International Journal of Grid and Distributed Computing
Vol. 14, No. 1, (2021), pp. 587-595

Evolutionary and Incremental Text Document Classifier using Deep Learning

Nihar M. Ranjan
Post Doc. Scholar
Lincoln University, Malaysia

MidhunChakkaravarthy
Faculty of Engineering
Lincoln University, Malaysia

Abstract
Currently most of the generated data are stored in text format so text mining is believed to have a high commercial potential value. There are many sources for knowledge extraction, still unstructured text considered as the largest readily available source of knowledge. These data which is mostly available in digital format need to manage. Text document classification is basically a task to automatically categorize the text documents in some pre-defined classes.
A lot of research work is already done in this domain and many classifiers have been developed including some ensemble classifiers. The classification accuracy of the classifier depends on many parameters such as preprocessing, features selection, data set, methodology etc. Any of these mentioned parameters can be significantly influential in increasing the accuracy of the classifier.

1. Introduction

Text classificationcan be defined as follows: Let's say we have a set of training records R= *{R*1*, R2*.... *,RN}*, each record is assigned with a class valuederived from a set of *k* discrete values indexed by *{*1 . . . *k}*. Two types of data are used to construct and validate the classifier out of which training data is used to construct a classification model[5]. The classification modelevaluates the features in the underlying record to one of the pre-defined class labels. Fora test instancethe class label is not known, the training modelof the classifier is used to predict the class label for this test instance. There are two version of the classification,a specific label is explicitly assigned tothe test instance in the hard version of the classification problem, whereas a probability value is assigned to the test instance in the soft versionof the classification problem.Some other variations allow ranking of different class possibilities for atest instance, or it may allow the assignment of more than one classlabels to a testinstance[2].

i. Applications

Text classification had a wide variety of applications in the domains of text mining. Some ofthe domainwhere text classification is used extensively are as follows:

Filtering of news article and its Organization: Now a daysmost of the news articles are electronic in nature and a large volume of news articles are created daily by an organization. In such cases, it is very tedious and hectic to organize these news articles manually. To make this process hassle free automated text categorization methods can be very useful. for news filtering and organization. This application of news filtering and organization is also referred to as text filtering[4].
Document Organization and Retrieval: This application is basically beyond the domain of news filtering and organization which is useful for many diversified domains. A variety ofsupervised learning methodscan be used for document organization in differentarea. These may include large digital contents of documents, web documents, scientific literature, historical literatures or even social media streams. The documents collection organized hierarchically can be very easy and useful for browsing and retrieval of the desired contents[10].
Sentiment Analysis: Sentiment analysis is a form of opinion mining wherecustomer reviews or opinions are analyzed.This can be treated as short text documents and can be mined to determine useful information from the review.
Email Categorization and Spam Filtering: This application categorize different types of email as per their content**.** It is always desirable to classify an email in order to determine whether the email is of use

ISSN: 2005 - 4262 IJGDC

or junkemail in an automated way. This type of application is also referred to as email filtering or spam filtering.

ii. Broad classification Techniques

A lot of diversified techniques have been designed for the text document classification. In this section we will discuss some broad categories and key methods generally usedfor classification task[4].

Decision Tree: Decision tree algorithms are designed with the use of a tree like structure which supports hierarchical division of the given data space with use different features of the text. In order to create class partitions which are more skewed in terms of their class distribution the hierarchical division of the data space is designed. Usually we determine the partition that it is most likely to belong to, and use it for the purposes of classification for a given text instance.

*Rule-based Classifiers:*Here we determine the pattern of the wordsused in the text documents which are most likely to be match with the different predefined classes. A set of rules is constructed, in which the left-hand side corresponds to pattern of the word, and the right-hand side points to a predefined class label. So, these set of rules are used for the purposes of the text document classification[4].

*SVM Classifiers:*Support Vector Machine Classifiers partition the data space with the use of linear or non-linear delineations between the different classes. The basic and key concept inthis classifier is to decide the optimal boundaries between the different class labels and use them for the purposes of text document classification.

*Neural Network Classifiers:*Neural networks are most innovative and advanced classifier used in wide range of domains for the purpose of the text document classification. One of the significant differencesof neural network classifiers is that it adapts the use of word features and it work well in the context of the text data. Neural network classifiers somewhat related to SVM classifiers, both belongs to the category of discriminative classifiers[4].

*Bayesian Classifiers:*In Bayesian classifiers (which is also called probabilistic classifier), an attempt is made to build a probabilistic classifier based on modeling the extracted word features in different predefined classes. The basic idea is to classify the underlying text based on the posteriorprobability of the text documents belonging to the different classes and the word present in the documents. Bayesian classifier is also known as generative classifier[9].

Nearest neighbor classifier: The nearest neighbor's algorithm (NN) is an algorithm for classifying the objects based on the closest training examples in the feature vector space. The k-nearest neighbor algorithm is one of the simplest of all machine-learning classification algorithms. The feature space is partitioned into different regions based on locations and labels of the training data. A data point in the feature space is assigned to the class if it belongs to the most frequent class label among the k nearest training samples.Euclidean distance is used to measure the distance between the data points in the feature space. One of the drawbacks of this distance metric is that it only works with the numerical data.In case of textdocument classification another metric like hamming distance or overlap metric can be used[3].

*Other Classifiers:*It has been observed thatalmost all classifiers adapted to the text data and work well with the text document classification. Some of the other classifiers include ensemble classifiers, genetic algorithm-based classifiers which can be also used to classify the text data.

2. Existing related work

Sense based analysis of the text:

This work suggests to extract the actual semantic meaning of the text by using Wordnet ontology. The semantic processing technique uses a bag of word approach for text classification. The two basic problems of synonyms and hyponyms are tried to handle in this research work. A synonym is a word which can be used to substitute another word without change is meaning. Hyponym is the lexical relationship between the meaning of the words[12].

Context based analysis of text:

In this work the selected features is mapped with its context in which they are used , this leads to better classification accuracy. The feature selection of the proposed connectionist model is based on entropy function and the extraction process utilizes semantic word processing along with the contextual approach

ISSN: 2005 - 4262 IJGDC

and topic terms to extract relevant features. This process involves semantic word processing, contextual word processing, key term identification and context term identification[13].

Design and development of an automatic classifier:

In this work an automatic text classification system has been proposed for classifying the bulk documents in the database based on the predefined labels. The proposed classification methodology has used semantic processing in feature extraction to reduce the dimensionality problem by avoiding the repetition of words as well as the occurrence of words with the same meaning. A new training algorithm called the back-Propagation Lion Algorithm (BP Lion) has been proposed through integrating the back propagation and the Lion algorithm to update the weights in the hidden neurons as well as to reduce the error measure of Neural network[12].

Design and development of a fuzzy neural network classifier:

In this research an incremental learning technique is proposed using contextual semantic word processing and hybrid fuzzy neural network for text document classification. Contextual and semantic word processing is used to extract the features and entropy method was used to reduce the space dimensionality of the features. A Lion optimization algorithm is used to update the weights of the network and to classify the dynamic database[13].

3. Incremental Learning:

Incremental learning refers to the phenomenon of accumulation and management of knowledge over time. The basic hypothesis of incremental learning are updated in response to the available new training data without the consideration of the previous data[1]. It is an intelligent knowledge learning system, which continuously learns from new samples and maintains the already learned knowledge. It is equivalent to the human learning aspect for the constant enhancement of the knowledge by accepting the evolution of new things. This learning strategy is economical, temporal and spatial as it discards the storage requirement and reprocessing of older instances.It is most suitable for learning tasks, where the training datasets are available for a prolonged time. The incremental learning classifier is productive in incremental learning system training, which makes the study on newly arrived data chunks along with the consideration of knowledge gathered from the older training data[6].

The performance of text categorization is largely influenced by the size of the trainingdata set (more exhaustive training set better the classification accuracy). The accuracy of classification and learning capability can be enhanced by increasing the size of the training dataset. But collection of huge training data set is time-consuming, costly and a complex process. So, the assumption of a comprehensive training sample is non-realistic, instead a classifier can be designed with the incremental learning methods. When the additional training samples are included, the classifier must be adaptive, retaining the previously acquired knowledge. As the training set would become massive, it won't be possible to accumulate all the old data. It is highly recommended to train the classifier incrementally, with the arrival of the new sample rather than retraining all the training set. In the incremental classifier, the initial training sample is denoted as codebook vectors, and the learning phase includes the incremental update of codebook vectors on the arrival of new training data[7]. The current values of the codebook vector are extremely influenced by the arrival of new data. When a classifier is trained with many samples, for further training, only a minimum number of samples are required. The learning with a minimum number of new samples will lead to unlearnt distribution on the basis of older samples. The assignment of optimal weight for the learned distribution must be made by a robust incremental learning approach. The two main advantages are, it uses the entire history of training output and improves recognition efficiency by dismissing unwanted samples for reducing training set. The other advantages of incremental text classification over the other trivial text classification schemes are: the cost of storage is minimized by eliminating older samples and faster learning by the usage of historical results of training [8].

4. Major Research Challenges:

On the basis of extensive literature survey on machine learning algorithms for text document classification following research challenges are identified which are stated as follows:

ISSN: 2005 - 4262 IJGDC

i. One of the major issues about text categorization using the chat is that there are numerous short words, which are not referred in proper English. The existing algorithms do not offer an effective solution for clustering these text data.

ii. In spite of the text preprocessing like removing the stop words and stemming dimensionality of the text features are remains high and because of the high dimensionalitythe classification process comparatively takes more time. Sparsity is another issue with the text features which must be addressed.

iii. Unstructured text is one of the major challenge need attentions, dealing with large number of attributes, ambiguity in text meaning are quite complex to handle. Handling the missing metadata and finding an effective classification algorithm is also a challenge.

iv. The biggest problemin text miningdomain is about the presence of the synonyms, polysemy, and antonyms in the text documents that makes the categorization of the documents difficult and complex as the semantic (context and sense) meaning of the words are not very clear. The presence of the abbreviation also create additional complexities and the existing methods fail to categorize the text holding the abbreviations.

v. The feature vector that is applied to represent a text document must consider the complex semantics that are usually in the form of natural language. One of the challenges in text classification is the creation and selection of quality features. The features extracted must be effective in such a way that it can be applied over a range of class definitions.

vi. The classification accuracy of the machine learning approach is another issue in the automatic text classification. The challenges like the architectural design, the training process and the objective criterion can be addressed with a deep learning method.

5. Problem Definition:

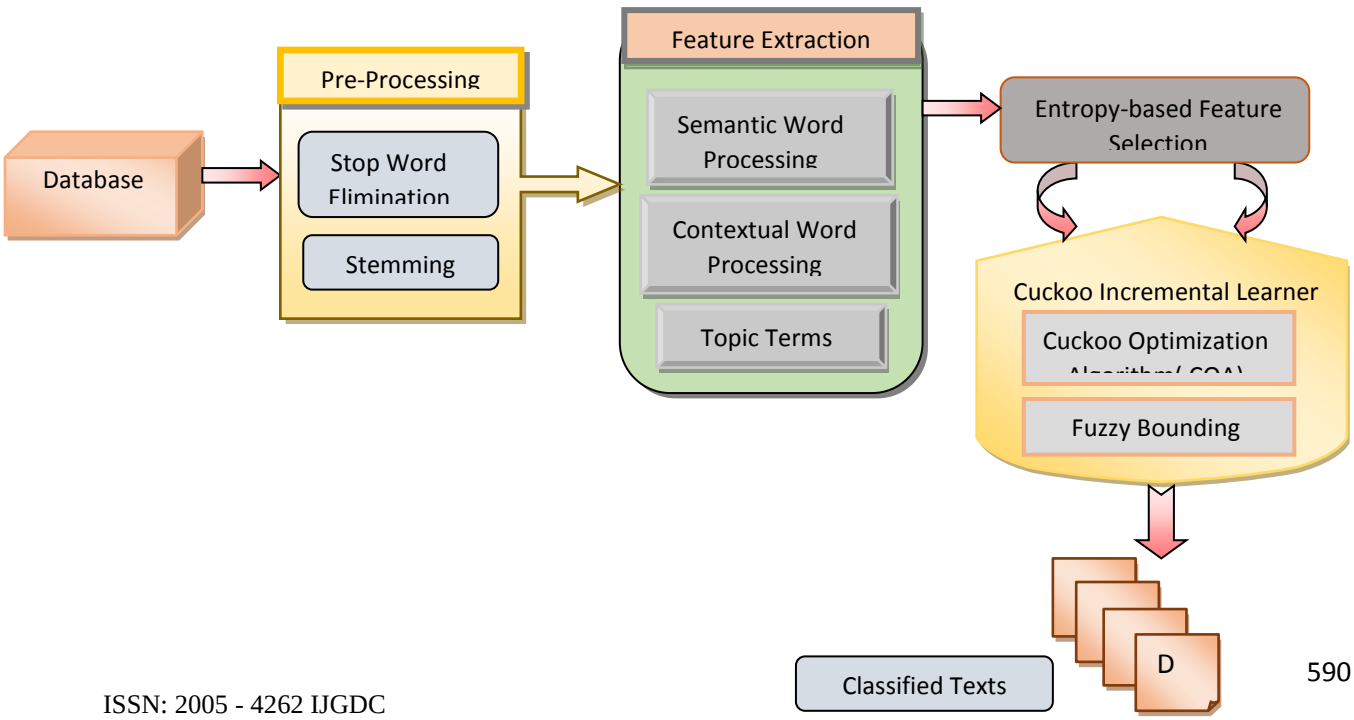
On the basis of my past research work and extensive literature survey, I planned to do further research to enhance the classification accuracy of the classifiers, brief outline of my ideas are as follows:

i. Selection/extraction of quality features and reduce the feature dimensions

ii. Applying deep learning architecture to develop evolutionary framework for Text Document Classification

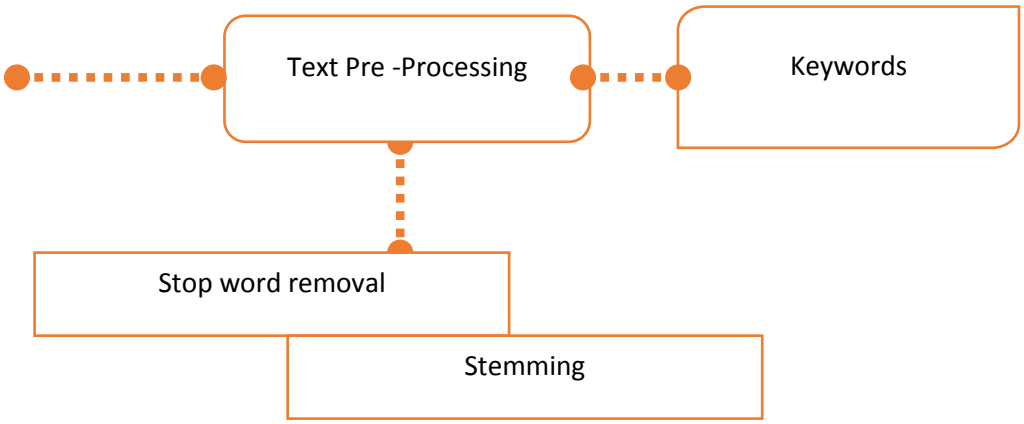
iii. Evaluate the system performance on various data sets and quality parameters (like precision, recall etc.).

6. Proposed Methodology:



ISSN: 2005 - 4262 IJGDC

i. *Text preprocessing:* This is the very first and basic step in the implementation of any of the text classification algorithms.



Pre-processing steps basically involves two important and well-known steps, namely stop word removal and stemming as shown in the **figure**. Stop words are the symbolic grammatical words that occurs very frequently in any text document but didn't have any discrimination power. These words is not taken in to consideration during the feature selection so not useful in the classification, So these words are removed during the pre-processing steps. Next step is stemming where the words in the text are brought to its root or base form by. By applying these two steps useful keywords are extracted from the text document of the data base[12].

The document set is represented as:

$$D = db_i; 1 \le i \le N$$

where, N denotes the number of documents

Based on the different topics of data collection, every document in the data set is represented as db_i.

$$db_i \in B_j; 1 \le j \le k$$

After the stop word removal and the stemming the data base db_i have the useful extracted keywords w_l and it is given by.

$$db_i \in w_l; 1 \le l \le n$$

ii. Semantic word processing:

The synonym are the words have the same meaning even different representation, are extracted from the keywords and expressed as,

$$S^W = \{S_1^W, S_2^W, \ldots, S_m^W\}$$

A hyponym provides a relationship between a word and a more distinct word. It is represented as shown in equation

$$H^W = \left\{ H_1^W, H_2^W, \ldots, H_m^W \right\}$$

After the extraction of the synonym and hyponym, a word dictionary is created using the keywords generated during semantic word processing. By integrating the synonym, the hyponym and the keywords, the constructed word dictionary is expressed using the equation

ISSN: 2005 - 4262 IJGDC

$$D = \left\{ x_j^i \left\| S_j^{W^i} \right\| H_j^{W^i} \right\}$$

iii. Feature selection by using entropy model

Let the dimension of the feature database be [$M \times N$], and the chosen keywords are arrangedon the basis of theclass of dimension denoted as Q. A new database is constructed by coordinating every keyword to that keyword in the class. Thus, the constructed database is speculated to have a distinct dimension reduced by t , i.e., $M \times (N - Q)$. For the number of classes N , in a finite dataset B , the entropy model is formulated as[13],

$$E = -P \log P$$

iv. Network initialization

. The neural network is initialized with the input feature set provided as the input to the neural network be $f_l = \{f_1, f_2, \ldots, f_N\}$ and the obtained output feature set be $o_l = \{o_1, o_2, \ldots, o_k\}$. The representation of the hidden layer is as follows:

$$h_j = \sum_{j=1}^{N_h} \left[f_j^t * w_t^i \right]$$

The estimation of output vector can be done as

$$O_j = \sum_{j=1}^{N_h} \left[h_j * w_t^o \right]$$

The error estimation is done by

$$E^t = \frac{1}{N_t} \sum_{j=1}^{N_t} \left[O_j^t - G_j^t \right]$$

Weight is updated after the error calculation by the equation

$$w^t = \min_{E_t} \left(w^t \right)$$

where, the optimal neuron weight is indicated as w^t and the error measure to be minimum is denoted as E^t .

v. Fuzzy bound incremental learning

In incremental learning during the inclusion of a new instance f^{t+1} in the network, the error measure E^{t+1} is computed by updating neuron weights without insight about the older instances. If the estimated error at a current instance is lower than the error measure of the previous instance, then the neuron weight is computed using the above equation.

A new fuzzy bond approach with Cuckoo Optimization Algorithm (COA)is used, the optimal weight is chosen by considering the modification degree.

$$MD = \left\| w_j^{t-2} - w_j^{t-1} \right\|$$

ISSN: 2005 - 4262 IJGDC

The estimated weight at $t-1$ instance is denoted as w_j^{t-1} and the estimated weight at the $t-2$ instance is denoted as w_j^{t-2}. Hypothesis weight is computed by using the equation,

$w^h = \left(w_j^t \pm F^B\right)$

The fuzzy bond is calculatedon the basis of a triangular membership function,

$$F^B = \frac{\alpha}{T^F}$$

vi. Weight update using CuckooOptimization Algorithm (COA)

COA is a meta heuristic globaloptimization algorithm based on obligated brood parasitic behavior of Cuckoo birds. In this algorithm eggs in the nest are considered as set of solutions and Cuckoo eggs are considered as new solution. New and better solutions will replace the less fit solutions. Each Cockoo can lay one egg at a time step, high quality nests will carry onto next generations. Number of host nest is fixed and Pf is the probability of discovering a foreign egg. Host birds can throw away the egg or leave the nest[15,16,17].

Initialization: n is the number of host nest, P_f is the probability of finding the foreign egg.

Generate the initial solution $x_i^{(t)}$, evaluate f($x_i^{(t)}$), Generate a new solution $x_i^{(t+1)} = x_i^{(t)} + \alpha$

Evaluate f ($x_i^{(t+1)}$), If f ($x_i^{(t)}$) < f ($x_i^{(t+1)}$), replace $x_i^{(t)}$ with $x_i^{(t+1)}$. So finally, $w^{(t+1)} = w^{(COA)}$

Where $w^{(COA)}$ is the weight update based on COA when the error measure at $t+1^{th}$ instant is higher than the error measure at the t^{th} instant, i.e.$E^{t+1} > E^t$. At new arrival of data instant, new feature is includedand the optimal weight is chosen within the fuzzy bound.This iterative process is continued until $E^{t+1} > E^t$. The process is terminated only when the maximal number of iterations,max_i is reached

7. Results and Discussions:

WEBKB (World Wide Knowledge Base) is used for the experimentation, It is a publicly available dataset collected by four different universities. There are 8282 documents belongs from 7 different categories. Two evaluation metrics is used namely Accuracy and Error.

i. $$Accuracy = \frac{D_{TP} + D_{TN}}{D_{TP}+D_{FP}+D_{FN}+D_{TN}}$$

Accuracy is calculated using the standard confusion matrix by using four well known parameters: True positive(TP), True negative(TN), False positive(FP), False negative(FN). Whole data set are divided into 4 equal number of chunks to support the incremental learning. Experimentation is performed with the chunk size of 2, 3 and 4, however here in the paper only the results with chunk size 4 is included.

Table1:Accuracy based values for chunk size 4 with WEBKB dataset

Algorithms	**T=50**	**T=75**	**T=100**
IBP	81.43	82.30	82.87
FI-BP	81.67	82.21	83.10
I-BPLION	81.25	82.65	83.86
COA	**86.98**	**88.00**	**90.76**

ISSN: 2005 - 4262 IJGDC

The result of the COA algorithm is compared with three existing algorithms, Incremental Back Propagation(IBP), Fuzzy Incremental Back Propagation (FIBP), Incremental Backpropagation Lion algorithm(IBPLION) which work well for the text document classification. Here classification accuracy is observed with different threshold values of 50, 75 and 100. Maximum classification accuracy of 89.76 is obtained with chunk size of 4 and threshold 100.

ii. Error$E^t = \frac{1}{N}\sum_{i=1}^{N}[P - Q]$

For the proposed COA algorithm the error rate is minimum as compared to the three existing algorithms. Error rate varies at the different threshold values and the minimum error rate of 4.90 is obtained with the data set of chunk size 4 and the threshold values of 100.

Table 2: Error values for chunk size 4 with WEBKB dataset

Methods	**T=50**	**T=75**	**T=100**
IBP	9.34	9.21	9.02
FI-BP	8.65	8.22	8.13
I-BPLION	8.20	8.03	7.90
COA	**5.12**	**5.10**	**4.90**

8. Conclusions:

The prime intention of this research work is to design and develop an efficient text document classifier for the unstructured data. Incremental learning approach is used to support many applications where data are changing over the time period. Backpropagation. Neural network methodologies are used to update the weights. Further Cuckoo Optimization Algorithm (COA) is used to minimize the error rate and enhance the classification accuracy. The classification accuracy of 90.76% is achieved with data chunk size of 4 and the error rate is reduced to 4.90. To minimize the dimensions of the feature space an entropy model is used. Important and quality features are extracted by using contextual-semantic word relations. A dynamic database is considered to improve the classifier's learning ability. The proposed model is based on incremental learning where classifier classifies the new instances without any knowledge of the previous instances. Further classifier can be designed and develop to manage and resolve the imbalanced data issues.

References:

1. Chen ZhiHang, Huang Liping et al., "Incremental Learning for Text Document Classification," International Joint Conference on Neural Networks, Florida, USA, pp. 12-17, August 2007.
2. Kim H., Howland P. et al. "Dimension Reduction in Text Classification with Support Vector Machines," Journal of Machine Learning Research, Vol.6, pp. 37–53, Jan. 2005.
3. King G., Lam P. et al. "Computer assisted Keyword and Document set Discovery from Unstructured Text," American Journal of Political Science, Vol. 456, 2014.
4. Fabrizio Sebastiani, "Machine Learning in Automated Text Categorization," ACM Computing Surveys, Vol.34, No.1, pp. 1–47, 2002.
5. Varma Bhavitha, Kumar Senthil, "A Survey on Text Categorization," International Journal of Advanced Research in Computer and Communications Engineering, Vol. 5, No. 8, August 2016.
6. Wang C., Zhang J., et al., "Incremental Learning for Compressed Pornographic Image Recognition," IEEE International Conference on Multimedia Big Data, Beijing, pp. 176–179, Apr. 2015.

ISSN: 2005 - 4262 IJGDC

7. J. Ratsaby et al. "Incremental Learning with Sample Queries," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, pp. 883 -888, August 1998.
8. Bohn C., "An Incremental Unsupervised Learning Scheme for Function Approximation," Proceedings of International Conference on IEEE Neural Networks, pp. 1792-1797, 1997.
9. Tang B., He H. et al., "A Bayesian Classification Approach using Class-Specific Features for Text Categorization," in IEEE Transactions on Knowledge and Data Engineering, vol. 28, pp. 1602-1606, June 1 2016.
10. Menaka S. and Radha N., "Text Classification using Keyword Extraction Technique," International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 3, No. 12, December 2013.
11. P Marques C., Elias Oliveira et al., "Multi-label Incremental Learning Applied to Web Page Categorization," Neural Computing and Applications, Vol. 24, pp. 1403–1419, May 2014.
12. RanjanNihar, R. S. Prasad, " Automatic Text Classification using BP-Lion Neural Network and Semantic Word Processing" Imaging Science Journal, Taylor & Francis, ISSN: 1368-2199, Sept. 2017.
13. Ranjan Nihar, R. S. Prasad, " LFNN: Lion Fuzzy Neural Network based evolutionary model for text classification using context and sense based features", Applied Soft Computing Journal, Elsevier, PP 994-1008, ISSN: 1568-4946,July 2018.
14. Fu L., Hsu H. et al. "Incremental Backpropagation Learning Networks," IEEE Transactions on Neural Networks, vol. 7, no. 3, pp. 757-761, 1996.
15. She-Xin y., Deb Suash, " Cuckoo Search via Levy Flight", IEEE: world congress on nature and biologically inspired computing, January 2010.
16. FisterIztok et al., " A Comprehensive Review of Cuckoo Search: Variants and Hybrids", International Journal of Mathematical Modelling and Numerical Optimization, Vol 4, 2013.
17. Shelke Priya, Rajesh P., " An improved anti-forensic JPEG image compression using Least Cuckoo Serach Algorithm", Imaging Science journal, Taylor & Francis, 2017.

ISSN: 2005 - 4262 IJGDC