P. and R.S. (2018) LFNN: Lion fuzzy neural network-based evolutionary model for text classification using context and sense based features. Applied Soft Computing.71.994-1008 ,
59. Feng, G., Li, S., Sun, T. Z., and B. (2018) A probabilistic model derived term weight- ing scheme for text classification. Pattern Recognition Letters, 110, 23–29.
60. Yan, S., Zhu, X., Liu, G. W., and J. (2017) Sparse multiple instance learning as document classification. Multimedia tools and. applications, 76, 4553–4570.
61. Xiao, Y., Liu, B., Hao, Z. C., and L. (2014) A similarity-based classification frame- work for multiple-instance learning. IEEE transactions on, 44, 500–515.
62. Xu, D., Wu, J., Li, D., Tian, Y., Zhu, X. W., and X. (2017) SALE: Self-adaptive LSH encoding for multi-instance learning. Pattern, 71, 460–482.

AUTHORS PROFILE

Y.Krishna Bhargavi is presently engaged in pursuit of Ph.D in Big Data Analytics and Cloud Computing registered at JNTUK, Kakinada. Prior to Ph.D, she had earned Bachelors of Technology in Information Technology and Masters of Technology in Software Engineering. She published papers in various International Journals and Conferences.

Dr Y S S R Murthy did his graduation from Andhra University, Post graduation in Computer Science & Engineering from Andhra University and PhD from JNTU Hyderabad. He worked in Industry for 6 years starting career as Trainee engineer and grew up to manager and also worked as ISO Coordinator. He also established a training Institute where developed passion towards teaching. He turned into an academician in the year 2001 in professional Engineering colleges. He is guiding 5 PhD's, guided 20 M.Tech Projects and published more than 15 research papers, presented more than 10 papers in conferences and organized 2 conferences and 5 workshops.

Dr.O.Srinivasa Rao did B.Tech, M.Tech and obtained Ph.D in CSE from JNTUK, KAKINADA. His Ph.D specialization is cryptography and Network security. He presented research papers more than 60 papers in various International journals and two papers in National conferences, one paper in international conference. He had more than 20 years of teaching experience and he was former Head of CSE at University College of Engineering vizianagaram, JNTUK and currently working as Professor of CSE at University College of Engineering, JNTUK, Kakinada. He guided more than 90 M.Tech and MCA students' projects and one Ph.D .Currently he is guiding 2 Ph.D and 8 M.Tech, 2 MCA students projects. His fields of interest are Cryptography, Network security, Image Processing and Data Mining.