Optimized deep belief network and entropy-based hybrid bounding model for incremental text categorization

V. Srilakshmi
CSE, JNTUA, Anantapur, India

K. Anuradha
CSE, GRIET, Hyderabad, India, and

C. Shoba Bindu
CSE, JNTUA, Anantapur, India

Received 18 March 2020
Revised 30 May 2020
Accepted 2 June 2020

Abstract

Purpose – This paper aims to model a technique that categorizes the texts from huge documents. The progression in internet technologies has raised the count of document accessibility, and thus the documents available online become countless. The text documents comprise of research article, journal papers, newspaper, technical reports and blogs. These large documents are useful and valuable for processing real-time applications. Also, these massive documents are used in several retrieval methods. Text classification plays a vital role in information retrieval technologies and is considered as an active field for processing massive applications. The aim of text classification is to categorize the large-sized documents into different categories on the basis of its contents. There exist numerous methods for performing text-related tasks such as profiling users, sentiment analysis and identification of spams, which is considered as a supervised learning issue and is addressed with text classifier.

Design/methodology/approach – At first, the input documents are pre-processed using the stop word removal and stemming technique such that the input is made effective and capable for feature extraction. In the feature extraction process, the features are extracted using the vector space model (VSM) and then, the feature selection is done for selecting the highly relevant features to perform text categorization. Once the features are selected, the text categorization is progressed using the deep belief network (DBN). The training of the DBN is performed using the proposed grasshopper crow optimization algorithm (GCOA) that is the integration of the grasshopper optimization algorithm (GOA) and Crow search algorithm (CSA). Moreover, the hybrid weight bounding model is devised using the proposed GCOA and range degree. Thus, the proposed GCOA + DBN is used for classifying the text documents.

Findings – The performance of the proposed technique is evaluated using accuracy, precision and recall is compared with existing techniques such as naive bayes, k-nearest neighbors, support vector machine and deep convolutional neural network (DCNN) and Stochastic Gradient-CAViaR + DCNN. Here, the proposed GCOA + DBN has improved performance with the values of 0.959, 0.959 and 0.96 for precision, recall and accuracy, respectively.

Originality/value – This paper proposes a technique that categorizes the texts from massive sized documents. From the findings, it can be shown that the proposed GCOA-based DBN effectively classifies the text documents.

Keywords Incremental learning, Bounding model

Paper type Research paper

International Journal of Web
Information Systems
© Emerald Publishing Limited
1744-0084
DOI 10.1108/IJWIS-03-2020-0015

1. Introduction

The progression in internet technologies has raised the count of document accessibility, and thus, the documents available online become countless. The text documents comprise of research articles, journal papers, newspapers, technical reports and blogs. These large documents are useful and valuable for processing real-time applications. Also, these massive documents are used in several retrieval methods. Text classification plays a vital role in information retrieval technologies and is considered as an active field for processing massive applications. The aim of text classification is to categorize the large-sized documents into different categories on the basis of its contents (Mohammad *et al.*, 2018). There exist numerous methods for performing text-related tasks such as profiling users, sentiment analysis and identification of spams, which is considered as a supervised learning issue and is addressed with text classifiers (Berge *et al.*, 2019). The text classifier contains different sub-processes in which some of them are more flexible such that it can be adapted for solving the issues of supervised learning whereas other classifiers are specially developed for addressing a specific task using expensive processes such as syntactic analysis and lemmatization (Tellez *et al.*, 2018). The text classification is also used for processing or assigning predefined categories for each text. The word is derived from the texts, which contains more than one paragraph and text must be illustrated from each other. The following text categorization contains three steps: the first task contains category predefinition and allocation of sample texts and novice classification. Moreover, the text categorization is helpful in filtering spam emails, detecting fraudulent documents (Sudhakar *et al.*, 2013), spotting topics and analyzing sentiments (Taeho, 2019).

In-text categorization, the documents are designed using the vector space where each word is considered as a feature. In vector space model (VSM), the features values are termed as word frequency or term frequency–inverse document frequency (TF-IDF). The major issue in text categorization is addressing the huge dimensionality of the obtained feature space. A huge number of features maximize the time taken for computation and degrades the classification accuracy. The selection of features and extraction of features are the two major tasks adapted in minimizing the dimensionality of text feature space. Here, the extraction of features is carried out for producing a new set of features by integrating or converting the original ones whereas in selecting features, the dimensions of the space are minimized by choosing the most protuberant feature (Labani *et al.*, 2018). In addition, the feature selection techniques can be widely classified into three groups, namely, wrapper, embedded and filters. The feature selection is mostly used for categorizing the texts. Numerous filter mechanisms had been devised named chi-square (χ^2), information gain (IG) and document frequency (DF) (Yang and Pedersen, 1997). The *N*-gram language model was devised for capturing term dependencies. *N*-gram is based on the corpus that they are trained. In Tang *et al.* (2016), Jeffreys–multi hypothesis divergence is devised for selecting features for text categorization.

Usually, the text is defined as a feature vector in VSM, in which the dimension of text feature vectors is very high and it can be 10s or even 1,000s. Moreover, high dimensional vector space not only minimizes the accuracy for representing texts but also maximizes the burden in the classification learning algorithm. Thus, the minimization of dimensions is highly recommended. There exist certain limitations in conventional feature selection methods such as DF, IG and χ^2 test (Chen *et al.*, 2018). Moreover, text data is devised using a VSM using high dimensional data as the word count and can grow 1,000s of data sets at a moderate-sized data set. It consists of a huge number of extraneous features, which destroys the performance of the classifier for text categorization (Kim and Zzang, 2018). Although text data sets contain a huge number of terms and this can destroy the accuracy (Lee *et al.*, 2019). Numerous classification mechanisms such as SVM (Cai and Hofmann, 2004; Ninu Preetha and Praveena, 2018), neural networks (NNs) (Ghuge *et al.*, 2019; George and Rajakumar, 2013), derived probability classification, nearest neighbor classification, are used for classifying texts (Camastra and Razi, 2019; Jo, 2019). Incremental learning algorithms are widely used for enhancing the data or a huge number of data such as log data or intelligence data. Decision trees or NN are the most widely used algorithms for text categorizations (Ma *et al.*, 2017). An incremental text classifier uses Kullback Leibler distance (Song *et al.*, 2009) for determining public transit issues and events from online social media. Naive bayes (NB) (Kim *et al.*, 2006) uses text classification to provide enhanced performance in incremental learning. Though the solution generated from the categorization is easier and effective, but the estimation of a certain parameter and the new data arrival lead to many complications. The other challenges are the minimization of secret keys for the data users, mapping of anonymous data regarding the existing topics (Wang and Al-Rubaie, 2015) and efficiency to search the required document (Wang *et al.*, 2018). The contemporary information operated by the availability of hypermedia and the World Wide Web leads to huge data and posed a rising challenge for several information retrieval systems in efficiently storing and retrieving the information (Charikar *et al.*, 2004). In (Yin and Xi, 2017), grasshopper crow optimization algorithm (GCOA) because of the diversity and ambiguity of conversational language becomes complex to determine the significant information that is hidden in the huge information. These challenges are considered as a motivation and a new method is proposed for the incremental clustering.

The main aim of the research is to model a technique that categorizes the texts from huge documents. At first, the input documents are pre-processed using the stop word removal and stemming technique such that the input is made effective and capable of feature extraction. In the feature extraction process, the features are extracted using the VSM and then, the feature selection is done for selecting the highly relevant features to perform text categorization. Once the features are selected, the text categorization is progressed using the deep belief network (DBN). The training of the DBN is performed using the proposed GCOA that is the integration of the grasshopper optimization algorithm (GOA) and Crow search algorithm (CSA). Moreover, the hybrid weight bounding model is devised using the proposed GCOA and range degree. Thus, the proposed GCOA + DBN is used for classifying the text documents.

The major contribution of the research: the major contribution of this work is the development of the GCOA by altering the update equation of the GOA algorithm with the CSA algorithm, to train the DBN. Moreover, the hybrid weight bounding model is devised using the proposed GCOA and range degree.

The paper is structured as follows: Section 1 illustrates the introduction based on text categorization and Section 2 illustrates the literature review of the existing methods of text categorization along with the challenges. The proposed text categorization method is deliberated in Section 3 and the results of the methodologies are elaborated in Section 4 and in, Section 5 the paper is concluded.

2. Literature review

The eight existing literary works employed for the classification are given as follows: Yao *et al.* (2018) developed a method named one-class support vector machine (OCSVM) for incremental learning. This model partitioned the input space into different parts. Then, the classifiers were devised to confound certain parts using support vectors. Throughout the class incremental learning process, the OCSVM of the new class was trained. Then, the support vectors from the old classes and the support vectors of the new class were reused for training 1VS1 classifiers. For devising more information, the support vectors were adapted in OCSVM. The method minimized the memory usage and cost of training time. As the classifier was devised using support vectors, and thus, it becomes complicated to solve the classification problem. Shu *et al.* (2017) devised a deep learning-based method named deep open classification (DOC) for classifying open text. Thus, the text data set was used to show that DOC performed better than other existing methods in terms of text and image classification domains. Moreover, the DOC was effective in dealing with a huge set of images. Furthermore, the DOC builds a multi-class classifier with sigmoids and softmax for reducing the open space risks. In addition, the method was applicable in text categorization and for image classification. However, the method failed to enhance the incremental learning method for learning new classes without any training. Thus, the method was not capable to learn itself to attain lifetime training. Srivastava *et al.* (2019) developed an enrichment protocol that helps to learn different aspects of feature selection like bag-of-words (feature *F0*), latent semantic indexing (feature *F1*) and TF-IDF, (feature *F2*), while applied to the classifier and enhances the overall performance. The method was analyzed with a variety of data sets named disease data with conjunctivitis and WebKB4 data set. The method concluded that it improved the machine learning-accuracy while using the health-care data sets. Moreover, the method was applicable to health care and non-health care data sets with improved accuracy. Sanghani and Kotecha (2019) devised a feature selection function named term frequency difference and category ratio (TFDCR) for learning incremental text classification. The method was devised with three contributions. First, the TFDCR -based feature selection function was devised for selecting the most prominent feature from the set of available features. Second, an incremental model was devised for enabling the classifier to update the dynamic discriminant function. Third, a heuristic function named selection rank weight was devised for upgrading the existing feature set, which finds the new set of features using the incoming data. Different data sets were used for evaluating the performance of filters. The method validated the feasibility and efficiency by enhancing the classification accuracy and minimizing the errors. However, the method failed to use a separate filter for learning to make a unique classification decision. Ranjan and Prasad (2018) developed a connectionist classification approach on the basis of lion fuzzy neural network-based incremental learning algorithm and context-semantic features for incremental learning text classification. The method considered a dynamic database for the classification and learned the model in a dynamic manner. Moreover, the incremental learning process adapted back propagation lion NN in which the lion algorithm and fuzzy bounding were adapted for providing reliable weight selection. Hence, the classifier performed improved classification while new instances were being added without considering old instances and error estimations. Xu *et al.* (2018) developed an algorithm named Markov resampling incremental support vector machines algorithm for illustrating the learning ability of incremental data. The method provided reduced misclassification rates with less computation time based on randomly independent sampling. However, the method faced several complications such as concept drift, multiclass classification problems and regression issues. Park and Kim (2018) developed a network named adaptive resonance theory-supervised predictive mapping for hierarchical classification (ARTMAP-HC) network for allowing incremental class learning using raw data without considering normalization. The method consists of hierarchically stacked modules and each module includes two fuzzy ARTMAP networks. The method was capable to learn incrementally with a huge number of added input data belonging to the new class. Moreover, the method was used for classifying the new data without considering any domain knowledge. The method was applicable in digital storytelling or multimedia recommendation system. Yin and Xi (2017). Designed an entropy model for text categorization in a cloud computing environment and was used for processing a huge number of text documents with enhanced feature selection algorithms. Moreover, the map reduce method was used for text pre-processing with improved feature selection algorithm to choose unique features that helped to improve the precision and recall in optimization algorithms. However, several inadequacies in text pre-processing and execution efficiencies while classifying huge documents and affect the classification result.

3. Proposed hybrid weight bounding model using proposed GCOA-deep belief network

This section illustrates the proposed incremental learning method using a hybrid weight bounding model for categorizing the texts. Initially, the keywords from the documents are fed to the pre-processing phase for eliminating the inconsistent and redundant words from the data using stop word removal and stemming processes. Once the pre-processing is done then, the feature extraction process using VSM (Li *et al.*, 2007) is carried out for extracting the TF-IDF and energy features. In VSM, the documents are represented in vectors and the document is mapped in high-dimensional space. The extracted features are used for selecting the best features for performing the text categorization. For incremental learning, the DBN is used, which poses the weights and biases using a hybrid weight bounding model with the proposed GCOA algorithm to perform effective text categorization. The GCOA is designed by integrating GOA (Łukasik *et al.*, 2017) and CSA (Askarzadeh, 2016). Thus, the incremental text categorization is devised on the basis of the hybrid weight bounding model that includes the GCOA and range degree and particularly, GCOA aims at the selection of the optimal weights for the range degree model. Figure 1 depicts a schematic view of proposed GCOA-based DBN developed for the text categorization.

Assume a Document D containing different attributes and is given as

$$D = \{D_{a,b}\};\ (1 \le a \le A)(1 \le b \le B) \tag{1}$$

where $D_{a,b}$ denote the document present in Database D with the *b-th* attribute in *a-th* data, B is the total number of attributes and A denotes the total number of data points.

3.1 Pre-processing

The pre-processing of the documents is done for removing the redundant words from the text database. The following two main steps in pre-processing are:

(1) Stop word removal; and

(2) Stemming.

The significance of pre-processing is to enable smooth processing of input documents. The text documents are generally huge in size, which contains redundant words and phrases that affect the text categorization process. Hence, it is essential to remove redundant and inconsistent words by using the pre-processing phase.

(i) Stop word removal: the stop words are the words, which are commonly used in the sentence, which include articles, prepositions or pronouns. In computing, the stop words are filtered out before processing the data. The stop word removal is the process of removing the stop words from the huge text documents. Here, the non-information behavior words are eliminated to reduce the noise contained in the data. The removal of stop words can be used to avoid large-space accumulation and enable faster processing to acquire effective results. Here, the stop words such as verbs, nouns are eliminated from the document.

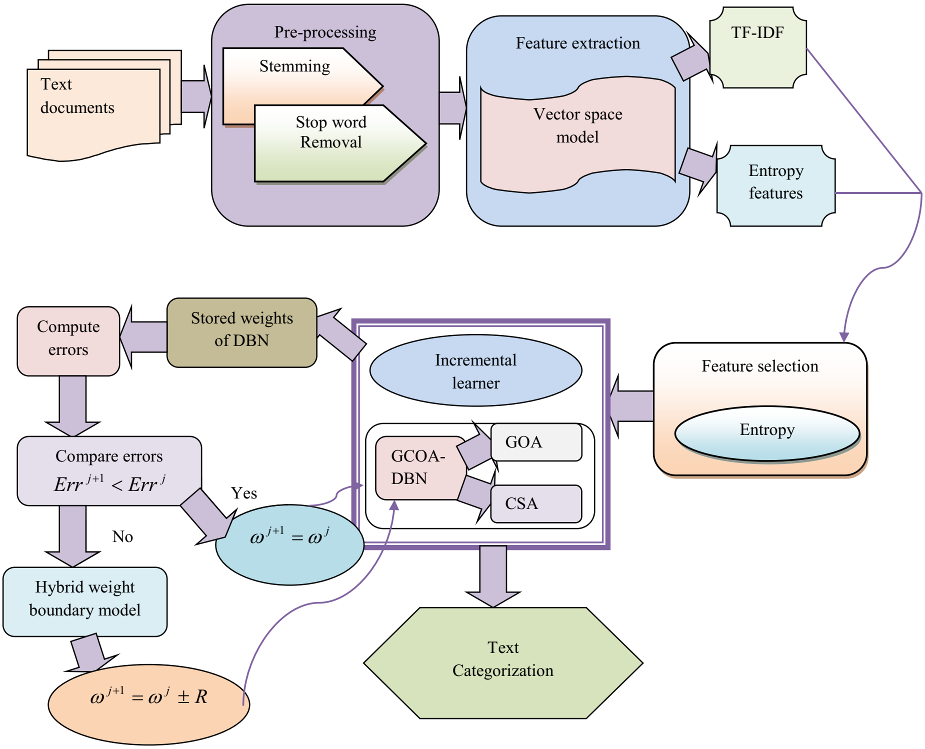


Figure 1.
Proposed incremental text categorization method using the hybrid weight bounding model

(ii) Stemming: the stemming process is used to transform the words into its stem. In large documents, various words are used that convey the same concept. The significant technique used for reducing the words to its root word is called stemming. Numerous words can be processed for reducing the words to their base form. For instance agree, agreeing and disagree belong to the word agree. The stemming is compact, easier to use, and is relatively accurate and moreover, it does not require the suffix list. Here, the stemming technique eliminates the terms that are not relentlessly a meaningful word to its root from the language.

3.2 Extraction of features for text categorization

The section deliberates the significant features extracted from the input document and the significance of feature extraction is to generate the highly relevant features that enable better text categorization using the available documents. On the other hand, the complexity of analyzing the document is minimized as the document is represented as a reduced set of features. Here, the feature extraction is used after pre-processing to extract significant features using a VSM.

3.2.1 Vector space model for information retrieval. The VSM (Li *et al.*, 2007) is the algebraic model used to represent the text documents as vectors. The VSM is applicable for retrieving and filtering the information, indexing and for relevancy rankings. Hence, the VSM is used to extract the entropy-related features and TF-IDF. Here, the feature extraction is performed after pre-processing to extract the features from the documents using TF-IDF.

3.2.1.1 Extraction of term frequency-inverse document frequency features for computing words occurrence. In TF-IDF, the TF is used for computing the occurrence of each word in each document, whereas the inverse document frequency (IDF) is used for computing the important word that occurs rarely in the document. The TF-IDF equation is formulated as follows:

$$KL(PQR) = K(PQ) \times L(PR) \tag{2}$$

where K (PQ) represents the term frequency for computing the occurrence of words in a document and L (PR) specifies the Inverse term frequency for computing important word that occurs rarely in the document, P is the number of words present in the document, R is the total number of documents where ($1 \leq Q \leq R$).

Similarly, the IDF equation is formulated as,

$$L(PR) = \log \frac{R}{1 + \{Q \in R : P \in Q\}} \tag{3}$$

3.3 Feature selection for categorizing texts

After extracting the features, the significant features are selected, in which the entropy model is used to define the rate of the uncertainty of the data points in the document, for selecting fundamental keywords. The feature selection is used for minimizing the dimensionality of the search space. The feature selection is required for categorizing texts, which not only minimizes the index size but also enhances the classifier performance.

3.3.1 Entropy model for selecting relevant features. The feature selection is devised based on document distribution, which contains the terms in categories and uses the documents to compute the entropy (Ranjan and Prasad, 2018). The features are chosen in such a way that it is capable to determine the quality of the feature. Moreover, the entropy is described as the uncertainty measure of a random outcome. Assume $M \times N$ is the dimension of the feature vector. The selected features are arranged in the class of dimension Y. The new class is constructed by matching the selected feature with that in the class. The obtained feature vector is of a new dimension and is given by $M \times (N - Y)$. For set C that has a D number of classes, and thus, the entropy is formulated as,

$$E = -F\log F \tag{4}$$

where F specifies the degree used for mapping elements for classification.

Thus, the features selected are given by T and the feature vector obtained based on the number of documents is given as,

$$y = [A \times j] \tag{5}$$

where A indicates the total number of documents and j refers to the count of unique words.

3.4 Incremental text categorization using proposed GCOA based deep belief network

This section elaborates on the performance of the incremental learning method using the proposed GCOA-based DBN for effective text categorization. When a new query arrives, the proposed GCOA-based DBN determines the equivalent class of the query and updates the weight of DBN. Thus, the proposed incremental learning can perform the classification even if the database is dynamic in nature. The DBN is used for attaining accurate results. The DBN is trained with proposed GCOA for obtaining optimal weights. The proposed GCOA is obtained by combining the GOA and CSA algorithm for attaining effectual text categorization. The CSA is developed from the motivation acquired from the intelligent behavior of the crows in searching for their prey and locating the prey based on the memory. Moreover, the algorithm exhibits a better trade-off between the diversification and the intensification phases effectively and the convergence rate is very high with minimal computational time. The GOA is inspired by the swarming behavior of grasshoppers to solve the optimization issues. The GOA has the capability to obtain the best solution for solving the optimization problems and balances the exploitation and exploration, which helps for obtaining improved results. However, the GOA algorithm exhibits better performance but suffered from a poor convergence rate. Thus, the integration of CSA with GOA resolves the demerits of the GOA algorithm, as CSA possesses a better convergence rate and converges to the global optimal solution. Moreover, a better trade-off between exploration and exploitation is exhibited using CSA that adds the effectiveness to the proposed optimization algorithm. Finally, the proposed GCOA is used for tuning the weights of DBN to acquire accurate results. The DBN provides precise results in solving real-world issues.

3.4.1 Architecture of deep belief network classifier. The basic architecture of the DBN (Hinton *et al.*, 2006) is illustrated in this section using figure 2. The DBN is a part of deep neural network (DNN) and consists of different layers of restricted Boltzmann machines (RBMs) and multilayer perceptrons (MLPs). RBMs contain hidden and visible units, which are linked based on weighted connections. The MLPs are considered as the feed-forward networks that consist of input, hidden and output layers. The network with multiple layers has the ability to solve any complicated tasks and thereby, make the classification of data more effective for determining the incremental text categorization.

The input given to the visible layer is the features obtained by Reuter database and 20 newsgroups database and the hidden layer of the first RBM is expressed as:

$$s^1 = \{s_1^1, s_2^1, \ldots, s_t^1, \ldots, s_l^1\}; 1 \leq t \leq l \tag{6}$$

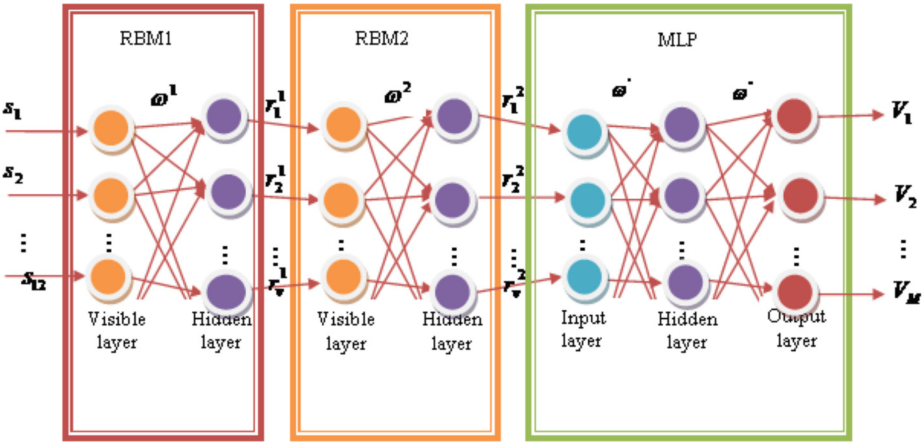


Figure 2.
Architecture of DBN classifier

$$r^1 = \{r_1^1, r_2^1, \ldots, r_u^1, \ldots, r_v^1\}; 1 \leq u \leq v \tag{7}$$

where s_t^1 denote the *t-th* visible neuron in the first RBM, r_u^1 represent the *u-th* hidden neuron and v indicate total hidden neurons. The visible and hidden layers contain neurons, where each neuron poses a bias. Consider x and y represents the biases in the visible layer and hidden layer and these biases for the first RBM layer is formulated as:

$$x^1 = \{x_1^1, x_2^1, \ldots, x_t^1, \ldots, x_{12}^1\} \tag{8}$$

$$y^1 = \{y_1^1, y_2^1, \ldots, y_u^1, \ldots, Y_v^1\} \tag{9}$$

where x_t^1 represents the bias corresponding to t-th visible neurons and y_u^1 indicates the bias corresponding to u-th hidden neurons. The weights used in the first RBM are expressed as:

$$\omega^1 = \{\omega_{t,u}^1\}; 1 \leq t \leq 12; 1 \leq u \leq v \tag{10}$$

where $\omega_{t,u}^1$ denotes the weight between the t-th visible neuron and u-th hidden neuron. Here, the output of the hidden layer from the first RBM is calculated using bias and the weights linked with each visible neuron. This is expressed as:

$$r_u^1 = \alpha\left[y_u^1 + \sum_t s_t^1 \omega_{t,u}^1\right] \tag{11}$$

where α refers to the activation function. Hence, the output obtained in the first RBM can be represented as:

$$r^1 = \{r_u^1\}; 1 \leq u \leq v \tag{12}$$

Then, the learning process of the second RBM layer begins based on the hidden layer output of the first one. The output of the first RBM, given in equation (12) is the input to the visible layer of the second RBM. So, the number of visible neurons here is equivalent to the number of hidden neurons in the first RBM and is expressed as:

$$s^2 = \{s_1^2, s_2^2, \ldots, s_v^2\} = \{r_u^1\}; 1 \leq u \leq v \tag{13}$$

where $\{r_u^1\}$ is the output vector of the first RBM. The hidden layer representation of the second RBM is given by:

$$r^2 = \{r_1^2, r_2^2, \ldots, r_u^2, \ldots, r_v^2\}; 1 \leq u \leq v \tag{14}$$

The biases in the visible layer and the hidden layer have similar representations given in equations (8) and (9) but are denoted as x^2 and y^2, respectively. For the second RBM, the weight vector is represented as:

$$\omega^2 = \{\omega_{uu}^2\}; 1 \leq u \leq v \tag{15}$$

where ω^2_{uu} is the weight between u-th visible neuron and u-th hidden neuron in the second RBM. The output of the u-th hidden neuron is measured similar to the first case as:

$$r^2_u = \alpha\left[y^2_u + \sum_t s^2_t \omega^2_{uu}\right] \forall s^2_t = r^1_u \tag{16}$$

where y^2_u is the bias associated with the u-th hidden neuron. Thus, the hidden layer output obtained is given by:

$$r^2 = \{r^2_u\}; 1 \leq u \leq v \tag{17}$$

The above equation forms the input to the MLP, where the number of neurons in the input layer is v. The input layer of MLP is represented as:

$$p = \{p_1, p_2, \ldots, p_u, \ldots, p_v\} = \{r^2_u\}; 1 \leq u \leq v \tag{18}$$

where v is the number of neurons in the input layer, which is provided by the hidden layer output of the second RBM $\{r^2_u\}$. The hidden layer of the MLP is given as:

$$n = \{n_1, n_2, \ldots, n_o, \ldots, n_M\}; 1 \leq O \leq M \tag{19}$$

where M is the total number of hidden neurons. Assume n_o as the bias of o-th hidden neurons. The third layer, which is the output layer, of the MLP is represented as,

$$V = \{V_1, V_2, \ldots, V_o, \ldots, V_w\}; 1 \leq o \leq w \tag{20}$$

where w is the number of neurons in the output layer. MLP has two weight vectors, one between the input layer and the hidden layer, and the other between the hidden layer and the output layer. Let ω' be the weight vector between the input and the hidden layers, as given below:

$$\omega^{'} = \{\omega^{'}{}_{uo}\}; 1 \leq u \leq v; 1 \leq o \leq M \tag{21}$$

where $\omega^{'}{}_{uo}$ is the weight between u-th input neuron and o-th hidden neuron. Based on the weights in the neurons together with the bias, the hidden layer output is calculated as:

$$n_o = \left[\sum_{u=1}^{v} \omega^{''}{}_{uO})^* P_u\right] U_o \forall P_u = r^2_u \tag{22}$$

where U_o is the bias of hidden neurons and $p_u = r^2_u$, as the input to the MLP is the output of the second RBM. The weights between the hidden layer and the output layer are denoted as ω'' and are given by:

$$\omega^{''} = \{\omega^{''}{}_{Oo}\}; 1 \leq O \leq M; 1 \leq o \leq w \tag{23}$$

Thus, the output vector can be computed based on the weight ω'' and the hidden layer output, as formulated below,

$$V_o = \sum_{O=1}^{M} \omega''_{oO} * n_o \tag{24}$$

where ω''_{oO} is the weight between the O-th hidden neuron and o-th output neuron and n_o is the output of the hidden layer.

3.4.2 Training of deep belief network using the GCOA algorithm. In this section, the training of DBN using the proposed GCOA algorithm is elaborated. The goal of proposed GCOA-based DBN is to categorize the texts and classify the massive text documents into predefined categories based on the features extracted from the input data. The proposed technique offers conceptual views of document sets and has many applications in the real world. The training of DBN is performed using the GCOA algorithm, which is generated by incorporating GOA in CSA. The GCOA algorithm inherits the advantages of both GOA (Łukasik *et al.*, 2017) and CSA (Askarzadeh, 2016) and provides the best performance for incremental text categorization. The CSA algorithm is a metaheuristic algorithm devised on the basis of the intelligent behavior of crows. The algorithm is used for controlling the diversity of the algorithm. The CSA is easier to implement and provides solutions with improved accuracy. The demerits of CSA are that it possesses lower convergence and it is highly sensitive to the hyperparameters. The demerits of CSA are overcome using GOA that offers a better convergence rate while obtaining a globally optimal solution. GOA is duly based on the behavior of grasshopper swarms. It is worth notable that the search for the prey is both associated within or outer the search spaces through a series of steps such as encircling, exploitation and exploration. GOA is capable of resolving the real problems with unknown search spaces. The steps involved in the training algorithm are discussed below.

3.4.2.1 Initialization. In the initial step, the weights of the DBN are initialized in a random manner and is represented as follows,

$$X = \{X_1, X_2, \ldots, X_g, \ldots, X_\alpha\}; 1 < g \leq \alpha \tag{25}$$

where α indicates total weights.

3.4.2.2 Error estimation. Apply the weight X and the selected features T to the DBN to find the output. The output error is the sum of the squares of the current output of the network and the training label output for training the network, given as,

$$Err^{e+1} = \frac{1}{D}\sum_{z=1}^{D}\left[O_z^e - Z_z^e\right] \tag{26}$$

where D is the total number of data samples, O_z^e is the estimated output at current iteration and Z_z^e is the predicted output.

3.4.2.3 Weight bound based on incremental learning. Whenever a new instance T^{e+1} is added to the network, the error Err^{e+1} is computed and the weights are updated, which is the goal of the incremental learning algorithm. If the error computed is smaller than that evaluated error for the previous instance, then the weight allocated to the network is generated using equation (38). On the other hand, a hybrid weight bound model is used, that bounds the weights and chooses the suitable one using proposed GCOA. Thus, the updated weights are computed by taking the difference between stored weights and range degree (Scholkopf *et al.*, 1997) and is given by:

$$X(e+1) = X(e) \pm R \tag{27}$$

where $X(e)$ denotes the stored weights and R is the range degree.

The range degree is given by:

$$R = \sqrt{\frac{\upsilon\left(\log\frac{2\ell}{\upsilon}+1\right) - \log\left(\frac{\mu}{4}\right)}{\ell}} \tag{28}$$

where υ denote Vapnik–Chervonenkis dimension of a set of functions and is used to describe the capacity of a set of functions implemented by a learning machine, ℓ is the training samples and μ denote the random number between [0, 1].

3.4.2.4 Weight update using proposed GCOA. The updated weight is determined based on the GCOA algorithm and is derived on the basis of the following equation.

According to GOA (Łukasik *et al.*, 2017):

$$X_h^i = d\left(\sum_{\substack{k=1\\k\neq h}}^{H} d\frac{J_i - O_i}{2} p\left(q_k^i - q_h^i\right)\frac{q_k - q_h}{c_{h,k}}\right) + \hat{U}_i \tag{29}$$

where d is the decreasing coefficient, J_i denote upper bound in i-th dimension, O_i represents lower bound in i-th dimension, p defines the social forces, q_k^i denote the position of k-th grasshopper in i-th dimension, q_h^i denote the position of h-th grasshopper in i-th dimension, $C_{h,k}$ indicates the distance between h-th grasshopper and k-th grasshopper and the tendency to move forward is given using a term denoted as $\hat{U}_i$.

The weight update based on CSA (Askarzadeh, 2016) is based on the probability of searching for the food and is given by:

$$X_h(e+1) = X_h(e) + t_h \times S_h(e) \times (o_k(e) - X_h(e)) \tag{30}$$

where $X_h(e)$ denotes the crow's position in the current iteration e, and the random number is denoted as t_h, the flight length of h-th dimension at the current iteration is given by $S_h\,(e)$ and memory of the crow is given by $o_k(e)$.

After rearranging, the above equation is represented as:

$$X_h(e+1) = X_h(e) + t_h \times S_h(e) \times o_k(e) - t_h \times S_h(e) \times X_h(e) \tag{31}$$

$$t_h \times S_h(e) \times o_k(e) = -X_h(e) + t_h \times S_h(e) \times X_h(e) + X_h(e+1) \tag{32}$$

$$o_k(e) = \frac{1}{t_h \times S_h(e)}\left[X_h(e+1) + X_h(e)\left[t_h \times S_h(e) - 1\right]\right] \tag{33}$$

After substituting equation (33) in equation (30) by assuming $\hat{U}_i = o_k(e)$, the obtained equation becomes:

$$X_h^i = d\left(\sum_{k\neq h}^{k=1}{}^{H} d\frac{J_i - O_i}{2} p\left(q_k^i - q_h^i\right)\frac{q_k - q_h}{c_{h,k}}\right) + \frac{1}{t_h \times S_h(e)}\left[X_h(e+1) + X_h(e)\left[t_h \times S_h(e) - 1\right]\right] \tag{34}$$

$$X_h(e+1) - \frac{-X_h(e+1)}{t_h \times S_h(e)} = d\left(\sum_{k\neq h}^{k=1}{}^{H} d\frac{J_i - O_i}{2} p\left(q_k^i - q_h^i\right)\frac{q_k - q_h}{c_{h,k}}\right) + \frac{X_h^i}{t_h \times S_h(e)}\left[t_h \times S_h(e) - 1\right] \tag{35}$$

$$X_h(e+1)\left[1 - \frac{1}{t_h \times S_h(e)}\right] = d\left(\sum_{k\neq h}^{k=1}{}^{H} d\frac{J_i - O_i}{2} p\left(q_k^i - q_h^i\right)\frac{q_k - q_h}{c_{h,k}}\right) + \frac{X_h^i}{t_h \times S_h(e)}\left[t_h \times S_h(e) - 1\right] \tag{36}$$

$$X_h(e+1)\left[\frac{t_h \times S_h(e) - 1}{t_h \times S_h(e)}\right] = d\left(\sum_{k\neq h}^{k=1}{}^{H} d\frac{J_i - O_i}{2} p\left(q_k^i - q_h^i\right)\frac{q_k - q_h}{c_{h,k}}\right) + \frac{X_h^i}{t_h \times S_h(e)}\left[t_h \times S_h(e) - 1\right] \tag{37}$$

The weights are updated using the proposed GCOA and are evaluated in such a way that the weights corresponding to the minimum value of error are used for training DBN as per the equation below:

$$X_h(e+1) = \frac{t_h \times S_h(e)}{t_h \times S_h(e) - 1} = d\left(\sum_{k\neq h}^{k=1}{}^{H} d\frac{J_i - O_i}{2} p\left(q_k^i - q_h^i\right)\frac{q_k - q_h}{c_{h,k}}\right) + \frac{X_h^i}{t_h \times S_h(e)}\left[t_h \times S_h(e) - 1\right] \tag{38}$$

Equation (38) is used for the selection of optimal weights. The minimal value of the error describes the better weight, and therefore, the solution with the minimum value of the error is chosen as the best weight.

3.4.2.5 Determination of feasible weights. Finally, the weights are updated using equation (38), which is obtained by the proposed GCOA algorithm.

3.4.2.6 Stopping criterion. The optimal weights are derived in an iterative manner until the maximum iteration limit is achieved.

4. Results and discussion

The analysis of results using proposed GCOA + DBN and existing methods based on recall, precision and accuracy is illustrated.

4.1 Experimental setup

The testing of the methods is performed in the personal computer with 2 GB RAM, Intel i-3 core processor, windows 10 operating system using JAVA.

4.2 Database description

The data set used for performing the text categorization includes 20 newsgroups database and Reuter database, which are illustrated below.

4.2.1. 20 newsgroups database. The 20 newsgroups data set (20 Newsgroup database, 2018) is donated by Ken Lang for the newsreader to refine the netnews. The data set is formed by accumulating 20,000 newsgroup documents, which is divided evenly across 20 different newsgroups. The data set is well known for experimentation of the text applications to deal with machine learning methods such as text clustering and text classification. The data set consists of 19,997 articles, which are arranged in 20 different newsgroups, each representing different topics.

4.2.2 Reuter database. The Reuters-21578 text categorization collection data set (Reuter database, 2018) is donated by David D. Lewis, which comprises of documents that appeared on Reuters newswires in 1987. The documents are organized and indexed on the basis of categories. The number of instances of the data set is 21,578 with five attributes. The number of web hits achieved by the data set is 163,417.

4.3 Evaluation metrics

The analysis of the methods is carried out based on three metrics, namely, precision, recall and accuracy.

4.3.1 Precision. Precision is defined by the nearness of more than two measurements to each other and is difficult from that of accuracy.

$$\mathrm{Pre}cision = \frac{t_p}{t_p + f_p} \tag{39}$$

where the term t_p denotes the true positive and f_p represents the false positive.

4.3.2 Recall. The recall is defined by computing the total number of actual positives that the system captures with the label of it as the true positive.

$$\mathrm{Rec}all = \frac{t_p}{t_p + f_n} \tag{40}$$

where f_n is the false negative.

4.3.3 Accuracy. The accuracy denotes the measure of the closeness of the GCOA + DBN approach for text categorization and is expressed as,

$$Accuracy = \frac{t_p + t_n}{t_p + t_n + f_p + f_n} \tag{41}$$

4.4 Comparative analysis

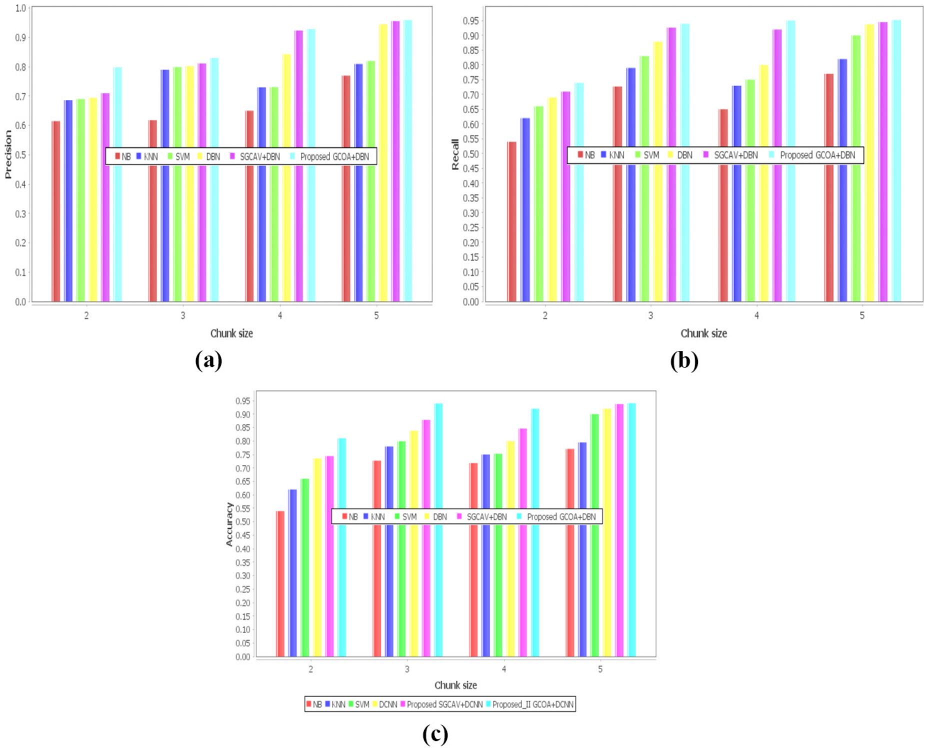
The analysis of the proposed GCOA + DBN and existing methods is done using two databases, namely, the Reuter database and 20 newsgroups database.

4.5 Competing methods

The comparative methods, includes NB (Scholkopf *et al.*, 1997), k-nearest neighbors (KNN) (Toker and Kirmemis, 2013), SVM (Joachims, 1998) and DBN (Hinton *et al.*, 2006), Stochastic Gradient-CAViaR (SGCAV) + DBN and proposed GCOA + DBN, which are used for the evaluation.

4.5.1 Comparative analysis using 20 newsgroup database

4.5.1.1 For entropy = 100. Figure 3 illustrates the analysis of methods based on accuracy, precision and recall parameter using entropy 100. The analysis of methods based on the precision parameter is portrayed in Figure 3(a). When the chunk size is 2, the corresponding precision values computed by existing NB, KNN, SVM, DBN, SGCAV + DBN, and proposed GCOA + DBN are 0.614, 0.686, 0.69, 0.694, 0.71 and 0.797, respectively. Likewise, when the chunk size is 5, the corresponding precision values computed by existing NB, KNN, SVM, DBN, SGCAV + DBN, and proposed GCOA + DBN are 0.77, 0.809, 0.82, 0.945, 0.955 and 0.959, respectively. The analysis of methods based on recall measure is portrayed in Figure 3(b). When the chunk size is 2, the corresponding recall values computed by existing NB, KNN, SVM, DBN, SGCAV + DBN, and proposed GCOA + DBN are 0.54, 0.62, 0.66, 0.69, 0.71 and 0.739, respectively. Likewise, when the chunk size is 5, the corresponding recall



(a)

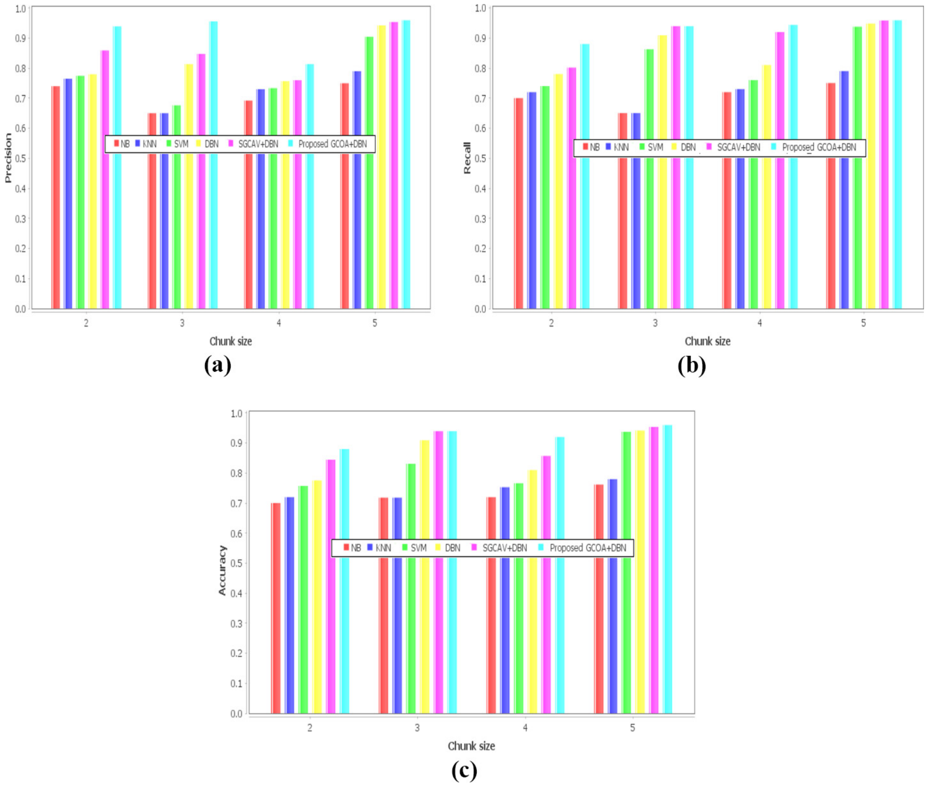
(b)

(c)

Figure 3.
Analysis of methods based on entropy 100 using (a) precision (b) recall (c) accuracy

values computed by existing NB, KNN, SVM, DBN, SGCAV + DBN, and proposed GCOA + DBN are 0.77, 0.82, 0.9, 0.937, 0.945 and 0.952, respectively. The analysis of methods using an accuracy metric is portrayed in Figure 3(c). When the chunk size is 2, the corresponding accuracy values computed by existing NB, KNN, SVM, DBN, SGCAV + DBN, and proposed GCOA + DBN are 0.54, 0.62, 0.66, 0.735, 0.744 and 0.810, respectively. Likewise, when the chunk size is 5, the corresponding accuracy values computed by existing NB, KNN, SVM, DBN, SGCAV + DBN and proposed GCOA + DBN are 0.770, 0.794, 0.9, 0.92, 0.937 and 0.940, respectively.

4.5.1.2 For entropy = 200. Figure 4 illustrates the analysis of methods based on accuracy, precision and recall parameter using entropy 200. The analysis of methods based on the precision measure is portrayed in Figure 4(a). When the chunk size is 2, the corresponding precision values computed by existing NB, KNN, SVM, DBN, SGCAV + DBN, and proposed GCOA + DBN are 0.74, 0.764, 0.774, 0.78, 0.859 and 0.939, respectively. Likewise, when the chunk size is 5, the corresponding precision values computed by existing NB, KNN, SVM, DBN, SGCAV + DBN, and proposed GCOA + DBN are 0.75, 0.79, 0.904, 0.942, 0.953 and 0.959, respectively. The analysis of methods based on recall measure is portrayed in Figure 4(b). When the chunk size is 2, the corresponding recall values computed by existing NB, KNN, SVM, DBN, SGCAV + DBN, and proposed GCOA + DBN are 0.7, 0.72, 0.74, 0.78, 0.801 and 0.88, respectively. Likewise, when the chunk size is 5, the corresponding recall values computed by existing NB, KNN, SVM, DBN, SGCAV + DBN,



(a)

(b)

(c)

Figure 4. Analysis of methods based on entropy 200 using (a) precision (b) recall (c) accuracy

and proposed GCOA + DBN are 0.75, 0.79, 0.937, 0.948, 0.958 and 0.959, respectively. The analysis of methods using accuracy measure is portrayed in Figure 4(c). When the chunk size is 2, the corresponding accuracy values computed by existing NB, KNN, SVM, DBN, SGCAV + DBN, and proposed GCOA + DBN are 0.7, 0.72, 0.757, 0.775, 0.844 and 0.88, respectively. Likewise, when the chunk size is 5, the corresponding accuracy values computed by existing NB, KNN, SVM, DBN, SGCAV + DBN, and proposed GCOA + DBN are 0.761, 0.779, 0.937, 0.941, 0.953 and 0.96, respectively.

4.5.2 Comparative analysis using the Reuter database

4.5.2.1 For entropy = 100. Figure 5 illustrates the analysis of methods based on accuracy, precision and recall parameter using entropy 100 with the Reuter database. The analysis of methods based on the precision measure is portrayed in Figure 5(a). When the chunk size is 2, the corresponding precision values computed by existing NB, KNN, SVM, DBN, SGCAV + DBN, and proposed GCOA + DBN are 0.400, 0.595, 0.696, 0.697, 0.727 and 0.763, respectively. Likewise, when the chunk size is 5, the corresponding precision values computed by existing NB, KNN, SVM, DBN, SGCAV + DBN and proposed GCOA + DBN are 0.605, 0.608, 0.717, 0.742, 0.745 and 0.772, respectively. The analysis of methods based on recall measure is portrayed in Figure 5(b). When the chunk size is 2, the corresponding recall values computed by existing NB, KNN, SVM, DBN, SGCAV + DBN and proposed GCOA + DBN are 0.540, 0.636, 0.636, 0.727, 0.763 and 0.947, respectively. Likewise, when the chunk size is 5, the corresponding recall values computed by existing NB, KNN, SVM, DBN, SGCAV + DBN and proposed GCOA + DBN are 0.590, 0.590, 0.636, 0.745, 0.772 and 0.946,

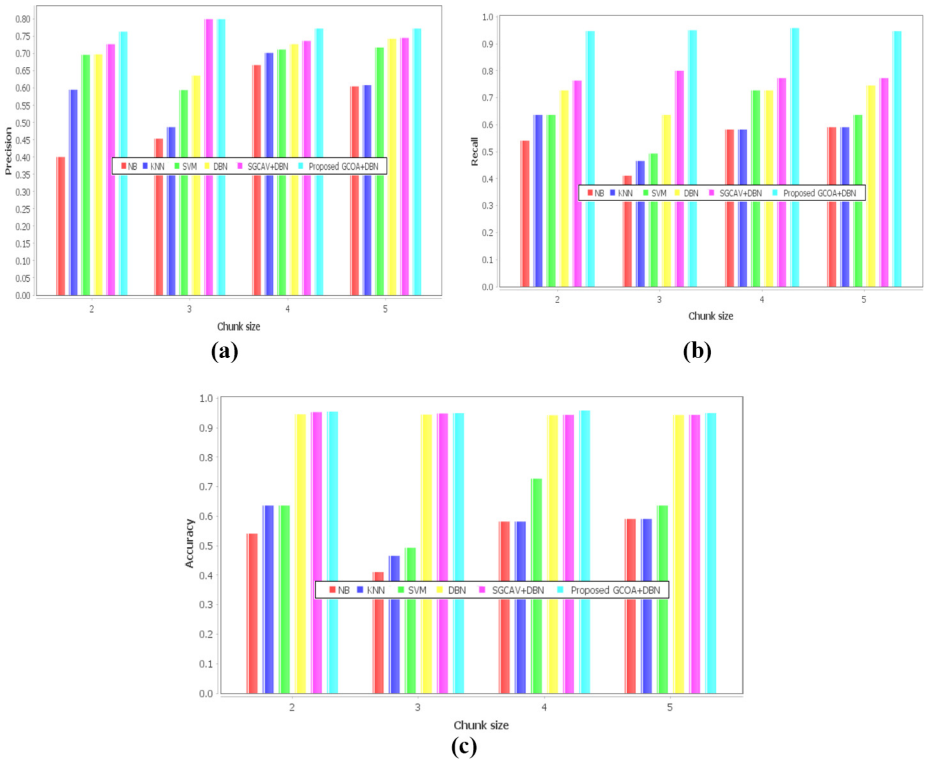
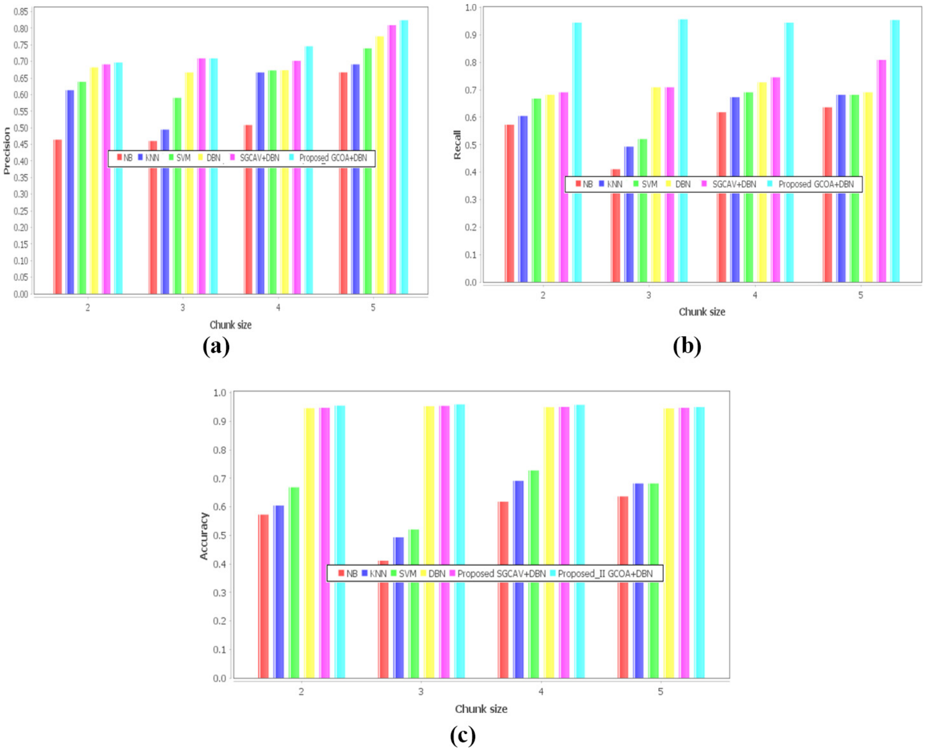


Figure 5.
Analysis of methods based on entropy 100 using (a) precision (b) recall (c) accuracy

respectively. The analysis of methods based on accuracy measures is portrayed in Figure 5 (c). When the chunk size is 2, the corresponding accuracy values computed by existing NB, KNN, SVM, DBN, SGCAV + DBN and proposed GCOA + DBN are 0.540, 0.636, 0.636, 0.945, 0.953 and 0.955, respectively. Likewise, when the chunk size is 5, the corresponding accuracy values computed by existing NB, KNN, SVM, DBN, SGCAV + DBN and proposed GCOA + DBN are 0.590, 0.590, 0.636, 0.943, 0.943 and 0.949, respectively.

4.5.2.2 For entropy = 200. Figure 6 illustrates the analysis of methods based on accuracy, precision and recall parameter using entropy 200 with the Reuter database. The analysis of methods based on the precision measure is portrayed in Figure 6(a). When the size of chunk is 2, the corresponding precision values computed by existing NB, KNN, SVM, DBN, SGCAV + DBN, and proposed GCOA + DBN are 0.464, 0.613, 0.638, 0.681, 0.690 and 0.696, respectively. Likewise, when the chunk size is 5, the corresponding precision values computed by existing NB, KNN, SVM, DBN, SGCAV + DBN and proposed GCOA + DBN are 0.667, 0.690, 0.739, 0.775, 0.809 and 0.823, respectively. The analysis of methods based on recall measure is portrayed in Figure 6(b). When the chunk size is 2, the corresponding recall values computed by existing NB, KNN, SVM, DBN, SGCAV + DBN and proposed GCOA + DBN are 0.572, 0.604, 0.668, 0.681, 0.690 and 0.944, respectively. Likewise, when the chunk size is 5, the corresponding recall values computed by existing NB, KNN, SVM, DBN, SGCAV + DBN, and proposed GCOA + DBN are 0.636, 0.681, 0.681, 0.690, 0.809 and 0.954, respectively. The analysis of methods based on accuracy measures is portrayed in Figure 6 (c). When the chunk size is 2, the corresponding accuracy values computed by existing NB,



(a)

(b)

(c)

Figure 6.
Analysis of methods based on entropy 200 using (a) precision (b) recall (c) accuracy

KNN, SVM, DBN, SGCAV + DBN, and proposed GCOA + DBN are 0.572, 0.604, 0.668, 0.945, 0.947 and 0.954, respectively. Likewise, when the chunk size is 5, the corresponding accuracy values computed by existing NB, KNN, SVM, DBN, SGCAV + DBN, and proposed GCOA + DBN are 0.636, 0.681, 0.681, 0.945, 0.947 and 0.949, respectively.

4.6 Comparative discussion

Table 1 elaborates the analysis of comparative methods using two databases, namely, 20 newsgroup databases and Reuter databases with respect to the accuracy, precision and recall parameter. The maximal precision is acquired by the proposed GCOA + DBN with an accuracy value of 0.959 whereas the precision values acquired by existing NB, KNN, SVM, and DBN, are 0.75, 0.79, 0.904, 0.942 and 0.953. The maximal recall is attained by proposed GCOA + DBN with a value of 0.959 whereas the recall values of existing NB, KNN, SVM, DBN, and SGCAV + DBN are 0.75, 0.79, 0.937, 0.948 and 0.958, respectively. The maximal accuracy is attained by proposed GCOA+DBN with a value of 0.96 whereas the accuracy values of existing NB, KNN, SVM, DBN, SGCAV + DBN, are 0.761, 0.779, 0.937, 0.941 and 0.953, respectively. The analysis reveals that the proposed GCOA + DBN outperformed other existing methods with maximal precision of 0.959, maximal recall of 0.959 and maximal accuracy of 0.96, respectively.

Table 2 shows the computational time of the proposed GCOA + DBN, and the existing methods such as NB, KNN SVM, DBN SGCAV + DBN, in which the proposed GCOA + DBN has the minimum computation time of 6.14 s.

5. Conclusion

This paper proposes a technique that categorizes the texts from massive sized documents. Initially, the input documents are pre-processed using the stop word removal and stemming technique such that the input is made effective and capable of the feature extraction process. In the feature extraction process, the features are extracted using the VSM, and then, the feature selection is done for selecting the highly relevant features to perform text categorization. Once the features are selected, the text categorization is progressed using the DBN. The training of the DBN is performed using the proposed GCOA that is the integration of the GOA and CSA. Moreover, the hybrid weight bounding model is devised using the proposed GCOA and range degree. Thus, the proposed GCOA-based DBN is used for classifying the text documents. The proposed GCOA + DBN outperformed other existing methods with maximal precision of 0.959, maximal recall of

Database	Metric	NB	KNN	SVM	DBN	SGCAV + DBN	Proposed GCOA + DBN
Using 20 newsgroup database	Precision	0.75	0.79	0.904	0.942	0.953	0.959
	Recall	0.75	0.79	0.937	0.948	0.958	0.959
	Accuracy	0.761	0.779	0.937	0.941	0.953	0.96
Using reuter database	Precision	0.667	0.690	0.739	0.775	0.809	0.823
	Recall	0.636	0.681	0.681	0.690	0.809	0.954
	Accuracy	0.636	0.681	0.681	0.945	0.947	0.949

Table 1. Comparative analysis

Methods	NB	KNN	SVM	DBN	SGCAV + DBN	Proposed GCOA + DBN
Time (sec)	12.03	11	10.4	8.96	7.28	6.14

Table 2. Computational time

0.959 and maximal accuracy of 0.96, respectively. The proposed text categorization approach is used in various applications such as spam email filtering, document organization and news groupings. The future extension is to adapt an advanced text categorization process for implementing electronic-mail classification or in the web page classification.

References

Askarzadeh, A. (2016), "A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm", *Computers and Structures*, Vol. 169, pp. 1-12.

Berge, G.T., Granmo, O.-C., Oddbjørn Tveit, T., Goodwin, M., Jiao, L. and Matheussen, B.V. (2019), "Using the Tsetlin machine to learn human-interpretable rules for high-accuracy text categorization with medical applications", *IEEE Access*, Vol. 7, pp. 115134-115146.

Cai, L. and Hofmann, T. (2004), "Hierarchical document classification with support vector machines", In *Proceedings of the thirteenth ACM international conference on information and knowledge management*, pp. 78-87.

Camastra, F. and Razi, G. (2019), "Italian text categorization with lemmatization and support vector machines", *Neural Approaches to Dynamics of Signal Exchanges*, Vol. 151, pp. 47-54.

Charikar, M., Chekuri, C., Feder, T. and Motwani, R. (2004), "Incremental clustering and dynamic information retrieval", *SIAM Journal on Computing*, Vol. 33 No. 6, pp. 1417-1440.

Chen, H., Hou, Y., Luo, Q., Hu, Z. and Yan, L. (2018), "Text feature selection based on water wave optimization algorithm", in *proceedings of Tenth International Conference on Advanced Computational Intelligence (ICACI)*, pp. 546-551.

George, A. and Rajakumar, B.R. (2013), "On hybridizing fuzzy min max neural network and firefly algorithm for automated heart disease diagnosis", in *the proceeding of Fourth International Conference on Computing, Communications and Networking Technologies, Tiruchengode, India, July*.

Ghuge, C.A., Chandra Prakash, V. and Ruikar, S.D. (2019), "Weighed query-specific distance and hybrid NARX neural network for video object retrieval", *The Computer Journal*,

Hinton, G.E., Osindero, S. and Teh, Y. (2006), "A fast learning algorithm for deep belief nets", *Neural Computation*, Vol. 18 No. 7, pp. 1527-1554.

Jo, T. (2019), "Improving K nearest neighbor into string vector version for text categorization", In *the proceeding of 21st International Conference on Advanced Communication Technology (ICACT), PyeongChang Kwangwoon_Do, Korea (South)*.

Joachims, T. (1998), "Text categorization with support vector machines: learning with many relevant features", In *proceedings of European conference on machine learning*, *Springer*, pp. 137-142.

Kim, K. and Zzang, S.Y. (2018), "Trigonometric comparison measure: a feature selection method for text categorization", *Data and Knowledge Engineering*, Vol. 119.

Kim, S.B., Han, K.S., Rim, H.C. and Myaeng, S.H. (2006), "Some effective techniques for navive Bayes text classification", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 18 No. 11.

Labani, M., Moradi, P., Ahmadizar, F. and Jalili, M. (2018), "A novel multivariate filter method for feature selection in text classification problems", *Engineering Applications of Artificial Intelligence*, Vol. 70, pp. 25-37.

Lee, J., Yu, I., Park, J. and Kim, D.W. (2019), "Memetic feature selection for multilabel text categorization using label frequency difference", *Information Information Sciences*, Vol. 485.

Li, H., Ma, B. and Lee, C.H. (2007), "A vector space modeling approach to spoken language identification", *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 15 No. 1, pp. 271-284.

Łukasik, S., Kowalski, P.A., Charytanowicz, M. and Kulczycki, P. (2017), "Data clustering with grasshopper optimization algorithm", *2017 Federated Conference on Computer Science and Information Systems (FedCSIS), Prague*, pp. 71-74.

Ma, T., Motta, G. and Liu, K. (2017), "Delivering real-time information services on public transit: a framework", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 18 No. 10, pp. 2642-2656.

Mohammad, A.H., Alwada'n, T. and Al-Momani, O. (2018), "Arabic text categorization using support vector machine", *GSTF Journal on Computing (Joc))*, Vol. 5no No. 1.

Newsgroup database (2018), http://qwone.com/~jason/20Newsgroups/, accessed on October.

Ninu Preetha, N.S. and Praveena, S. (2018), "Multiple feature sets and SVM classifier for the detection of diabetic retinopathy using retinal images", *Multimedia Research (MR)*, Vol. 1 No. 1, pp. 17-26.

Park, J.-Y. and Kim, J.-H. (2018), "Incremental class learning for hierarchical classification", *IEEE Transactions on Cybernetics*.

Ranjan, N.M. and Prasad, R.S. (2018), "LFNN: Lion fuzzy neural network-based evolutionary model for text classification using context and sense based features", *Applied Soft Computing*, Vol. 71.

Reuter database (2018), https://archive.ics.uci.edu/ml/datasets/reuters-8+text+categorization+collection accessed on October 2018.

Sanghani, G. and Kotecha, K. (2019), "Incremental personalized E-mail spam filter using novel TFDCR feature selection with dynamic feature update", *Expert Systems with Applications*, Vol. 115, pp. 287-299.

Scholkopf, B., Sung, K.K., Burges, C.J., Girosi, F., Niyogi, P., Poggio, T. and Vapnik, V. (1997), "Comparing support vector machines with gaussian kernels to radial basis function classifiers", *IEEE Transactions on Signal Processing*, Vol. 45 No. 11, pp. 2758-2765.

Shu, L., Xu, H. and Liu, B. (2017), "Doc: Deep open classification of text documents", *arXiv Preprint arXiv:1709.08716*.

Song, S., Xiaofei, Q. and Chen, P. (2009), "Hierarchical text classification incremental learning", In *proceedings of International Conference on Neural Information Processing ICONIP, Neural Information Processing*, pp. 247-258.

Srivastava, S.K., Singh, S.K. and Suri, J.S. (2019), "Effect of incremental feature enrichment on healthcare text classification system: a machine learning paradigm", *Computer Methods and Programs in Biomedicine*, Vol. 172, pp. 35-51.

Sudhakar, R.V., Mruthyunjayam, A., Suguna Kuamari, D., Ravi Kumar, M. and Ramesh Babu, B.V.S. (2013), *Improving Login Authorization by Providing Graphical Password (Security)*, Vol. 3 No. 6, pp. 484-489.

Taeho, J. (2019), "K nearest neighbor for text categorization using feature similarity", *ICAEIC-2019*, Vol. 2 No. 1, p. 99.

Tang, B., Kay, S. and He, H. (2016), "Toward optimal feature selection in naive Bayes for text categorization", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 28 No. 9, pp. 2508-2521.

Tellez, E.S., Moctezuma, D., Miranda-Jiménez, S. and Graff, M. (2018), "An automated text categorization framework based on hyperparameter optimization", *Knowledge-Based Systems*, Vol. 149, pp. 110-123.

Toker, G. and Kirmemis, O. (2013), *Text Categorization Using k Nearest Neighbor Classification*, Technical University.

Wang, D. and Al-Rubaie, A. (2015), "Incremental learning with partial-supervision based on hierarchical Dirichlet process and the application for document classification", *Applied Soft Computing*, Vol. 33, pp. 250-262.

Wang, N., Fu, J., Bhargava, B.K. and Zeng, J. (2018), "Efficient retrieval over documents encrypted by attributes in cloud computing", *IEEE Transactions on Information Forensics and Security*, Vol. 13 No. 10, pp. 2653-2667.

Xu, J., Xu, C., Zou, B., Tang, Y.Y., Peng, J. and You, X. (2018), "New incremental learning algorithm with support vector machines", *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 49 No. 11, pp. 1-12.

Yang, Y. and Pedersen, J.O. (1997), "A comparative study on feature selection in text categorization", In *proceedings of International Conference on Machine Learning*, pp. 412-420.

Yao, C., Zou, J., Luo, Y., Li, T. and Bai, G. (2018), "A class-incremental learning method based on one class support vector machine", *arXiv Preprint arXiv:1803.00159*.

Yin, C. and Xi, J. (2017), "Maximum entropy model for mobile text classification in cloud computing using improved information gain algorithm", *Multimedia Tools and Applications*, Vol. 76 No. 16, pp. 16875-16891.

Corresponding author

V. Srilakshmi can be contacted at: srilakshmiv064@gmail.com