Evolutionary Intelligence
https://doi.org/10.1007/s12065-020-00449-x

RESEARCH PAPER

Stochastic gradient-CAViaR-based deep belief network for text categorization

V. Srilakshmi[1] · K. Anuradha[2] · C. Shoba Bindu[1]

Received: 25 November 2019 / Revised: 24 March 2020 / Accepted: 2 July 2020

Abstract
Text categorization is defined as the process of assigning tags to text according to its content. Some of the text classification approaches are document organization, spam email filtering, and news groupings. This paper introduces stochastic gradient-CAViaR-based deep belief networks for text categorization. The overall procedure of the proposed approach involves four steps, such as pre-processing, feature extraction, feature selection, and text categorization. At first, the pre-processing is carried out from the input data based on stemming, stop-word removal, and then, the feature extraction is performed using a vector space model. Once the extraction is done, the feature selection is carried out based on entropy. Subsequently, the selected features are given to the text categorization step. Here, the text categorization is done using the proposed SG-CAV-based deep belief networks (SG-CAV-based DBN). The proposed SG-CAV is used to train the DBN, which is designed by combining conditional autoregressive value at risk and stochastic gradient descent. The performance of the proposed SGCAV + DBN is evaluated based on the metrics, such as recall, precision, F-measure and accuracy. Also, the performance of the proposed method is compared with the existing methods, such as Naive Bayes, K-nearest neighbours, support vector machine, and deep belief network (DBN). From the analysis, it is depicted that the proposed SGCAV + DBN method achieves the maximal precision of 0.78, the maximal recall of 0.78, maximal F-measure of 0.78, and the maximal accuracy of 0.95. Among the existing methods, DBN achieves the maximum precision, recall, F-measure and accuracy, for 20 Newsgroup database and Reuter database. The performance of the proposed system is 10.98%, 11.54%, 11.538%, and 18.33% higher than the precision, recall, F-measure, and accuracy of the DBN for 20 Newsgroup database, and 2.38%, 2.38%, 2.37%, and 0.21% higher than the precision, recall, F-measure and accuracy of the DBN for Reuter database.

Keywords Text categorization · Deep belief network · Stochastic gradient descent · CAViaR · Vector space model

1 Introduction

One of the most important processes in the knowledge discovery process is text mining [1]. As most of the information on the internet is unstructured, hence it is appropriate to regulate the data process text mining-based algorithms. Several intelligent algorithms, like neural networks, case-based reasoning and probability reasoning and combination of text processing technology, have been proven effective in the text mining process. The text mining algorithms regulate the unstructured document and help in extracting the key concept and the relationship among the characters. The text mining algorithms help in text classification by accessing useful knowledge and information from the database [2]. The documents present online have improved, due to the growth of the internet. Regularly text documents contain research articles, blogs, journal papers, and newspapers, and so on. This vast number of documents may be valuable and useful [3, 4]. Documents are generally denoted by "bag-of-words", such as every phrase or word present in documents once or several times took as the feature. For the given data set, a collection of entire phrases or words forms the "dictionary" with several features. To ameliorate the "curse of dimensionality" problem and to boost up the learning process, it is necessary for performing feature reduction to mitigate the feature size [5].

TC is utilized for finding relevant categories from the given text. It is the core method for several applications,

✉ V. Srilakshmi
srilakshmiv064@gmail.com

[1] CSE, JNTUA, Anantapur, India

[2] CSE, GRIET, Hyderabad, India

Published online: 12 July 2020

like review recommendation, spam detection, clinical analysis, and sentiment analysis. A single text can be assigned to numerous categories, and therefore, task categorization requires multilabel classification [6]. TC is a branch of text mining, where the classification scheme is constructed to define a set of logical rules for classifying the documents from the given set of categories. After that, the classification is performed for assigning the perfect class. TC is classified into multi-label and single labels. One class contains a single label document, whereas multiple classes are contained in multi-label documents [7, 8]. TC in data mining provides better information from large data. TC plays a very significant role in classifying the huge electronic documents efficiently in various sources [9]. TC is used in several applications, like news monitoring, email filtering, scientific research [10] and searching for interesting information on the web [11]. Few methods are utilized for text categorization, namely k-nearest neighbour (KNN) [12], Naive Bays, support vector machine (SVM) [13, 14], and decision tree [15, 16]. A new method SVMCNN by combining convolutional neural networks and support vector machine [8], short-term memory recurrent neural network (LSTM-RNN), data collection and preparation, feature extraction and classification using LSTM [7], self-paced learning (ASPL) [17], MNB, decision tree, random forest [18] are used for text categorization. An incremental text classifier using Kullback–Leibler distance (KLD) [19–21] is performed for detecting public transit issues and events from online social media. Naive Bayes [8, 22] based text classification ensures improved performance in incremental learning.

However, it is difficult to extract information from the unstructured textual resource, for classifying the document to a set of predefined categories [17]. Also, the construction of features for the new user is difficult [9, 23]. Even though the solution obtained from categorization is proved to be efficient, and simple, the estimation of the parameter in the classifier is complex. To overcome these drawbacks is the main objective of the proposed SGCAV + DBN.

In this research paper, a text categorization technique is developed using SGCAV + DBN. The overall procedure of the proposed text categorization involves the following four steps, such as pre-processing, feature extraction, feature selection, and text categorization. At first, the documents are pre-processed based on stop-word removal and stemming techniques, followed by the feature extraction carried out using the vector space model. Depending on the feature extracted, feature selection is performed using entropy. At last, the text categorization is performed based on the selected features using the proposed SGCAV + DBN, which is modified using SGD and CAViaR.

The main contribution of the research paper is developing a text categorization approach using the proposed SGCAV + DBN in which the DBN is trained using SGCAV for effective text classification.

The paper is structured in the following manner: Sect. 1 provides the introductory part of the text categorization and Sect. 2 discusses existing methods of text categorization with challenges of the methods that remain the motivation for the research. The proposed method of SGCAV + DBN is demonstrated in Sect. 3, and Sect. 4 describes the results of the methods. At last, Sect. 5 concludes the research work.

2 Literature survey

The review of the existing methods is given as follows: Tang et al. [24, 25] developed five-way joint mutual information (FJMI) for mitigating the computational complexity. Five-dimensional joint mutual information was utilized for computing the interaction terms, and then the nonlinear approach was introduced for avoiding overestimation problems, but this method needs Bayesian networks and adversarial networks for improving the selection of features. In this method, the accuracy based on a benchmark dataset is 80%. Tellez et al. [2] developed a minimalist and global approach for the categorization of text. This approach was employed to find the competitive text classifier from the set of candidates'. Here, the text classifier was determined by the parameters to determine the functionality of classifiers. The actual accuracy of a different kind of pre-processing was 0.8265, 0.8340, 0.8310, 0.8373, and 0.8413, for raw, all-terms, no-short, no-stopwords, and stemmed, respectively.

Liu et al. [26] developed selective multiple instance transfer learning (SMITL) for text categorization. SMITL was utilized for transferring the files safely from a source task to the target task, but needs the perfect solution to boost up the multiple transfer instances, and also failed to reduce the cost. The maximum accuracy produced by this method was 0.796. Kim and Zzang [27] developed trigonometric comparison measure (TCM) by considering relative document frequencies. TCM achieved better performance for text classification, but limited sensitive to parameter selection than normalized difference measure (NDM). This method was produced the highest F1 values for SVM and NB classifier, when the number of features is larger than 100. Anyhow, this method did not provide better results for 10 or 20 features. Feng et al. [28] modelled supervised weighting technique using a probabilistic model for text categorization. This method contains the latent term selection indicator for addressing the non-discriminating term weighting. The developed method is fully Bayesian, and the prior information is introduced into the weighting scheme. The accuracy of this method was 80% for 20 newsgroup datasets, and 83.9% for Reuter dataset. Yang et al. [29] developed a modified convolutional neural network that is

Table 1 Related terminologies

Terms	Definition
Text mining	It defines the process of extracting non-trivial patterns from a huge data pool
TC	It is a task of dividing the unlabeled electronic documents automatically, such as advertisements, call records e-mails, news articles, and so on
Precision	Precision is defined by the nearness of more than two measurements to each other, and is difficult from that of accuracy
Recall	Recall is defined by computing the total number of actual positives that the system captures with the label of it as the true positive
Accuracy	Accuracy denotes the measure of the closeness of the SGCAV + DBN approach for text categorization
F-measure	F-measure is a measure of the test's accuracy by considering both the recall and the precision

based on the dropout and the stochastic gradient (SGD) optimizer. This method improves the feature recognition rate and reduces the time costs of CNN. The analysis of this method had been done using the metrics, such as recognition rate, time, and cost. The average recognition rate of this method was 84.93%. Dai et al. [30] developed a transfer-learning algorithm for text classification, which was very effective in several different pairs of domains. However, this method did not produce better results in theoretical measures. The classification accuracy of this method was 83.8%. Camastra and Razi [31] developed an Italian language text categorizer by SVMs and Lemmatization. This was used for the large dataset text categorization. Anyhow, this categorization was not considering the synonyms of the words. The accuracy of this method was 92.92%, 84.68%, 83.93%, 79.53%, 77.36%, 76.51%, 75.80%, 75.65%, and 81.62% for the datasets sport (SP), motors (M), entertainment and culture (EC), science (S), news section (NC), business (B), politics (P), foreign (F), and all test set, respectively. Lee et al. [6] developed a feature selection method for text categorization, which used feature wrapper for the improvement in memetic search capability, but it did not support the multilabel text feature selection method. This method had 6.8725, 6.6732, 17.0896, and 17.5298 for multi-label accuracy, subset accuracy, hamming loss, and ranking loss, respectively. Jo [32] developed a string vector-based version of the KNN for text categorization, which was applicable for the larger set of databases. However, this method was considered less number of features for the performance analysis. In this method, the accuracy was a range between 0.49 and 1.0. Berge et al. [33] developed a Tsetlin machine for the text categorization, which had the capacity to produce human-interpretable rules with accuracy. This method was not effective to address the complex nonlinear patterns. This method was had a precision of 69.9%, recall of 72.8% and F-measure of 76.9%.

2.1 Research gaps

Existing text categorization techniques face the following challenges:

- The challenges faced by text classification are learning from the high dimensional data. To ignore the problem of "curse of dimensionality" and to boost up the learning procedure, it is required for performing feature reduction to mitigate the size of the feature [7].
- The text classification techniques are useful for various applications, since they are computationally different. In these cases, various features have been reduced [20], which may change the system performance of classification.
- In machine learning, text classification faces various issues, like a huge amount of text categories, high dimensionality feature space, and training data that are hard for handling [19].
- Another challenge faced by text classification is the construction of features. The extracted features in classification are efficient so that it may be applied to the range of class definitions, but it is complex for creating new features for every user [9].
- One of the challenges in text classification is hard to capture high-level semantics and natural languages through simple words. This is because the words have semantic uncertainties, such as synonymy and hyponymy [19].

To take these research gaps as a motivation, and overcome these challenges a novel approach SG-CAV + DBN is proposed.

3 Related terminologies

The terms related to this research are given in Table 1.

4 Text categorization using proposed SGCAV-based DBN

This section presents the proposed SGCAV-based DBN for text categorization. At first, the keywords from the documents are given to pre-processing to remove redundant and unnecessary words from the data using stop word removal

and stemming. After pre-processing, the feature extraction is done using a vector space model to find the keywords of the document. Then, the feature selection is performed using entropy, and finally, the text categorization is performed using SGCAV-based DBN, which is employed for training the DBN. The proposed SG-CAV is developed by combining the stochastic gradient algorithm with CAViaR [34]. Figure 1 illustrates the schematic diagram of the proposed SGCAV-based DBN developed for the text categorization.

4.1 Pre-processing

The first step involved in the document clustering is the pre-processing of the text. The input database contains unnecessary words or phrases, which may affect the clustering process. Consider D be the set of documents and it comprises of n number of documents in the database and is denoted as $D = \{d_i;\ 1 \le i \le n\}$. Therefore, the pre-processing is considered for removing the redundant words from the text database. The two main steps in pre-processing are: (1) stop word removal, and (2) stemming.

Stop word removal: The stop word is the unnecessary words available in the text document, like an, a, the, in, etc.

Stemming: In this step, the stemming technique converts the terms that are not inevitably a meaningful word to its root from the language. The number of documents in the *i*th database is expressed as,

$$d_i = \left\{ w_j^i,\ 1 \le j \le m_i \right\} \tag{1}$$

where the term m_i denotes the extracted words from the *i*th documents. After extracting the keywords, W unique keywords is obtained as,

$$W = \left\{ b_l, 1 \le x \le k \right\} \tag{2}$$

where k represents the total number of words in the dictionary or unique keywords from the documents. Thus, the dictionary words are generated from the pre-processing step, and then the feature is extracted from the dictionary words.

4.2 Feature extraction based on vector space model

Feature extraction is performed after pre-processing by extracting the keywords from documents using the vector space model. The vector space model indicates the text documents as the vectors of identifiers. It is employed

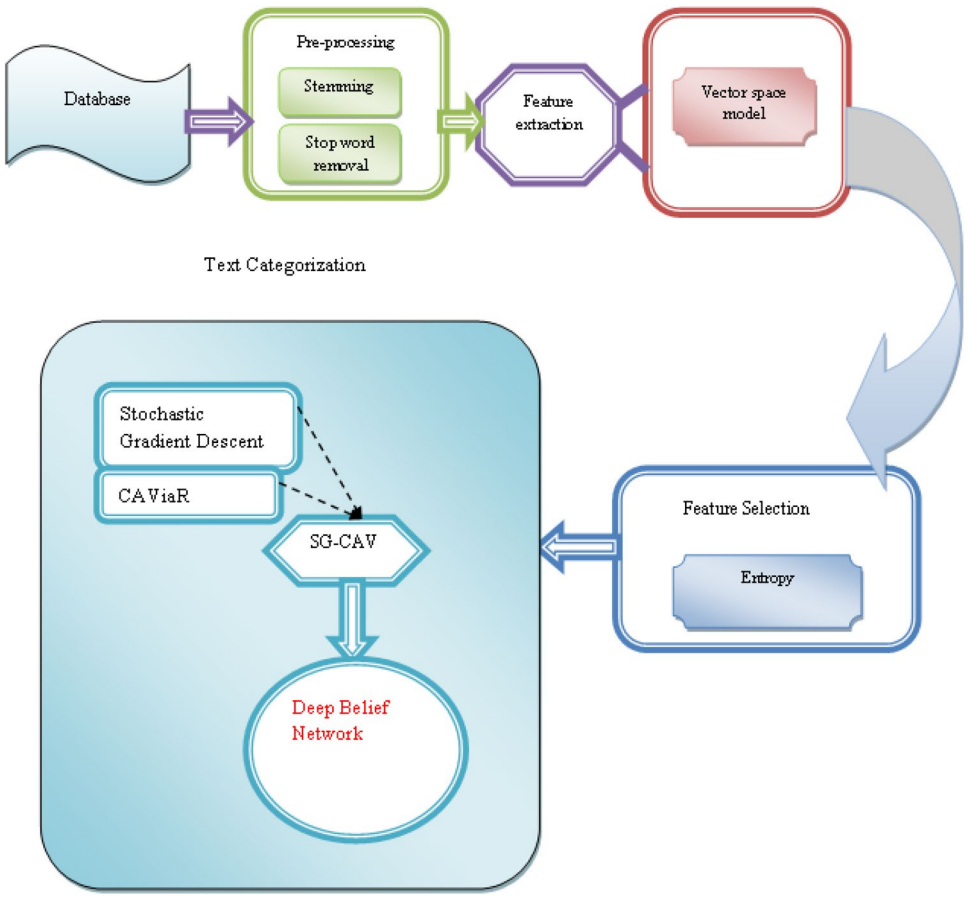

Fig. 1 Proposed SG-CAV-based DBN algorithm for text categorization

in information retrieval, indexing information filtering, and relevancy rankings. Here, the score functions are based on term-frequency (TF) and inverse-document-frequency (IDF). The TF-IDF features are considered as the weighting factor for data mining, and the text categorization based applications. TF is utilized to compute the occurrence of each word in each document. IDF is used for calculating important word that occurs rarely in the document.

$$Q(b_l, D) = \log \frac{n}{\left|\{d \in D : b_l \in d\}\right|} \tag{3}$$

$Q(b_l, D)$ be the IDF of b_lth word in database D, b_l represents the words in the document, d refers to each document, and n indicates the collection of documents. Therefore, the extracted features are represented as,

$$V = \{U_{ij},\ 1 \le i \le n; 1 \le j \le 2k\} \tag{4}$$

where U_{ij} represents the jth keyword in ith document, and n signifies the total number of documents, $2k$ refer to the total number of extracted features using TF, and IDF.

4.3 Feature selection

After extraction, the feature selection is performed based on entropy function [35] to mitigate the time by reducing the dimension of the search space for classification. The entropy function is duly based on the distribution of documents with the term in documents and considers its entropy. The selected features determine the quality of the feature. Entropy is defined by measuring the uncertainty of the random outcome. Let us consider $B \times X$ be the dimension of the feature database. The selected keywords are then structured in a class of dimension S. Then, the matching is done to create a new database. Therefore, the resultant database is considered to have the new dimension that is reduced by q, that is $B \times (X - S)$. The entropy function is expressed as,

$$E_j = -\sum_{i=1}^{n} P_{ij} \log P_{ij} \tag{5}$$

where the term P_{ij} refers to the ith symbol of the jth feature varying from 1 to $2k$. The term n denotes the number of symbols in the jth feature. After finding entropy of every jth feature, the top M features are selected based on minimum values. After selecting the features, the feature selected database can be indicated as,

$$V^{red} = \{U_{ij}; 1 \le i \le n, 1 \le j \le M\};\ \ M < 2k \tag{6}$$

where M denotes the features.

4.4 Proposed SGCAV-based DBN for text categorization

This section elaborates on the text categorization using the proposed SGCAV-based DBN. The selected features are presented for the classification using DBN and the training of the classifier is done through the proposed training algorithm, called SG-CAV, which is the modification of the gradient descent algorithm with the CAViaR [36]. The main aim of the proposed SG-CAV is to perform the classification effectively. The architecture of DBN and the algorithmic steps of the proposed ST-CAV are described below.

4.4.1 Architecture of the DBN

The DBN [37] is the part of deep neural network (DNN) and comprises of various layers of multilayer perceptrons (MLPs), and restricted Boltzmann machines (RBMs). RBMs consists of visible and hidden units, which are connected based on weighted connections. The MLPs are considered as the feed-forward networks that contain input layers, hidden layers, and output layers. The network with the multiple layers has the ability to resolve any complicated tasks and thereby, make the text categorization more effective (Fig. 2).

The input given to the visible layer is the features obtained using entropy function and the hidden layer of the first RBM is expressed as,

$$U^1 = \left\{U_1^1, U_2^1, \ldots, U_g^1, \ldots, U_p^1\right\};\quad 1 \le g \le p \tag{7}$$

$$G^1 = \left\{G_1^1, G_2^1, \ldots, G_i^1, \ldots, G_s^1\right\};\quad 1 \le i \le s \tag{8}$$

where the term U_g^1 represents the gth visible neuron in RBM 1, and the term G_i^1 signifies the ith hidden neuron and the total hidden neuron is denoted as s. The hidden and visible layers contain neurons in which every neuron poses the bias. Let X and Y be the biases in the hidden and visible layer and these biases for RBM 1 layer is given as,

$$\mathrm{X}^1 = \left\{\mathrm{X}_1^1, \mathrm{X}_2^1, \ldots, \mathrm{X}_g^1, \ldots, \mathrm{X}_p^1\right\} \tag{9}$$

$$Y^1 = \left\{Y_1^1, Y_2^1, \ldots, Y_i^1, \ldots, Y_s^1\right\} \tag{10}$$

where the bias related to gth visible neuron is denoted as X_g^1, and the term Y_i^1 refers to the bias related to ith hidden neurons. For the first RBM, the weight vector is expressed as,

$$\varpi^1 = \left\{\varpi_{g,i}^1\right\};\quad 1 \le g \le p;\quad 1 \le i \le s \tag{11}$$

where the weight between gth visible neuron and ith hidden neuron is denoted as $\varpi_{g,i}^1$. In this case, the output of hidden

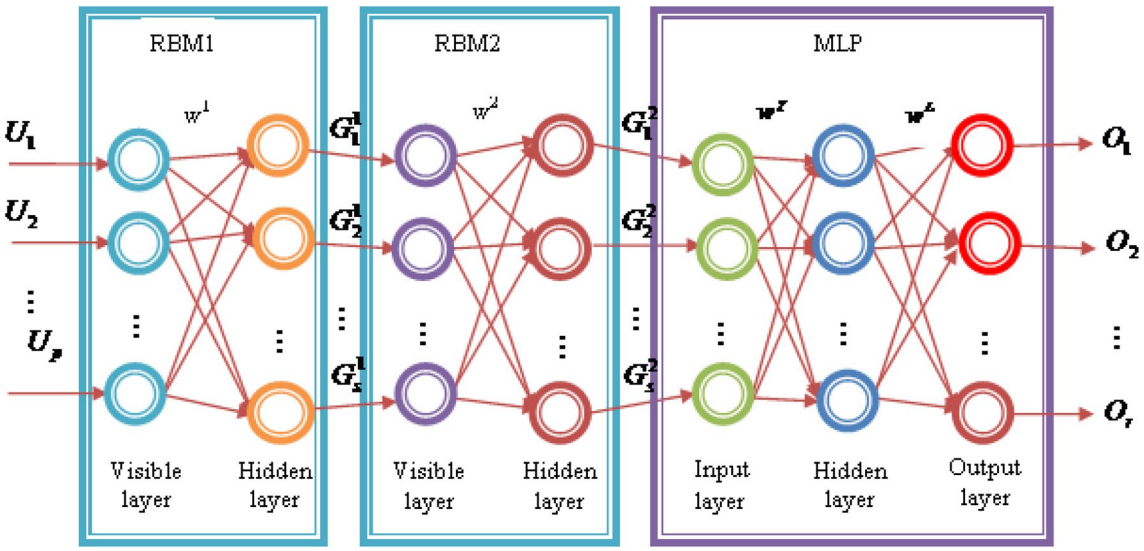

Fig. 2 Architecture of DBN classifier

layer from RBM 1 is computed based on weights and bias connected with each visible neuron, and is given as,

$$R_i^1 = \gamma\left[\Upsilon_i^1 + \sum_{\varpi}(U_g^i)^1 \varpi_{g,i}^1\right] \tag{12}$$

where the activation function is denoted as γ. Thus, the output generated from the first RBM is given by,

$$R^1 = \left\{R_i^1\right\}; \quad 1 \le i \le s \tag{13}$$

Here, the output of RBM 1 is forwarded as the input to the visible layer of RBM 2 as input. Hence, the input of the RBM 2 layer is indicated as U^2. Similarly, the hidden layer of RBM 2 is represented by G^2. The bias in the hidden layer and visible layer is indicated with X^2 and Υ^2. The RBM 2 weight vector is given as ϖ^2 the output of the ith hidden neuron is represented as G_i^2, and Υ_i^2 is the bias linked with the ith hidden neuron. Therefore, the output obtained from the hidden layer is given by G^2.

The output obtained from the hidden layers of RBM 2 is fed to MLP as an input, with the input layer. The input layer of MLP is given as,

$$H = \left\{H_1, H_2, \ldots, H_i, \ldots, H_o\right\} = \left\{R_i^2\right\}; \quad 1 \le i \le s \tag{14}$$

where, the total neuron available in the input layer is denoted as i, which is given by the output of the hidden layer of RBM 2 $\{R_i^2\}$. The hidden layer of MLP is given below,

$$C = \left\{C_1, C_2, \ldots, C_T, \ldots, C_{\mathrm{A}}\right\}; \quad 1 \le T \le \mathrm{A} \tag{15}$$

where the total hidden neurons is denoted as A. Let us consider a_T represents the bias of T hidden neurons, where $T = 1, 2, \ldots, \mathrm{A}$. The output layer of MLP is formulated as,

$$\mathrm{Z} = \left\{\mathrm{Z}_1, \mathrm{Z}_2, \ldots, \mathrm{Z}_{\mathrm{N}}, \ldots, \mathrm{Z}_{\mathrm{B}}\right\}; \quad 1 \le \mathrm{N} \le \mathrm{B} \tag{16}$$

where the total neurons available in the output layer are denoted as B. Here, the MLP considers two weight vectors; one is present between the hidden and input layer, and the other between output and hidden layer. Assume ϖ^{α} denotes the weight vector between input and hidden layers and is represented as,

$$\varpi^{\alpha} = \left\{\varpi_{iT}^{\beta}\right\}; \quad 1 \le i \le s; \quad 1 \le T \le \mathrm{A} \tag{17}$$

where the term ϖ_{iT}^{β} indicates the weight between ith neuron and Tth hidden neuron. The output of the hidden layer is computed by,

$$C_i = \left[\sum_{i=1}^{s} \varpi_{i,T}^{\alpha} * C_T\right] a_T \tag{18}$$

where C_T signifies the Tth input layer of MLP. The weights between output and hidden layer are given by ϖ^{ρ} and are represented as,

$$\varpi^{\rho} = \left\{\omega_{T\mathrm{N}}^{\rho}\right\}; \quad 1 \le T \le \mathrm{A}; \quad 1 \le \mathrm{N} \le \mathrm{B} \tag{19}$$

Therefore, the output vector is computed using the weights ϖ^{ρ} and output of the hidden layer and is expressed as,

$$O_N = \sum_{T=1}^{A} \varpi_{TN}^{\rho} * C_i \tag{20}$$

where the term ϖ_{TN}^{ρ} represents the weight between the Tth hidden neuron and Nth output neuron and C_i denotes hidden layer output.

4.4.2 Training of DBN based on SG-CAV

This section elaborates on the training procedure of the proposed SG-CAV-based DBN classifier. Here, the DBN [37] that is composed of RBM layers and MLP layers is trained using a stochastic gradient descent method. The training process of MLP is based on SG-CAV algorithm by giving the training data, which is the output of the hidden layer of the second RBM layer, over the network. Examining the data, the network is accustomed repeatedly until the optimal weights are chosen. Moreover, SG-CAV is employed for computing the optimal weights, which are evaluated using an error function. The integration of CAViaR in the SGD algorithm [29, 38] inherits the advantages of both CAViaR [34, 36] and the SGD algorithm. The stochastic algorithm is highly efficient as it is linear for the number of training data, and it is capable of approximating the true gradient for every single training data over time. The demerits of SGD are that it possesses lower convergence and it is highly sensitive to the hyperparameters. The demerits of SGD are overcome using CAViaR that offers a better convergence rate while obtaining an optimal solution. The update equation of SGD is obtained as follows:

$$W_{z+1} = W_z - \eta \frac{\partial}{\partial w}(w_u) \tag{21}$$

Here, the update equation is modified based on the update rule of CAViaR. CAViaR is a semi-parametric approach based on simple intuition and failed to require any assumption based on time series distribution. The VaR is computed using quantile regression, which allows the time series to change from one stochastic process to another. CAViaR is based on simple intuition that is best to model VaR directly from quantile. The CAViaR equation is expressed as,

$$W_{z+1} = \gamma_0 + \gamma_1 W_z + \gamma_2 W_{z-1} + \gamma_1 f(W_z) + \gamma_2 f(W_{z-1}) \tag{22}$$

$$W_z = \frac{1}{\gamma_1}\left[W_{z+1} - \gamma_0 - \gamma_2 W_{z-1} - \gamma_1 f(W_z) - \gamma_2 f(W_{z-1})\right] \tag{23}$$

Substituting Eq. (23) in Eq. (21),

$$W_{z+1} = \frac{1}{\gamma_1}\left[W_{z+1} - \gamma_0 - \gamma_2 W_{z-1} - \gamma_1 f(W_z) - \gamma_2 f(W_{z-1})\right] - \eta \frac{\partial}{\partial w}(w_u) \tag{24}$$

$$W_{z+1} = \frac{1}{\gamma_1} W_{z+1} + \frac{1}{\gamma_1}\left[-\gamma_0 - \gamma_2 W_{z-1} - \gamma_1 f(W_z) - \gamma_2 (W_{z-1})\right] - \eta \frac{\partial}{\partial w}(w_u) \tag{25}$$

$$W_{z+1}\left[1 - \frac{1}{\gamma_1}\right] = \frac{1}{\gamma_1}\left[-\gamma_0 - \gamma_2 W_{z-1} - \gamma_1 f(W_z) - \gamma_2 f(W_{z-1})\right] - \eta \frac{\partial}{\partial w}(w_u) \tag{26}$$

$$W_{z+1} = \frac{1}{(\gamma_1 - 1)}\left[-\gamma_0 - \gamma_2 W_{z-1} - \gamma_1 f(W_z) - \gamma_2 f(W_{z-1})\right] - \eta \frac{\partial}{\partial w}(w_u) \tag{27}$$

Equation (25) is the update equation of CAViaR for finding the most appropriate position of the search agent, where $f(.)$ represents fitness.

The steps involved in the proposed SG-CAV are deliberated in the following steps.

1. *Initialization:* The first step is the initialization of the feature weights that are given as,

$$\varpi = \varpi_1^o = \varpi_2^o = \cdots = \varpi_q^o = \frac{1}{\sqrt{q}} \tag{28}$$

where q represents the features and ϖ_q^e signifies the oth attribute weights corresponding to the qth feature.

2. *Error estimation:* The fitness of the solutions is evaluated for individual iteration, to determine the best solution so that the position update of the search agents in the iteration follows the best solution. The fitness is evaluated based on the minimum value of the error, and the solution corresponding to the minimum error is evaluated as the best solution. The error is determined as,

$$MSE = \frac{1}{n}\left[\sum_{h=1}^{n} R_{target} - N_J^K\right]^2 \tag{29}$$

where N_J^K and R_{target} are the estimated and target output of the classifier.

3. *Weight update using proposed SG-CAV:* The weights updated using the algorithms SG and CAV are evaluated in such a way that the weights corresponding to the minimum value of error are employed for training DBN as per the equation below,

$$W_{z+1} = \begin{cases} W_{z+1}^{SG}; if\ e_{avg}^{SG} < e_{avg}^{CAV} \\ W_{z+1}^{CAV}; Otherwise \end{cases} \tag{30}$$

4. *Determination of feasible weights:* In this step, the weights for training DBN are determined individu-

ally using SG and CAV and the update is based on the weights that contribute towards the minimum value of the error.

5. *Terminate:* The steps are repeated until the iteration reaches the maximal count. After reaching the maximum iteration, the best solution is attained, and it is considered as the optimal weight.

The algorithm for the proposed SG-CAV-based DBN is given below.

Algorithm 1: Algorithm for the proposed SG-CAV-based DBN

1	Input: feature weight ϖ
2	Output: best optimal weight
3	Initialization $\varpi = \varpi_1^o = \varpi_2^o = \cdots = \varpi_q^o = \frac{1}{\sqrt{q}}$
3	Evaluate the fitness based on minimum value of the error (Eq. 29)
4	While (min. error < max. no. of iterations)
5	Update weight by Eq. (30)
6	Determine the feasible weight
7	Determine the best optimal weight
8	z = z + 1
9	End While
10	Return best optimal weight

The flowchart of the proposed SGCAV + DBN is given in Fig. 3.

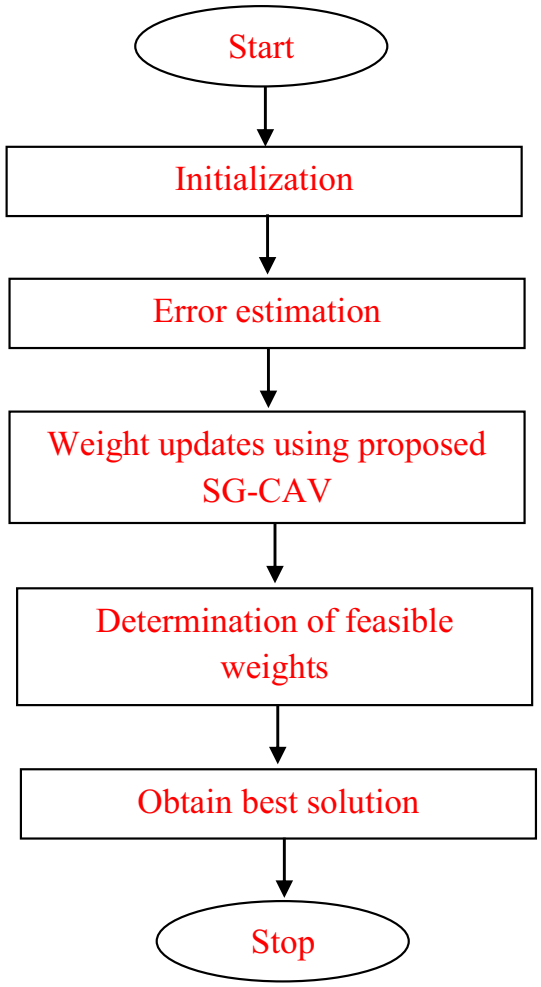

Fig. 3 Flowchart for the proposed SG-CAV + DBN

5 Discussion of results

The results obtained by the proposed SGCAV + DBN are described in this section. The performance of the proposed SGCAV + DBN is analyzed using three measures, which include precision, recall, F-measure and accuracy.

5.1 Experimental setup

The experimentation of the proposed technique of text categorization is performed in the system with 2 GB RAM, Intel i-3 core processor, Windows 10 Operating System. The proposed method is executed in JAVA.

5.2 Database description

The dataset is taken from the Reuter database and 20 Newsgroups database for text categorization.

5.2.1 Newsgroups database

The 20 Newsgroups dataset [39] is collected nearly about 20,000 documents that are partitioned from 20 groups. Ken Lang gathered this database where each of the new groups represents a topic. It is a benchmark corpus typically used in the research field of text categorization (or text clustering). This corpus consists of 19,997 articles that are organized into 20 different categories, and it is highly balanced since each category has nearly 1000 texts.

5.2.2 Reuter database

This Reuter dataset [40] consists of documents that are related to the newswire stories. The documents are divided into PEOPLES, TOPICS, ORGS, PLACES, and EXCHANGES. It was originally gathered by Carnegie Group, Inc. and Reuters, Ltd. There are several versions, such as Apte' split 90 categories and Apte' split 115 categories. The version of Apte' split 90 categories which

contain 11,406 texts for 90 categories. The instance distribution over the 90 categories is highly imbalanced.

5.3 Performance metrics

The evaluation of the proposed technique is performed based on three metrics, namely precision, recall, F-measure, and accuracy.

Precision: The formula for precision is,

$$Precision = \frac{t_p}{t_p + f_p} \tag{31}$$

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

Recall: The formula for recall is,

$$Recall = \frac{t_p}{t_p + f_n} \tag{32}$$

where f_n be the false negative.

Accuracy: The accuracy is expressed as,

$$Accuracy = \frac{t_p + t_n}{t_p + t_n + f_p + f_n} \tag{33}$$

F-measure: The formula for F-measure is given as,

$$F-measure = 2\left[\frac{Precision \cdot Recall}{Precision + Recall}\right] \tag{34}$$

5.4 Comparative analysis

The comparative analysis of the developed SGCAV + DBN by evaluating the performance of other comparative techniques is elaborated in this section. The comparative analysis is performed by varying the training data percentage, and the results are evaluated based on precision, recall, F-measure, and accuracy.

5.5 Competing methods

The methods, such as Naive Bayes NB [30], K-nearest neighbours (KNN) [41], support vector machine (SVM) [42], and DBN [37], are utilized for the comparison with the proposed SGCAV + DBN for the analysis.

5.5.1 Comparative analysis using 20 Newsgroup database

5.5.1.1 For entropy = 100 The analysis of the comparative methods based on precision, recall, F-measure, and accuracy for entropy = 100 is depicted in Fig. 4. Figure 4a shows the analysis in terms of precision by varying the percentage of training data. For the training data = 50%, the existing techniques, like NB, KNN, SVM, and DBN, possess the precision of 0.5369, 0.550, 0.575, and 0.66, respectively, which is comparatively lower than the SGCAV + DBN. For the same training data, the developed SGCAV + DBN acquired the precision of 0.75. Similarly, when the training percentage increased to 90%, the methods, NB, KNN, SVM, and DBN, attained the precision of 0.450, 0.4868, 0.6015, and 0.61, whereas the precision of the developed method is 0.76. From the above interpretation, it is seen that the proposed SGCAV + DBN achieved improved precision of 0.78 at 80% training data.

The analysis in terms of recall metric is depicted in Fig. 4b. When 60% of training data is considered, the methods, like NB, KNN, SVM, and DBN acquires the recall value of 0.49, 0.56, 0.6475, 0.67, respectively. Meanwhile, the proposed SGCAV + DBN obtained the recall value of 0.74. When 80% of training data is considered, the recall of the methods, like NB, KNN, SVM, and DBN is 0.43, 0.56, 0.63, and 0.69, whereas the proposed SGCAV + DBN achieved the recall of 0.78.

The analysis in terms of F-measure metric is depicted in Fig. 4c. When 60% of training data is considered, the methods, such as NB, KNN, SVM, and DBN acquires the F-measure value of 0.49, 0.523, 0.577, 0.67, respectively. Meanwhile, the proposed SGCAV + DBN obtained the F-measure value of 0.74. When 80% of training data is considered, the F-measure of the methods, such as NB, KNN, SVM, and DBN is 0.44, 0.52, 0.62, and 0.69, whereas the proposed SGCAV + DBN achieved the F-measure of 0.78.

The analysis in terms of accuracy by varying the training data percentage is depicted in Fig. 4d. Here, for 70% of training data, the previous techniques, such as NB, KNN, SVM, and DBN attained the accuracy of 0.49, 0.595, 0.665, and 0.709, but the proposed SGCAV + DBN acquired the accuracy of 0.8939. While considering 80% of training data, the existing techniques, such as NB, KNN, SVM, and DBN achieved the accuracy of 0.43, 0.56, 0.63, and 0.735, respectively. Meanwhile, the proposed SGCAV + DBN attained an accuracy of 0.775. From Fig. 4d, the proposed SGCAV + DBN is found to possess the maximum specificity of 0.9348 at 90% of training data.

5.5.1.2 For entropy = 200 The analysis of the comparative methods based on precision, recall, and accuracy for entropy = 200 is depicted in Fig. 5. Figure 5a shows the analysis in terms of precision by varying the training data percentage. For the training data = 60%, the existing techniques, like NB, KNN, SVM, and DBN possess the precision of 0.5, 0.578, 0.5823, and 0.6575, respectively, which is comparatively lower than SGCAV + DBN. For the same training data, the proposed SGCAV + DBN acquired the precision of 0.73. Similarly, when the training percentage

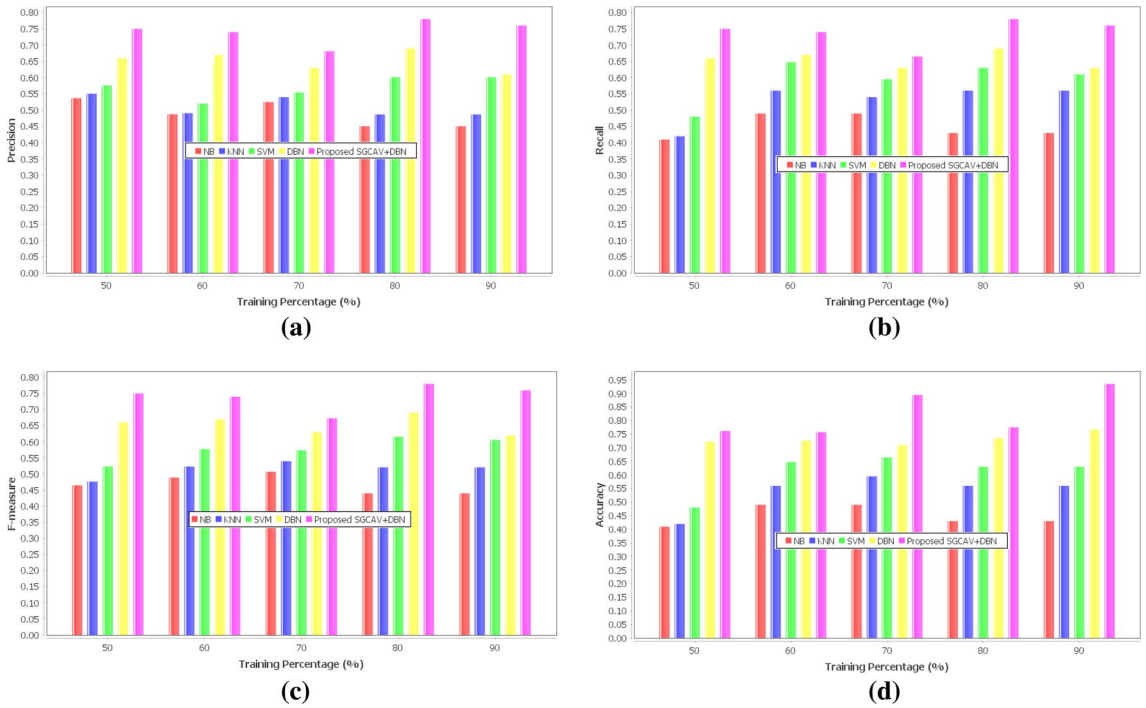

Fig. 4 Comparative analysis using 20 Newsgroups database for entropy = 100. **a** Precision, **b** recall, **c** F-measure and **d** accuracy

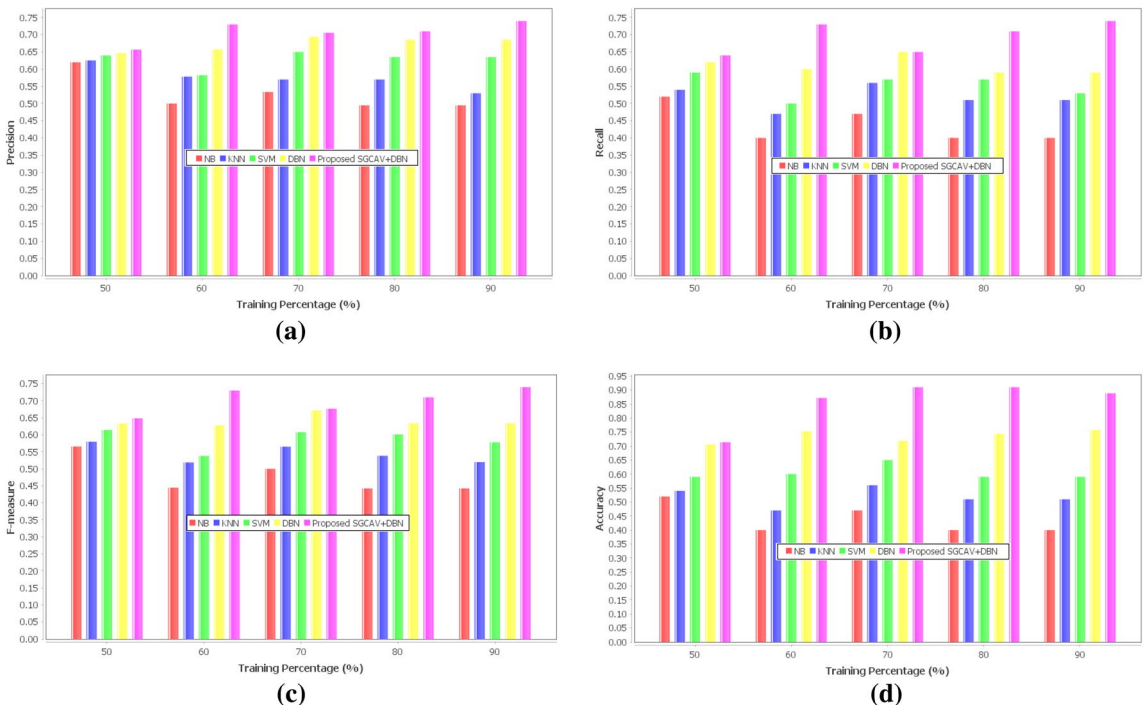

Fig. 5 Comparative analysis using 20 Newsgroups database for entropy = 200. **a** Precision, **b** recall, **c** F-measure and **d** accuracy

increased to 80%, the methods, NB, KNN, SVM, and DBN, attained the precision of 0.4948, 0.57, 0.6352, and 0.6856, respectively, whereas the precision of the developed method is 0.71. From the above interpretation, it is seen that the proposed SGCAV + DBN achieved increased precision of 0.74 at 90% training data.

The comparative analysis based on recall metric is depicted in Fig. 5b. When 70% of training data is considered, the existing methods, like NB, KNN, SVM, and DBN acquires the recall value of 0.47, 0.56, 0.57, and 0.65, respectively. Meanwhile, the proposed SGCAV + DBN obtained the recall value of 0.65. When 90% of training data is considered, the recall of the existing methods, like NB, KNN, SVM, and DBN is 0.4, 0.51, 0.53, and 0.59, respectively, whereas the proposed SGCAV + DBN attained the recall of 0.74.

The analysis in terms of F-measure metric is depicted in Fig. 5c. When 60% of training data is considered, the methods, such as NB, KNN, SVM, and DBN acquires the F-measure value of 0.44, 0.518, 0.538, 0.63, respectively. Meanwhile, the proposed SGCAV + DBN obtained the F-measure value of 0.73. When 80% of training data is considered, the F-measure of the methods, such as NB, KNN, SVM, and DBN is 0.44, 0.538, 0.601, and 0.634, whereas the proposed SGCAV + DBN achieved the F-measure of 0.71.

The analysis in terms of accuracy by varying the training data percentage is depicted in Fig. 5d. Here, for 80% of training data, the existing techniques, like NB, KNN, SVM, and DBN possess the accuracy of 0.4, 0.51, 0.59, and 0.744, but the proposed SGCAV + DBN acquired the accuracy of 0.910. While considering 90% of training data, the techniques, like NB, KNN, SVM, and DBN attained the accuracy of 0.4, 0.51, 0.59, and 0.7573, respectively. Meanwhile, the proposed SGCAV + DBN attained an accuracy of 0.888. From Fig. 5d, the proposed SGCAV + DBN are found to possess the maximum specificity of 0.9104 at 80% training data.

5.5.2 Comparative analysis using Reuter database

5.5.2.1 For entropy = 100 The analysis of the comparative methods based on precision, recall, and accuracy using the Reuter database for entropy = 100 is depicted in Fig. 6. Figure 6a shows the analysis in terms of precision by varying the training data percentage. For the training data = 50%, the techniques, such as NB, KNN, SVM, and DBN, possesses the precision of 0.4315, 0.4810, 0.5, and 0.504, respectively, which is comparatively lower than SGCAV + DBN. For the same training data, the proposed SGCAV + DBN acquired the precision of 0.5818. Similarly, when the training data increased to 90%, the methods, NB, KNN, SVM, and DBN, attained the precision of 0.3252, 0.3513, 0.3734, and 0.5636, whereas the precision of the proposed method is 0.7090. From the above interpretation, it is seen that the proposed SGCAV + DBN achieved increased precision of 0.7636 at 70% training data.

The comparative analysis in terms of recall metric is depicted in Fig. 6b. When 60% of training data is considered, the methods, like NB, KNN, SVM, and DBN acquire the sensitivity value of 0.2863, 0.3181, 0.3181, and 0.6090, respectively. Meanwhile, the proposed SGCAV + DBN obtained the recall value of 0.7. When 80% of training

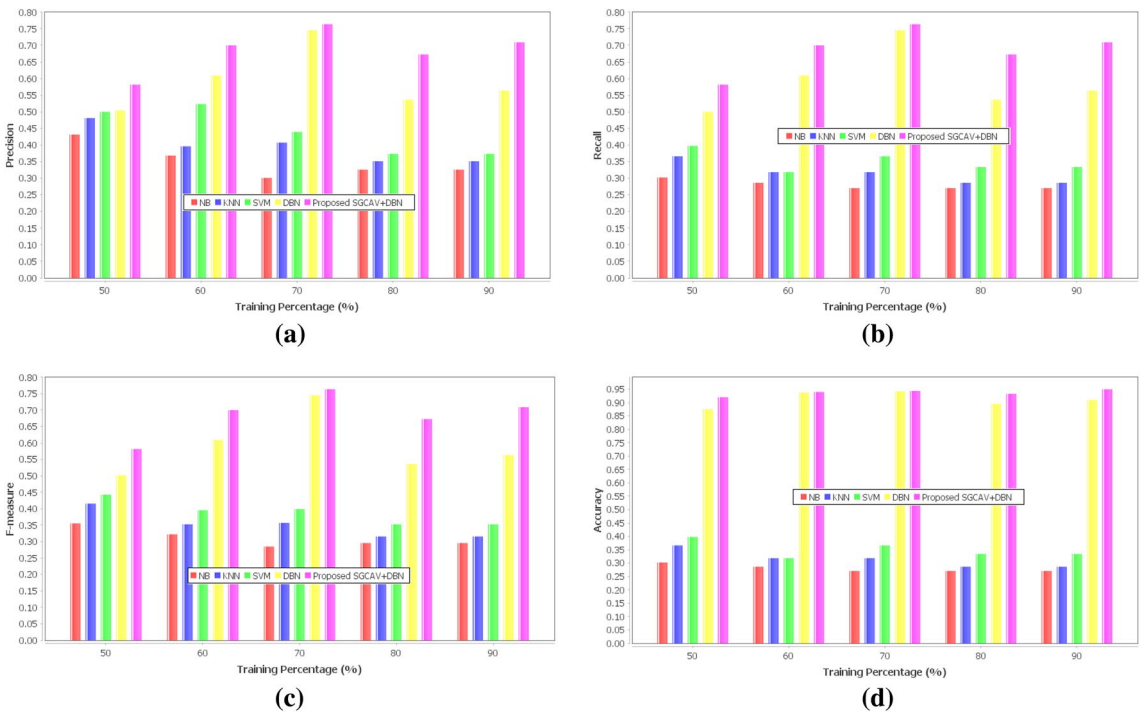

Fig. 6 Comparative analysis using 20 Newsgroups database for entropy = 100. **a** Precision, **b** recall, **c** F-measure and **d** accuracy

data is considered, the recall of the existing methods, like NB, KNN, SVM, and DBN is 0.270, 0.286, 0.334, 0.5363, respectively, whereas the developed SGCAV + DBN attained the recall of 0.6727.

The analysis in terms of F-measure metric is depicted in Fig. 6c. When 60% of training data is considered, the methods, such as NB, KNN, SVM, and DBN acquires the F-measure value of 0.32, 0.353, 0.395, 0.609, respectively. Meanwhile, the proposed SGCAV + DBN obtained the F-measure value of 0.7. When 80% of training data is considered, the F-measure of the methods, such as NB, KNN, SVM, and DBN is 0.29, 0.315, 0.352, and 0.536, whereas the proposed SGCAV + DBN achieved the F-measure of 0.672.

The analysis in terms of accuracy by varying the training data percentage is depicted in Fig. 6d. Here, for the 70% training data, the techniques, like NB, KNN, SVM, and DBN achieved the accuracy of 0.270, 0.3181, 0.3659, and 0.9425, but the proposed SGCAV + DBN acquired the accuracy of 0.9437. While considering 90% of training data, the existing techniques, like NB, KNN, SVM, and DBN attained the accuracy of 0.270, 0.286, 0.334, and 0.9103, respectively. Meanwhile, the proposed SGCAV + DBN attained an accuracy of 0.95. From Fig. 6d, the proposed SGCAV + DBN is found to possess the maximum specificity of 0.95 at 90% training data.

5.5.2.2 For entropy = 200 The analysis of the comparative methods based on precision, recall, and accuracy using the Reuter database for entropy = 200 is depicted in Fig. 7. Figure 7a shows the analysis based on precision by varying the training data percentage. For the training data = 50%, the existing techniques, like NB, KNN, SVM, and DBN, possess the precision of 0.3028, 0.4306, 0.4415, and 0.6273, respectively, which is comparatively lower than the proposed SGCAV + DBN. For the same training data, the developed SGCAV + DBN acquired the precision of 0.6727. Similarly, when the training data increased to 90%, the methods, NB, KNN, SVM, and DBN, attained the precision of 0.2431, 0.3357, 0.3810, and 0.7090, whereas the precision of the developed method is 0.7090. From the above interpretation, it is seen that the proposed SGCAV + DBN achieved improved precision of 0.7545 at 80% training data.

The comparative analysis in terms of recall metric is depicted in Fig. 7b. When 60% of training data is considered, the existing methods, like NB, KNN, SVM as well as DBN acquire the sensitivity value of 0.3181, 0.4136, 0.4136, and 0.6090, respectively. Meanwhile, the proposed SGCAV + DBN obtained the recall value of 0.6363. When 80% of training data is considered, the recall of the existing methods, like NB, KNN, SVM, and DBN is 0.2545, 0.2704, 0.2863, and 0.7090, whereas the proposed SGCAV + DBN attained the recall value of 0.7090.

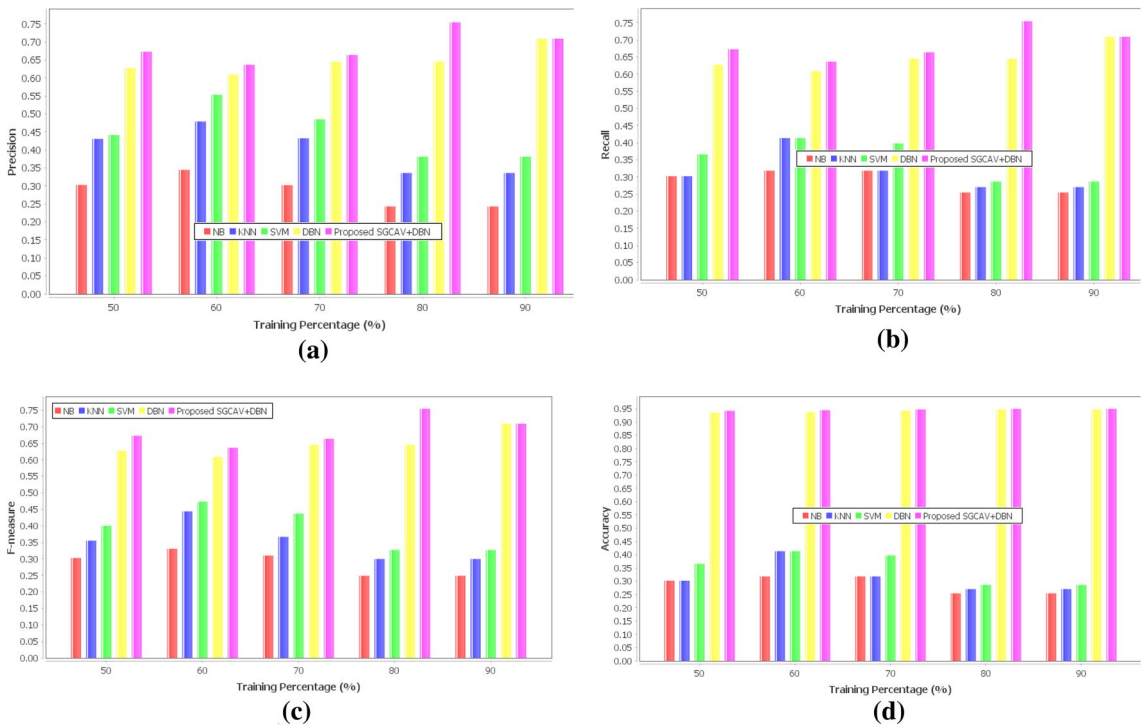

Fig. 7 Comparative analysis using 20 Newsgroups database for entropy = 200. **a** Precision, **b** recall, **c** F-measure and **d** accuracy

The analysis in terms of F-measure metric is depicted in Fig. 7c. When 60% of training data is considered, the methods, such as NB, KNN, SVM, and DBN acquires the F-measure value of 0.331, 0.443, 0.473, 0.609, respectively. Meanwhile, the proposed SGCAV + DBN obtained the F-measure value of 0.636. When 80% of training data is considered, the F-measure of the methods, such as NB, KNN, SVM, and DBN is 0.249, 0.3, 0.327, and 0.645, whereas the proposed SGCAV + DBN achieved the F-measure of 0.755.

The analysis in terms of accuracy by varying the training data percentage is depicted in Fig. 7d. Here, for 60% of training data, the methods, like NB, KNN, SVM, and DBN achieved the accuracy of 0.3181, 0.4136, 0.4136, and 0.9375, but the proposed SGCAV + DBN acquired the accuracy of 0.945. While considering 80% of training data, the techniques, like NB, KNN, SVM, and DBN achieved the accuracy of 0.2545, 0.2704, 0.2863, and 0.9475, respectively. Meanwhile, the proposed SGCAV + DBN attained an accuracy of 0.95. From Fig. 7d, the proposed SGCAV + DBN are found to possess the maximum specificity of 0.95 at 80%, and 90% of training data.

5.6 Comparative discussion

Table 1 depicts the comparative discussion of the existing NB, KNN, SVM, and DBN and the proposed SGCAV + DBN in terms of precision, recall, F-measure, and accuracy parameters by varying the training data percentage. The maximum performance measured by proposed SGCAV + DBN in terms of the precision parameter is 0.78, whereas the precision values of existing NB, KNN, SVM, and DBN are 0.62, 0.6252, 0.65, and 0.6943, respectively. The maximal recall is computed by the proposed SGCAV + DBN with a value of 0.78, whereas the existing NB, KNN, SVM, and DBN acquired the recall of 0.52, 0.56, 0.6475, and 0.69, respectively. The maximum performance measured by proposed SGCAV + DBN in terms of F-measure parameter is 0.78, whereas the F-measure values of existing NB, KNN, SVM, and DBN are 0.53, 0.55, 0.60, and 0.69, respectively. The accuracy value computed by the proposed SGCAV + DBN is 0.9382, whereas the existing NB, KNN, SVM, and DBN methods acquired the accuracy of 0.52, 0.595, 0.665, and 0.7662, respectively.

From Table 2, in the existing methods, DBN has maximum precision, recall, F-measure, and accuracy. The proposed system has 10.98%, 11.54%, 11.538%, and 18.33% better precision, recall, F-measure and accuracy, than the DBN.

Table 3 depicts the comparative discussion of the existing NB, KNN, SVM, and DBN and the proposed SGCAV + DBN in terms of precision, recall, F-measure, and accuracy parameters by varying the training data percentage. The maximum performance measured by proposed SGCAV + DBN in terms of the precision parameter is 0.7636, whereas the precision values of existing NB, KNN, SVM, and DBN are 0.4315, 0.4810, 0.5532, and 0.7454, respectively. The maximal recall is computed by proposed SGCAV + DBN with a value of 0.7636, whereas the existing NB, KNN, SVM, and DBN acquired the recall of 0.3181, 0.4136, 0.4136, and 0.7454, respectively. The maximum performance measured by proposed SGCAV + DBN in terms of F-measure parameter is 0.7636, whereas the F-measure values of existing NB, KNN, SVM, and DBN are 0.3555, 0.4156, 0.443, and 0.7455, respectively. The accuracy value computed by the proposed SGCAV + DBN is 0.95, whereas the existing NB, KNN, SVM, and DBN methods acquired the accuracy of 0.3181, 0.4136, 0.4136, and 0.948, respectively.

From Table 3, in the existing methods, DBN has maximum precision, recall, F-measure, and accuracy. The proposed system has 2.38%, 2.38%, 2.37% and 0.21% better precision, recall, F-measure, and accuracy, than the DBN.

Table 2 Comparative analysis based on 20 Newsgroup database

Methods	Precision	Recall	F-measure	Accuracy
NB	0.62	0.52	0.53	0.52
KNN	0.6252	0.56	0.55	0.595
SVM	0.65	0.6475	0.60	0.665
DBN	0.6943	0.69	0.69	0.7662
Proposed SGCAV + DBN	0.78	0.78	0.78	0.9382

Table 3 Comparative analysis in terms of Reuter database

Methods	Precision	Recall	F-measure	Accuracy
NB	0.4315	0.3181	0.3555	0.3181
KNN	0.4810	0.4136	0.4156	0.4136
SVM	0.5532	0.4136	0.443	0.4136
DBN	0.7454	0.7454	0.7455	0.948
Proposed SGCAV + DBN	0.7636	0.7636	0.7636	0.95

6 Conclusion

This research paper presents an approach for text categorization using SGCAV + DBN. At first, the documents are given to pre-processing to remove redundant and unnecessary words from the data using two steps, namely, stop word removal and stemming. After pre-processing, the feature extraction is carried out using the vector space model to find the important keywords from the document. Then, the feature selection is performed based on entropy, and at last, the text categorization is done using the proposed

SGCAV + DBN. Thus, the proposed SGCAV + DBN method is utilized for training DBN. Experimentation is carried out using two databases, namely 20 newsgroup databases and the Reuter database. The performance of the SGCAV + DBN is evaluated based on recall, precision, F-Measure, and accuracy. The proposed method produces the maximal precision of 0.78, maximal recall of 0.78, maximal F-measure of 0.78, and the maximal accuracy of 0.95, which indicates the superiority of the proposed method. The proposed text categorization method can be used in different fields, such as document organization, spam email filtering, and news groupings. In future, the proposed SGCAV + DBN will be further enhanced using a recent hybrid optimization approach to improve the classification performance.

References

1. Al-Salemi B, Ayob M, Noah SAM (2018) Feature ranking for enhancing boosting-based multi-label text categorization. Expert Syst Appl 113:531–543
2. Tellez ES, Moctezuma D, Miranda-Jiménez S, Graff M (2018) An automated text categorization framework based on hyper parameter optimization. Knowl-Based Syst 149:110–123
3. Saad MK, Ashour W (2010) Arabic text classification using decision trees. In: Proceedings of 12th international workshop on computer science and information technologies CSIT, Moscow-Saint Petersburg, Russia
4. Mohammad AH, Alwadan T, Al-Momani O (2016) Arabic text categorization using support vector machine. Naïve Bayes Neural Netw 5(1):108–115
5. Tang B, He H, Baggenstoss PM, Kay S (2016) A Bayesian classification approach using class-specific features for text categorization. IEEE Trans Knowl Data Eng 28(6):1602–1606
6. Lee J, Yu I, Park J, Kim DW (2019) Memetic feature selection for multilabel text categorization using label frequency difference. Inf Sci 485:263–280
7. Alwehaibi A, Roy K (2018) Comparison of pre-trained word vectors for arabic text classification using deep learning approach. In: Proceedings of 17th IEEE international conference on machine learning and applications (ICMLA), Orlando, FL, pp 1471–1474
8. Hu Y, Yi Y, Yang T, Pan Q (2018) Short text classification with a convolutional neural networks based method. In: Proceedings of 15th international conference on control, automation, robotics and vision (ICARCV), Singapore, pp 1432–1435
9. Xu Z, Li J, Liu B, Bi J, Li R, Mao R (2017) Semi-supervised learning in large scale text categorization. J Shanghai Jiatong Univ 22(3):291–302
10. Attaccalite C, Cannuccia E, Grüning M (2017) Excitonic effects in third-harmonic generation: the case of carbon nanotubes and nanoribbons. Phys Rev B 95(12):125403
11. Nguyen HM, Khoa BT (2019) The relationship between the perceived mental benefits, online trust, and personal information disclosure in online shopping. J Asian Finance 6(4):261–270
12. Tu F, Yin S, Ouyang P, Tang S, Liu L, Wei S (2017) Deep convolutional neural network architecture with reconfigurable computation patterns. IEEE Trans Very Large Scale Integr Syst 25(8):2220–2233
13. Ninu Preetha NS, Praveena S (2018) Multiple feature sets and SVM classifier for the detection of diabetic retinopathy using retinal images. Multimed Res 1(1):17–26
14. Alzubi J, Nayyar A, Kumar A (2018) Machine learning from theory to algorithms: an overview. J Phys: Conf Ser 1142:012012
15. Bhopale AP, Kamath SS, Tiwari A (2018) Concise semantic analysis based text categorization using modified hybrid union feature selection approach. In: Proceedings of 4th international conference on recent advances in information technology (RAIT), Dhanbad, pp 1–7
16. Haryanto AW, Mawardi EK, Muljono (2018) Influence of word normalization and chi squared feature selection on support vector machine (SVM) text classification. In: Proceedings of international seminar on application for technology of information and communication, Semarang, pp 229–233
17. Zheng T, Wang L (2018) Unlabeled text classification optimization algorithm based on active self-paced learning. In: Proceedings of IEEE international conference on big data and smart computing (BigComp), pp 404–409
18. Parmar PS, Biju PK, Shankar M, Kadiresan N (2018) Multiclass text classification and analytics for improving customer support response through different classifiers. In: Proceedings of international conference on advances in computing, communications and informatics (ICACCI), Bangalore, pp 538–542
19. Bigi B (2003) Using Kullback–Leibler distance for text categorization. In: Advances in information retrieval, vol 2633. Springer, Berlin, pp 305–319
20. Ma T, Motta G, Liu K (2017) Delivering real-time information services on public transit: a framework. IEEE Trans Intell Transp Syst 18(10):2642–2656
21. Kouretas GP, Zarangas L (2005) Conditional autoregressive value at risk by regression quantiles estimating market risk for major stock markets, no. 0521
22. Kim S-B, Han K-S, Rim H-C, Myaeng SH (2006) Some effective techniques for naive Bayes text classification. IEEE Trans Knowl Data Eng 18(11):1457–1466
23. Liu C, Wang W, Tu G, Xiang Y, Wang S, Lv F (2017) A new centroid-based classification model for text categorization. Knowl Based Syst 136:15–26
24. Tang X, Dai Y, Xiang Y (2019) Feature selection based on feature interactions with application to text categorization. Expert Syst Appl 120:207–216
25. Zheng T, Zheng T, Wang L (2018) Unlabeled text classification optimization algorithm based on active self-paced learning. In: Proceedings of IEEE international conference on big data and smart computing
26. Liu B, Xiao Y, Hao Z (2018) A selective multiple instance transfer learning method for text categorization problems. Knowl-Based Syst 141:178–187
27. Kim K, Zhang SY (2018) Trigonometric comparison measure: a feature selection method for text categorization. Data Knowl Eng 119:1–12
28. Feng G, Li S, Sun T, Zhang B (2018) A probabilistic model derived term weighting scheme for text classification. Pattern Recogn Lett 110:23–29
29. Yang J, Yang G (2018) Modified convolutional neural network based on dropout and the stochastic gradient descent optimizer. Algorithms 11(3):28
30. Dai W, Xue G-R, Yang Q, Yu Y (2007) Transferring Naive Bayes classifiers for text classification. In: AAAI, vol 7, pp 540–545
31. Camastra F, Razi G (2019) Italian text categorization with lemmatization and support vector machines. In: Neural approaches to dynamics of signal exchanges, vol 151, pp 47–54
32. Jo T (2019) Improving K nearest neighbor into string vector version for text categorization. In: 21st international conference on advanced communication technology (ICACT), PyeongChang Kwangwoon_Do, Korea (South)

33. Berge GT, Granmo O-C, Tveit TO, Goodwin M, Jiao L, Matheussen BV (2019) Using the Tsetlin machine to learn human-interpretable rules for high-accuracy text categorization with medical applications. In: IEEE Access, vol 7, pp 115134–115146
34. Engle RF, Manganelli S (2004) CAViaR: conditional autoregressive value at risk by regression quantiles. J Bus Econ Stat 22(4):367–381
35. Ranjan NM, Prasad RS (2018) LFNN: lion fuzzy neural network-based evolutionary model for text classification using context and sense based features. Appl Soft Comput J 71:994–1008
36. Huang D, Yu B, Fabozzi FJ, Fukushima M (2009) CAViaR-based forecast for oil price risk. Energy Econ 31:511–518
37. Hinton GE, Osindero S, Teh Y (2006) A fast learning algorithm for deep belief nets. Neural Comput 18:1527–1554
38. Zinkevich M, Weimer M, Li L, Smola AJ (2010) Parallelized stochastic gradient descent. In: Advances in neural information processing systems 23 (NIPS 2010)
39. Newsgroup database. http://qwone.com/~jason/20Newsgroups/. Accessed October 2018
40. Reuter database. https://archive.ics.uci.edu/ml/machine-learningdatabases/reuters21578-mld/. Accessed October 2018
41. Wajeed MA, Adilakshmi T (2011) Using KNN algorithm for text categorization. In: Proceedings of international conference on computational intelligence and information technology, pp 796–801
42. Parmar PS, Biju PK, Shankar M, Kadiresan N (2018) Multiclass text classification and analytics for improving customer support response through different classifiers. In: Proceedings of international conference on advance in computing, communications, and informatics (ICACCI)