DOI 10.1007/s00500-016-2189-8

Study on suitability and importance of multilayer extreme learning machine for classification of text data

Rajendra Kumar Roul[1,2] · Shubham Rohan Asthana[1] · Gaurav Kumar[1]

Published online: 21 June 2016
© Springer-Verlag Berlin Heidelberg 2016

Abstract The dynamic Web, which contains huge number of digital documents, is expanding day by day. Thus, it has become a tough challenge to search for a particular document from such a large volume of collections. Text classification is a technique which can speed up the search and retrieval tasks and hence is the need of the hour. Aiming in this direction, this study proposes an efficient technique that uses the concept of connected component (CC) of a graph and Wordnet along with four established feature selection techniques [e.g., TF-IDF, Chi-square, Bi-Normal Separation (BNS) and Information Gain (IG)] to select the best features from a given input dataset in order to prepare an efficient training feature vector. Next, multilayer extreme learning machine (ML-ELM) (which is based on the architecture of deep learning) and other state-of-the-art classifiers are trained on this efficient training feature vector for classification of text data. The experimental work has been carried out on DMOZ and 20-Newsgroups datasets. We have studied the behavior and compared the results of different classifiers using these four important feature selection techniques used for classification process and observed that ML-ELM achieved the maximum overall F-measure of 72.28 % on DMOZ dataset using TF-IDF as the feature selection technique and 81.53 % on 20-Newsgroups dataset using BNS as the feature selection technique compared to other state-of-the-art classifiers which signifies the usefulness of deep learning used by ML-ELM for classifying the text data. Experimental results on these benchmark datasets show the stability and effectiveness of our approach over other competing approaches.

Keywords Connected component · Deep learning · Extreme learning machine · Feature selection · Multilayer extreme learning machine

1 Introduction

The Web has indexed at least 4.76 billion of documents.[1] Organizing these documents on the Web in an effective manner is the real challenge for the present search engine. The ultimate aim of the search engine is to satisfy the internet user who is looking for the desired information every time he queries. The most time-consuming job is searching these informations in the net. If this happens efficiently, then the user can effectively absorb and use the knowledge of the documents. Text classification is an attempt in this direction, which not only reduces the searching time but also makes available the required information to the user for which he is looking for. It is a vital topic in machine learning where learning is done over the text. Classification is a well-known machine learning technique where the set of label datasets is used to trained the classifier before it is applied to the test dataset for deciding the target class. Based on the number of classes used in the process, classification can be broadly classified into two categories: binary classification where a test instance is categorized into one of two predefined classes and multi-class classification where the test instance deals with more than two classes. In order to classify the text data more effectively, selection of top features is highly essential and this in turn generates a technique called feature selection

Communicated by A. Di Nola.

✉ Rajendra Kumar Roul
rkroul@goa.bits-pilani.ac.in

[1] BITS, Pilani-K.K.Birla Goa Campus, Goa, India

[2] Department of Computer Science, BITS, Pilani-K.K.Birla Goa Campus, Goa, India

[1] http://www.worldwidewebsize.com/.

upon which the generalization capability of a machine learning algorithm depends. The performance of the classifiers depends on how robust the feature vector is. Feature selection involves reducing the number of features by selecting a subset of it which would help in building the required model. It is important in text classification for two main reasons:

1. Effective number of features are reduced, and hence, training the classifier will consume less network bandwidth, time and storage in the training phase.
2. Classification errors due to noise features are eliminated, and thus, accuracy of classification process improves.

Generally, the feature selection methods are either unsupervised or supervised. As the name suggests 'unsupervised' and hence no class labels are required to select the top features, but on the other hand supervised methods do require class labels. Some of the unsupervised feature selection methods are 'document frequency,' 'term contribution,' 'TF-IDF metric,' etc. Supervised feature selection methods are further categorized into two sub-categories: accuracy-based and correlation-based.

1. Accuracy-based: This method chooses the features which maximize the occurrence of features in the positive class and minimize the occurrences of the features in the negative class. Some of the existing methods are odds ratio,' 'probability ratio,' 'GU metric,' 'Bi-Normal Separation (BNS),' 'power metric,' 'Fisher criterion,' etc.
2. Correlation-based: This method evaluates the features by finding the correlation of the features with the various classes and choose the features which have the highest correlation score. For example, 'Chi-square metric,' 'NGL coefficient,' 'GSS coefficient,' 'MI-judge' and 'Information Gain' are some of the existing correlation-based methods.

The techniques used for feature selection are categorized as wrapper, filters and embedded methods. For constructing a feature set, wrapper and embedded methods need the involvement of classifier which increases the running time and computationally intensive. But filter method does not require any classifier interaction for preparing the feature set and hence more preferable compared to the other two methods.

The next important thing after the feature selection which affects the text classification process is an efficient classifier. There are many traditional classifiers exist for text classification which includes decision trees, k-nearest neighbor, Naive Bayes, SVM etc. But they have their own limitations, and most of them use the shallow neural networks algorithms in which there are certain restrictions for the capabilities to achieve approximating the complex function. Deep learning has aroused interest in the past decade in many research domains such as computer vision, automatic speech recognition and pattern recognition and recently has attracted much attention in the field of machine learning. It is a multilayer perception artificial neural network algorithm. There is no such restriction found in deep learning (i.e., capabilities to achieve approximating the complex function) which removes the difficulty of optimization associated with the deep models (Ding et al. 2015) and achieves an approximation of complex function. Extreme Learning Machine (Huang et al. 2006) is able to approximate any complex nonlinear mappings directly from the training samples, but it has shallow architecture similar to traditional SLFNs. Hence, it may need a large network to perfectly fit the highly variant input data, which is difficult to implement. Recently designed multilayer ELM (Kasun et al. 2013) is able to address this issue which combines deep learning (i.e., ELM autoencoder) with ELM, decomposes the original input data into multiple hidden layers and performs unsupervised learning layer-wise.

Considering that selection of informative features and efficient classifier are able to generate good performance for text classification process, this study uses ML-ELM as the classifier which earned name quickly in the field of machine learning owing to its fast speed, easy implementation and ability to handle a large volume of data. To prepare an efficient feature vector, we have considered four standard feature selection techniques, such as TF-IDF, Chi-square, BNS and IG, which are generally used for text classification. The concept of connected component of a graph along with the Wordnet has been used that help us for selecting the top features from each class of a given corpus after calculating the TF-IDF/Chi-square/BNS/IG for each feature (i.e., keyword) of a class. Finally, the reduced feature vector of each class is combined together to form the final reduced feature vectors [one for each feature selection technique (i.e., TF-IDF, Chi-square, BNS and IG)]. ML-ELM and other traditional classifiers including ELM and SVM are trained on these final reduced feature vectors for the classification of text data. The experimental work which focused on text classification process is carried out on two benchmark datasets: DMOZ (Open Directory Project) and 20-Newsgroups. The performance of different classifiers is compared in the experimental section, and it has been observed that ML-ELM outperforms the other established classifiers including ELM and SVM. The empirical results show that the performance of the proposed approach is promising compared to other existing approaches.

The paper is outlined in this way: The literature review based on different classification techniques used for text data is discussed in Sect. 2. Section 3 describes different existing feature selection techniques and model structure of ELM and ML-ELM. The proposed approach for classifying the text data is discussed in Sect. 4. Section 5 describes the analysis of empirical results and compares the proposed approach with other existing approaches. We concluded our work with some future enhancements in Sect. 6.

2 Literature review

Recently, ELM and ML-ELM have attracted the attention of many researchers in the field of text classification. Working in this direction, Huang et al. (2012) in their approach have discussed three important things. First, ELM provides unified learning platform, second, compared to PSVM and LS-SVM, ELM has less optimization constraints and third, in theory ELM can classify any disjoint regions and approximate any target continuous function. Their simulation results show that ELM has good performance and scalability at much faster learning speed compared to SVM and LS-SVM. Bai et al. (2014) have worked on sparse ELM and showed that sparse ELM can reduce the training time and storage space compared to the unified ELM. It has very good performance with faster learning speed compared to the state-of-the-art SVM classifier. It also has the ability to handle large-scale binary classification compared to the unified ELM. Ding et al. (2014) have introduced ELM and described different principles and algorithms used in ELM. In their studies, typical variants of ELM like incremental ELM, two-stage ELM, pruning ELM, error-minimized ELM, evolutionary ELM and online sequential ELM have been described. They have summarized the applications of ELM for classification, function approximation, regression, pattern recognition, etc.

Very less research work has been done where ML-ELM is used as the classifier (Ding et al. 2015; Mirza et al. 2016; Yang and Wu 2015; Tang et al. 2014). Many other state-of-the-art mechanisms have also been used for text classification. A new Web page classification based on SVM-weighted voting scheme has been proposed by Chen and Hsieh (2006). In their work, latent semantic analysis is used to find the hidden information from the documents and to extract text features from each Web page. This helps the SVM to classify the Web pages. Experimental results show that their approach is better than the traditional approaches. Wan et al. (2012) have introduced a new text document classification, which is a combination of k-nearest neighbor (kNN) and SVM techniques. They have tested their approach on many benchmark datasets, and the results show that the accuracy of the combined approach has less impact on the values of the parameters as compared to the traditional kNN technique. A rough set approach to SVM classification is proposed by Lingras and Butz (2007), which is mostly useful when handling noisy data. Their work has proposed two new approaches, extension (1-v-r) and (1-v-1) to SVM multi-classification by using the boundary region in rough sets. They have justified that extended (1-v-r) can reduce the training time of the traditional (1-v-r) approach. The experimental results support their theoretical results. Gomez and Moens (2012) have discussed a method to classify the Web documents into a predefined hierarchy using textual content of the documents. They have developed a Stratified Discriminant Analysis (SDA) technique to reduce the feature vectors of the Web documents. Rujiang et al. (2011) have suggested a model called SUMO (The Suggested Upper Merged Ontology) based on text classification, which is integrated with Wordnet ontology to classify the Web pages. Experimentally they claimed that their method can reduce the dimensionality of the vector space and increase the performance of the text classification. Li et al. (2012) have proposed a hierarchical-vertical classification of framework that built a hierarchical classifier after discovering the inherent hierarchical structure of relationships among vertical Web pages based on flat datasets. They have used SVM using odds ratio to select discriminative features which obtained best results. Klassen and Paturi (2010) have worked on a technique for Web pages classification using keywords as the attributes from documents and random forest learning method. Their work identifies that the random forest learning method is better than other state-of-the-art machine learning mechanisms for classification.

Introducing ML-ELM which uses deep learning extensively in the field of text classification can begin a new era in the field of machine learning. Our approach has used Wordnet and connected component of the graph to select the best features using different feature selection techniques. Experimental results on two large benchmark datasets demonstrate the effectiveness of our approach over the other existing approaches.

3 Background

3.1 Different feature selection techniques

This section discusses the most important existing feature selection techniques we have used in our proposed work for feature vectors preparation and the architecture of ELM and multilayer ELM.

i. *TF-IDF:*
Rare appearance of features (or words) in a text document reflects the category of the text document in a better manner. To identify such important words, term frequency-inverse document frequency[2], a statistical measure has been used extensively. Term frequency (TF) or local frequency of a word w in a document d indicates how important the word w for d is. $TF_{w,d} =$

[2] http://nlp.stanford.edu/IR-book/html/htmledition.

$\left(\frac{p}{q}\right)$ where 'p' represents frequency of w in d and 'q' represents sum of frequency of all the words in d. Inverse document frequency (IDF) or global frequency of a word w in the entire corpus C measures how important the word w for C is. $IDF_w = log\left(\frac{r}{s}\right)$, where 'r' represents the total number of documents in C and 's' represents the number of documents of C which contain the word w.

$$(TF\text{-}IDF)_w = TF_w \times IDF_w$$

ii. *Chi-square* (χ^2):
This technique is based on Chi-square distribution of statistics and generally used to test the independence of two events. In feature selection, the two events are occurrence of the keyword and occurrence of the class. It measures the confidence in association between two categorical variables (based on available statistics). The keywords are ranked with respect to Eq. 1 mentioned below.

$$\chi^2(w, c) = \sum_{e_w \in 0,1} \sum_{e_c \in 0,1} \frac{(O_{e_w e_c} - E_{e_w e_c})^2}{E_{e_w e_c}} \quad (1)$$

where w is the word and c is the class of documents, 'O' and 'E' represent the observed and the expected frequency, respectively (Manning et al. 2008), and e_w and e_c are the binary variables. If a document d contains w, then $e_w = 1$ else $e_w = 0$. Similarly, if the class c contains the document d, then $e_c = 1$ else $e_c = 0$.

iii. *Information Gain*:
Information Gain (IG) of a word w measures how much presence or absence of w in a document d affects the class c to take a correct decision on classification. It is a measure of the decrease in entropy of the class variable after the value for the word is observed, and it can be generalized to any number of classes (Yang and Pedersen 1997). Equation 2 measures the Information Gain of w.

$$\begin{aligned} IG(w) = &-\sum_{i=1}^{m} p(c_i)log\ p(c_i) \\ &+ p(w)\sum_{i=1}^{m} p(c_i|w)log\ p(c_i|w) \quad (2) \\ &+ p(\overline{w})\sum_{i=1}^{m} p(c_i|\overline{w})log\ p(c_i|\overline{w}) \end{aligned}$$

where,
m: number of predefined classes,
$p(c_i)$: a prior probability of ith class,
$p(w)$: probability of word w in a given data set,
$p(c_i|w)$: conditional probability of ith class given w,
$p(\overline{w})$: complementary probability of $p(w)$, and
$p(c_i|\overline{w})$: conditional probability of ith class in the absence of w.

iv. *Bi-Normal Separation*:
Bi-Normal Separation (BNS) originally developed by Forman (2003) tries to find the words which have high difference between their tpr (true-positive rate) and fpr (false-positive rate). It is the difference between the inverse of the standard normal distribution of the true-positive and false-positive rate and is represented in Eq. 3.

$$BNS(w, c_i) = \left|\phi^{-1}\left(\frac{n_{iw}}{n_i}\right) - \phi^{-1}\left(\frac{n_{\bar{i}w}}{n_{\bar{i}}}\right)\right| \quad (3)$$

where,
n_i: number of documents belongs to class c_i,
n_{iw}: number of documents contains the word w and belongs to the class c_i,
$n_{\bar{i}}$: number of documents not belongs to class c_i,
$n_{\bar{i}w}$: number of documents contains the word w but does not belongs to the class c_i, and
ϕ^{-1}: inverse of the standard normal distribution.

3.2 Extreme learning machine

ELM proposed by Huang et al. (2006) is a single-layer feed-forward neural networks (SLFNs). ELM become popular over the other established classifiers which is mainly due to the following reasons:

(i) Input weights and hidden layer biases adjustment which consumes more time are not required in ELM as they are assigned randomly.
(ii) Neither hidden layer requires to be tuned nor to be neuron alike.
(iii) Easy to implement and very fast learning speed.
(iv) Ability to handle a large volume of data.
(v) No back propagation.
(vi) Gives good performance with less human intervention.
(vii) Avoids local minimization.
(viii) Parallelization of computation.
(ix) Produces one optimal solution with negligible errors.

The computational speed of ELM is exceptionally good compared to SVM, and this increases drastically when the training dataset increases (Liu et al. 2012).

ELM at a Glance:
For N arbitrary distinct examples (x_i, y_i), where $x_i = [x_{i1}, x_{i2}, \ldots, x_{in}]^T \in R^n$ and $y_i = [y_{i1}, y_{i2}, \ldots, y_{im}]^T \in R^m$, such that $(x_i, y_i) \in R^n \times R^m$, $i = 1, 2, \ldots, N$. Along with this, ELM is having an activation function $g(x)$ and L

hidden nodes. For a given input **x**, the output function of extreme learning machine is as follows:

$$\boldsymbol{g_L(x_j) = \sum_{i=1}^{L} \beta_i g(w_i \cdot x_j + b_i) = y_j, j = 1, \ldots, N} \tag{4}$$

Here, (w_i, b_i) are hidden node parameters generated randomly where i lies between 1 and L, $w_i = [w_{i1}, w_{i2} \ldots w_{in}]^T$ represents the weight vector which connects the input nodes of 'n' numbers into the ith hidden node and b_i is the bias of ith hidden node. $\boldsymbol{\beta}$ which connects each hidden node to every output nodes is the weight vector and is represented as $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_L]^T$. The output vector $g(\mathbf{x})$ maps the n-dimensional input space to a L-dimensional feature space. Here, $\boldsymbol{H}$ represents the output matrix of hidden layer. The compact form of Eq. 4 is represented by Eq. 5 as follows:

$$\boldsymbol{H\beta = Y} \tag{5}$$

where

$$\boldsymbol{H} = \begin{bmatrix} g(\boldsymbol{w_1} \cdot \boldsymbol{x_1} + b_1) & \ldots & g(\boldsymbol{w_L} \cdot \boldsymbol{x_1} + b_L) \\ g(\boldsymbol{w_1} \cdot \boldsymbol{x_2} + b_1) & \ldots & g(\boldsymbol{w_L} \cdot \boldsymbol{x_2} + b_L) \\ . & \cdots\cdot & . \\ . & \cdots\cdot & . \\ . & \cdots\cdot & . \\ g(\boldsymbol{w_1} \cdot \boldsymbol{x_N} + b_1) & \ldots & g(\boldsymbol{w_L} \cdot \boldsymbol{x_N} + b_L) \end{bmatrix}_{N \times L}$$

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_{11} \ldots \beta_{1m} \\ \beta_{21} \ldots \beta_{2m} \\ . \\ . \\ . \\ \beta_{L1} \ldots \beta_{Lm} \end{bmatrix}_{L \times m} \quad \boldsymbol{Y} = \begin{bmatrix} y_{11} \ldots y_{1m} \\ y_{21} \ldots y_{2m} \\ . \\ . \\ . \\ y_{N1} \ldots y_{Nm} \end{bmatrix}_{N \times m}$$

Till the number of hidden layer nodes is large enough, the parameters of the network do not all need to adjust (Huang 2003). Smallest training error and smallest norm of output weights can be achieved by ELM and can be represented by Eq. 6 as follows:

$$minimize : \| \boldsymbol{H\beta - Y} \|^2 \; and \; \| \boldsymbol{\beta} \| \tag{6}$$

β can be derived in many ways and one of such technique to derive β is using Moore–Penrose (Liang et al. 2006) generalized inverse of matrix $\boldsymbol{H}$ which when multiplied with Y gives $\boldsymbol{\beta}$. The system diagram of Extreme Learning Machine is shown in Fig. 1.

3.3 Multilayer ELM

Multilayer ELM is a machine learning approach based on the architecture of artificial neural network and is inspired by deep learning and extreme learning machine. Deep learning was first proposed by Hinton and Salakhutdinov (2006) who in their work used deep structure of multilayer autoencoder and established a multilayer neural network on the unsupervised data. In their proposed method, first they used an unsupervised training to obtain the parameters in each layer. Next, the network is fine-tuned by supervised learning. Hinton et al. (2006), who proposed the deep belief network, outperforms the traditional multilayer neural network, SLFNs, SVMs, but it has slow learning speed. Working in this direction, recently Kasun et al. (2013) proposed multilayer ELM which performs unsupervised learning from layer to layer, and it does not need to iterate during the training process, and hence, it does not spend a long time in the training phase. Compared to other conventional deep networks, it

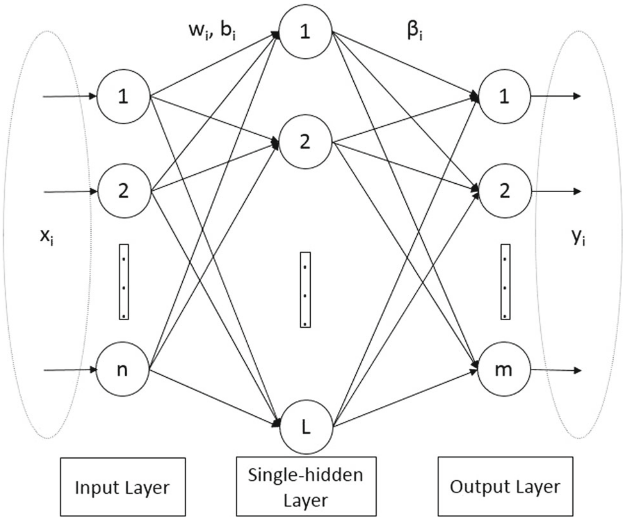

Fig. 1 Architecture of ELM

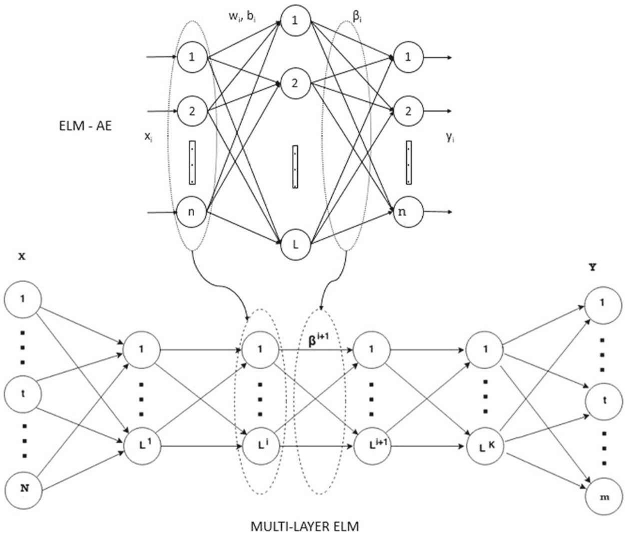

Fig. 2 ELM-AE and multilayer ELM

has a better or comparable performance. Figure 2 shows the system architecture of ML-ELM.

3.3.1 ELM autoencoder (ELM-AE)

Autoencoder is an unsupervised neural network. The outputs and inputs of the autoencoder are same. Like ELM, ELM-AE has 'n' input layer nodes, single hidden layer of 'L' nodes and 'n' output layer nodes. In spite of many resemblance between these two, there are two major differences that exist between them which are as follows:

i. ELM is a supervised neural network and the output of ELM is a class label while ELM-AE is an unsupervised one and its output is same as the input.
ii. Input weights and biases of the hidden layer are random in case of ELM, but they are orthogonal in ELM-AE.

Depending on the number of hidden layer nodes, the ELM-AE can be divided into the following three categories.

(i) Compressed representation ($n > L$):
In compressed representation, features of training dataset need to be represented from a higher-dimensional (or sparse) input signal space to a lower-dimensional (or compressed) feature space.
(ii) Equal dimension representation ($n = L$):
In this representation of features, the dimension of input signal space and feature space needs to be equal.
(iii) Sparse representation ($n < L$):
It is just the reverse of compressed representation where features of training dataset need to be represented from a lower-dimensional input signal space to a higher-dimensional (or sparse) feature space.

The multilayer ELM is considerably faster than deep networks because iterative tuning mechanism is not require in case of ML-ELM and obtained better or similar performance compared to deep networks. It is also known that in ELM, for L hidden nodes and N training examples $(\boldsymbol{x_j}, \boldsymbol{y_j})$, the following Eq. 7 holds:

$$\boldsymbol{g_L}(\boldsymbol{x_j}) = \sum_{\boldsymbol{i=1}}^{\boldsymbol{L}} \boldsymbol{\beta_i} \boldsymbol{g_i}(\boldsymbol{x_j}, \boldsymbol{w_i}, \boldsymbol{b_i}) = \boldsymbol{y_j} \ , \ \boldsymbol{j} = \mathbf{1}, \ldots, N \tag{7}$$

where each symbol has the same meaning as in Eq. (4). In case of ELM-AE, the output weights $\boldsymbol{\beta}$ can be computed using Eq. (8), (9) or (10), and this is different from the computation of $\boldsymbol{\beta}$ in case of ELM.

In order to perform unsupervised learning, few modifications have been done in ELM-AE whose working principle is similar to regular ELM, which are described as follows:

(1) The output data and the input data remain same for every hidden layer. Hence, for every input data $\boldsymbol{X}$:

$$\boldsymbol{Y} = \boldsymbol{X}$$

(2) To improve the performance of ELMs, we need to consider the weights and the biases of the random hidden nodes to be orthogonal and can be represented as follows:

$$\boldsymbol{h} = \boldsymbol{g}(\boldsymbol{w} \cdot \boldsymbol{x} + \boldsymbol{b}), \boldsymbol{w^T} \cdot \boldsymbol{w} = \boldsymbol{I} \ \boldsymbol{and} \ \boldsymbol{b^T} \cdot \boldsymbol{b} = 1$$

(3) the output weight $\boldsymbol{\beta}$ is decided based on the following conditions:

i. if $\boldsymbol{n} > \boldsymbol{L}$ then

$$\boldsymbol{\beta} = \left(\frac{\boldsymbol{I}}{C} + \boldsymbol{H^T H}\right)^{-1} \boldsymbol{H^T X} \tag{8}$$

ii. if $\boldsymbol{n} = \boldsymbol{L}$ then

$$\boldsymbol{\beta} = \boldsymbol{H^{-1} X} \tag{9}$$

iii. if $\boldsymbol{n} < \boldsymbol{L}$ then

$$\boldsymbol{\beta} = \boldsymbol{H^T}\left(\frac{\boldsymbol{I}}{\boldsymbol{C}} + \boldsymbol{HH^T}\right)^{\mathbf{-1}} \boldsymbol{X} \tag{10}$$

where C is a scale parameter which adjusts structural and experiential risk. ELM-AE is used for training the parameters in each layer of ML-ELM. The general equation representing ML-ELM is described as follows:

$$\boldsymbol{H^n} = \boldsymbol{g}((\boldsymbol{\beta^n})^{\boldsymbol{T}} \boldsymbol{H^{n-1}}) \tag{11}$$

For $\boldsymbol{n} = 0$, the 0th hidden layer or the first layer is considered to be the input layer $\boldsymbol{X}$. Equation (11) shows how the transformations of the data take place from layer to layer until it reaches the last but one layer before the final (i.e., output) layer $\boldsymbol{Y}$. The final output matrix $\boldsymbol{Y}$ can be obtained by computing the results between the last hidden layer and the output layer using the regularized least squares technique (Rifkin et al. 2003).

4 Proposed approach

In this section, first we have discussed the architecture of our approach and then summarized the complete approach with the details of algorithms to implement it.

4.1 Architecture of the proposed approach

Given a corpus of classes having text documents, the propose approach involves the following steps:

1. **Preprocessing of text documents of different classes**

   i. Stop words and unwanted words are removed from the text documents of each class from the corpus.
   ii. Other categories need to be ignored, such as verbs, adverbs, adjectives, pronounce. Minipar[3] is used to select nouns as the keywords.
   iii. Now every class in the corpus have preprocessed documents of keywords.

2. **Features score generation with the help of training dataset**

   i. Keywords from preprocessed text document of each class are taken to generate the term–document matrix.
   ii. Separate new documents are made, where each new document represents a particular class in the corpus. A document 'D_{new}' representing a class 'C' is constructed by putting all the preprocessed content (i.e., keywords) of all documents (also known as training dataset) belonging to the class 'C' into the document 'D_{new}'. In other words, a pool of keywords is constructed from all documents of class 'C' and stored in 'D_{new}.' Hence, now we have per class only one new document which consists all keywords of that class or one can say, training set has one instance for each class.
   iii. Now those documents ('D_{new}') are sent as an input to different feature selection techniques (TF-IDF/Chi-square/BNS/IG) as discussed in Sect. 3.1) separately for comparison purpose to generate the scores of each feature (i.e., keyword). Then for each class represented by 'D_{new}', we have a list of keywords in that class along with their corresponding TF-IDF/Chi-square/BNS/IG scores which represent different feature vectors, one for each feature selection technique.

3. **Reduce feature vectors generation by selecting most important keywords for each feature selection (TF-IDF/ Chi-square/BNS/IG) technique**
   Next, we need to select 'n' most important keywords from each 'D_{new},' resulting in a vector of dimensions 'nm,' where 'm' represents the number of predefined classes. In order to obtain 'n' most important keywords from each 'D_{new},' we take into consideration the idea of connected components of graph theory. In graph theory, an undirected graph 'G' is called connected if between any two vertices in the graph there exist a path.[4] If a graph consists of only one vertex, then it is always connected or in other words every individual vertex of a graph is a connected component of that graph. Figure 3 shows three connected component (0-1-2-3-4, 5-6 and 7) of an undirected graph 'G.' In our approach, we consider each keyword as the vertex and the semantic relationship between two keywords forms an edge between them which generates an undirected graph. Here, each connected component will consist of related keywords. A keyword 'a' is related to keyword 'b,' if 'b' is either in the synonym or in the lemma list of 'a.' Each connected component represents keywords of similar context. For example, all synonym and lemma list for 'a' become one connected component. Figure 4 shows the connected component of 'a' where 'b,' 'c,' 'f,' 'h,' 'm,' 'n,' 's' and 'x' are in the synonym and lemma list of 'a.' Table 1 shows some of the synonym lists of certain keywords. For each 'D_{new},' a list of connected components are generated using Wordnet.
   Next, from each connected component of 'D_{new}' the keyword with the highest TF-IDF/Chi-square/BNS/IG score will be selected as the representative keyword (or important keyword) of that component. At the end, a reduced feature vector with 'n' most important keywords will be generated from each 'D_{new}' based on the feature selection technique used (i.e., TF-IDF/Chi-square/BNS/IG). In other way, for every 'D_{new},' four reduced feature vectors with 'n' most important keywords are generated, one for each of the feature selection technique. Details are discussed in Step 3–8 of Sect. 4.2.

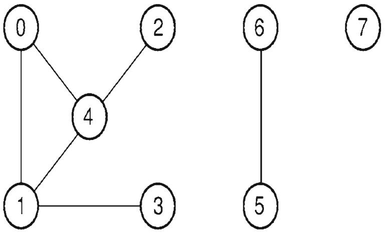

Fig. 3 Connected components of an undirected graph

4. **ELM classification (One-Against-All)**
   For ELM classification, we choose ELM One-Against-All (OAA) scheme after generating the reduced feature vectors in Step 3.
   In this scheme, the number of nodes in the output layer is set as equal to the number of distinct classes. In other words, for each training instance $\mathbf{x}$, m bits are required to represent the target output $\mathbf{y}$ i.e., $(y_1, \ldots, y_m)^T$. Thus, for N training examples (x_n, y_n) of dimension R_n ×

[3] http://ai.stanford.edu/~rion/parsing/minipar_viz.html.

[4] http://computation.pa.msu.edu/NO/ConnCompPresentation.html.

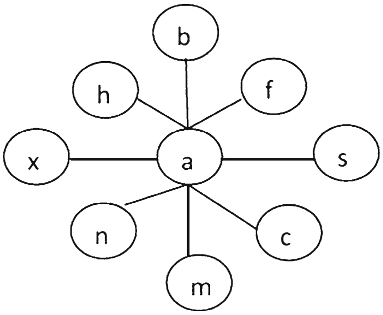

Fig. 4 Connected component of ‘a’

Table 1 Synonym list of certain keywords

Keyword	Synonym list
Amazing	Astonishing, astounding, extraordinary, fabulous, fantastic, improbable, incredible, unbelievable, wonderful
Begin	Ample, broad, gigantic, great, enormous, tall, huge, substantial, immense, vast, gargantuan, tremendous, colossal, large, sizable, grand, mammoth, astronomical, expansive, spacious, stout, titanic, mountainous
Calm	Aloof, composed, quiet, peaceful, still, mild, serene, smooth, tranquil, collected, unruffled, level-headed, unexcited, detached
Describe	Portray, recount, report, characterize, represent, narrate, picture, relate, record
Great	Powerful, worthy, distinguished, noteworthy, remarkable, considerable, grand, much, mighty
Important	Notable, significant, primary, necessary, vital, critical, indispensable, distinguished, valuable, essential, principal, considerable, famous, well-known
Plan	Plot, contrivance, design, scheme, method, map, diagram, draw, procedure, arrangement, intention, way, device, blueprint
Think	Assume, believe, consider, deem, judge, contemplate, mediate, reflect

$\boldsymbol{R_m}$ and $\boldsymbol{L}$ hidden nodes, we randomly assign hidden node parameters $(\boldsymbol{w_i}, \boldsymbol{b_i})$, $\boldsymbol{i} = \mathbf{1}, \ldots, \boldsymbol{L}$ and calculate the output weights $\boldsymbol{\beta}$ and the hidden node output matrix $\boldsymbol{H}.\boldsymbol{\beta} = \boldsymbol{H}^{+}\boldsymbol{Y}$.

Upon receiving the results of $\boldsymbol{m}$ output nodes, the results are submitted to a decision function to find the class label for the instance $\mathbf{x}$, and it is decided by the target output node having the maximum value using the voting mechanism. Mathematically, it can be represented by Eq. 12 as follows:

$$\hat{g}(\mathbf{x}) = \boldsymbol{argmax}_{\boldsymbol{i}=\mathbf{1},\ldots,\boldsymbol{m}} g_{\boldsymbol{i}}(\mathbf{x}) \tag{12}$$

Figure 5 shows the ELM OAA architecture.

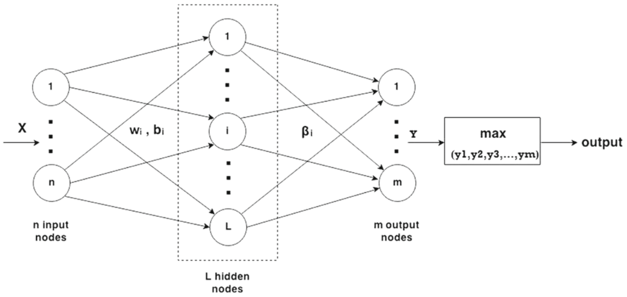

Fig. 5 ELM one-against-all

5. **SVM classification (One-Against-One)**
Similarly for SVM classification, we classify the text documents from the test data using SVM classifier. In order to classify the text documents into different categories, multi-class SVM with one-against-one (OAO) approach is used after the preparation of reduced feature vectors in Step 3. For m classes, $\frac{m(m-1)}{2}$ classifiers are built, one for each pair of classes. The prediction of class for a new document is based on a voting scheme. It involves giving test document as an input to binary SVM’s which classifies the document into one of the two classes and voting up that class. Hence, it constructs $\frac{m(m-1)}{2}$ classifiers, and at the end of comparisons, the test document is categorized into that class which receives the maximum votes.

6. **Usefulness of Wordnet during Feature vector preparation and mapping**
Wordnet[5] developed at the University of Princeton is a thesaurus for the English language. Some of the semantic relations available in Wordnet are synonymy, antonym, hyponymy, etc. Synonyms are words that have similar meanings. A synonym set, or synsets, is a group of synonyms, and the synonyms contained within a synsets are called lemmas. We have made use of Wordnet for feature vector preparation (discussed in Step 3).

We summarized the above steps in a concise manner in Sect. 4.2, and implementation details are discussed in Algorithm 1 and 2.

4.2 Detailed summary of the proposed approach

(1) For each preprocessed class (i.e., ‘$\boldsymbol{D_{new}}$’), separately calculate the score of each keyword in that class using the feature selection techniques (TF-IDF/Chi-square/BNS/IG).
(2) Select a class ‘$\boldsymbol{D_{new}}$’ randomly from the corpus, remove duplicates from its keyword set and store them

[5] http://wordnet.princeton.edu/.

in a list called Keyword_List. Initially, each word of Keyword_List represents an individual connected component.

(3) Using Wordnet, find the synonym list and lemma list of a keyword $\boldsymbol{k}$ (which selected randomly from the Keyword_List of $\boldsymbol{D_{new}}$) and store them in a list called Synonym_Lemma_List of $\boldsymbol{k}$. Now find out common keywords between Synonym_Lemma_List of $\boldsymbol{k}$ and Keyword_List of $\boldsymbol{D_{new}}$. If such common keywords are found, then remove them from Keyword_List of $\boldsymbol{D_{new}}$ and add them to a list called Synonym_Required_List which is the required connected component of $\boldsymbol{k}$.
Add the Synonym_Required_List of $\boldsymbol{k}$ to a list called List_of_List that contains the connected components of all those keywords picked up randomly from the Keyword_List of $\boldsymbol{D_{new}}$.

(4) Repeat Step 3 till the Keyword_List of $\boldsymbol{D_{new}}$ get exhausted. We now have List_of_List of components where each list represents a connected component.

(5) From each component of List_of_List, find the keyword with highest TF-IDF/Chi-square/BNS/IG value and add it to feature list $\boldsymbol{i}$. This feature list represents the feature_vectors of class $\boldsymbol{i}$ ($\boldsymbol{FV}(\boldsymbol{C_i})$) one for each feature selection technique.

(6) Repeat Step 2 to 5 for each class ($\boldsymbol{D_{new}}$) of the corpus.

(7) Now we have feature_vectors for each $\boldsymbol{D_{new_i}}$ ($\boldsymbol{i}$ from 1 to $\boldsymbol{m}$ and $\boldsymbol{m}$ is the number of classes). Remove those keywords from each feature_vector of $\boldsymbol{D_{new_i}}$ which are occurring more than the threshold number of feature_vectors.[6] This decides the maximum number of occurrences of any keyword in all the feature vectors.

(8) Select the top '$\boldsymbol{n}$' keywords from each feature_vector of $\boldsymbol{D_{new_i}}$ which have highest TF-IDF/Chi-square/BNS /IG value. Thus, the words are filtered out according to their priority, importance and semantics. Now we have the reduced feature vectors of $\boldsymbol{D_{new_i}}$ one for each feature selection technique (i.e., TF-IDF/Chi-square/BNS/IG).

(9) Combine all the feature_vectors of each class (separately for each feature selection technique used) to obtain the final unique reduced feature vector ($\boldsymbol{V}$) of size $\boldsymbol{nm}$.

(10) The final reduced feature vector ($\boldsymbol{V}$) is then used for training and testing all the traditional classifiers as follows:

   i. The final reduced feature vector ($\boldsymbol{V}$) is mapped into Multilayer ELM and other conventional classifiers for training purpose.

   ii. Once the model is trained, test data (excluding class label) are passed to ML-ELM and other trained classifiers separately to get predictions of a text instance belongs to which class.

(11) Calculate the precision, recall and F-measure of the target classifier (i.e., ML-ELM and other established classifiers) by using the known class label of a test instance and the output prediction of the target classifier for that test instance.

Algorithm 1: Feature vectors generation of the entire corpus

```
Input: $FC = \{C_1, C_2, \ldots C_m\}$ where each preprocessed class $C_i$ having different TF-IDF/Chi-square/BNS/IG values for every keywords
Output:Feature Vectors $I(C) \leftarrow \phi$ // contains required feature vectors of $FC$
$Keyword\_List(KL) \leftarrow \phi$ // contains all keywords of $C_i$
$Synonym\_Lemma\_List_k(SL_k) \leftarrow \phi$ //contains all synonyms and lemmas of a keyword $k$ found in Wordnet
$Synonym\_Required\_List_k(SRL_k) \leftarrow \phi$ //contains all synonyms and lemmas of a keyword $k$ found both in $SL_k$ and $KL$ which forms the connected component of $k$
$List\_of\_List(LOL) \leftarrow \phi$ // contains all synonyms and lemmas (called the connected component of $k$) of all keywords for each $C_i$
$Feature\_Vector(FV(C_i)) \leftarrow \phi$ // Feature vector of class $C_i$
for all $C_i \in FC$ do
  $KL \leftarrow$ keywords $\in C_i$ //consider keyword as a separate vertex in the graph G
   for all keyword $k$ (selected randomly) $\in KL$ do
     $SL_k \leftarrow k$'s synonyms and lemmas present in Wordnet
     for all $s \in KL$ do
       //$s$ is keyword
       if $s \in SL_k$ then
         $SRL_k = SRL_k \cup \{s\}$ //add $s$ to the synonym_required_list of $k$
         $KL = KL - \{s\}$ //drop $s$ from $KL$
       end if
     end for
     $LOL \leftarrow LOL \cup SRL_k$
     $SRL_w \leftarrow \phi$
     $KL \leftarrow KL - \{k\}$ //drop that keyword $k$ from $KL$
     $(SL_k) \leftarrow \phi$
   end for
   for all $SRL_k \in LOL$ do
     select the keyword $t$ having highest TF-IDF/Chi-square/BNS/IG value from $SRL_k$
     $FV(C_i) \leftarrow FV(C_i) \cup \{t\}$ //append the important keyword into the feature vector of $C_i$
   end for
   $I(C) \leftarrow FV(C_i)$ //append each class feature vectors for different feature selection techniques into the final feature vector
   $FV(C_i) \leftarrow \phi$
end for
return $I(C)$
```

[6] Determined through experiment, iterating the values over a range and considered the value at which best results were obtained.

Algorithm 2: Final reduced feature vector preparation of the corpus

```
Input: Features vectors $I(C)$ resulted by Algorithm 1
Output: Reduced feature vector $V$
$count\_list_k \leftarrow \phi$
$TFL \leftarrow threshold\_feature\_list$ //decided by the experiment. Cutoff on required number of feature vectors which should contain a keyword
$L \leftarrow \phi$
$V \leftarrow \phi$
$Feature\_Vector(FV(C_i)) \leftarrow \phi$
// Feature vector of class $C_i$
for all $FV(C_i) \in I(C)$ do
  for all $k \in FV(C_i)$ do
    $count\_list_k \leftarrow$ number of classes containing keyword $k$ in their feature vector $FV(C_i)$
    if $count\_list_k \geq TFL$ then
      $FV(C_i) \leftarrow FV(C_i) - \{k\}$
      //remove $k$ from feature vector of $FV(C_i)$
    end if
  end for
end for
for all $FV(C_i) \in I(C)$ do
  $L \leftarrow$ select top $n$ keywords which have highest TF-IDF/Chi-square/BNS/IG value from $FV(C_i)$
  $V \leftarrow L$ //append top $n$ keywords of each feature selection technique to $V$
  $L \leftarrow \phi$
end for
return $V$
```

5 Experimental results and discussion

5.1 Experimental setup

In order to demonstrate the performance of our approach, precision, recall and F-measure are calculated. Testing is conducted on two benchmark datasets (DMOZ Open Directory Project[7] and 20-Newsgroups).[8] Python language has been used for implementation of the approach. The algorithm has been run on a machine having 16 Processors - Intel Xeon Processor E5-2690 @ 2.90 GHZ, 64GB RAM running Ubuntu 14.04. We have used four different feature selection techniques (TF-IDF, Chi-square, BNS and IG) for the experimental work and observed their performances on these two datasets using different classifiers. We ran our algorithm extensively on various feature vector lengths used for different classifiers, different number of hidden layer nodes used in ELM and number of hidden layers used for ML-ELM on both datasets. But considered only those length of the feature vector, number of hidden layer nodes and hidden layers for which we obtained the maximum overall F-measure, and are shown in Table 2 and 3 on DMOZ and 20-Newsgroups datasets, respectively. The precision, recall and F-measure of the proposed approach are calculated as follows:

5.1.1 Precision (P)

Precision is the fraction of the documents retrieved by the propose approach which are relevant.

$$P = \frac{(\text{relevant}_{documents}) \cap (\text{retrieved}_{documents})}{\text{retrieved}_{documents}}$$

5.1.2 Recall (R)

Recall is the fraction of the relevant documents which are retrieved by the propose approach.

$$R = \frac{(\text{relevant}_{documents}) \cap (\text{retrieved}_{documents})}{\text{relevant}_{documents}}$$

5.1.3 F-Measure (F)

F-measure[9] is the overall performance measurement of a system which gives equal importance to both precision and recall and can be represented by Eq. 13 as follows:

$$F = 2 * \frac{(P * R)}{(P + R)} \tag{13}$$

The overall values for precision, recall and F-measure of the proposed approach using different classifiers on both the datasets have been calculated using Eqs. 14, 15 and 16, respectively.

$$\text{Overall precision} = \frac{\sum_{i=1}^{n} (p_i . d_i)}{\text{Total no. of test documents}} \tag{14}$$

$$\text{Overall recall} = \frac{\sum_{i=1}^{n} (r_i . d_i)}{\text{Total no. of test documents}} \tag{15}$$

$$\text{Overall F-measure} = \frac{\sum_{i=1}^{n} (f_i . d_i)}{\text{Total no. of test documents}} \tag{16}$$

where p_i, r_i, f_i and d_i are the precision, recall, F-measure and the number of testing documents of the ith category, respectively, and n is the number of categories of the dataset.

5.2 DMOZ dataset

DMOZ is an Open Directory Project which consists of 14 categories of Web pages. For our work, we considered 69,068 documents out of which 38,000 documents are used for training and 31,068 documents are used for testing purposes. We

[7] http://www.dmoz.org.

[8] http://qwone.com/~jason/20Newsgroups/.

[9] http://www.gabormelli.com/RKB/F-Measure.

Table 2 Different parameters used for DMOZ dataset

Feature selection technique	Feature vector length	Hidden nodes	Hidden layers
TF-IDF	1858	2000	5
Chi-square	1408	1800	5
BNS	1789	1800	5
IG	1240	1600	5

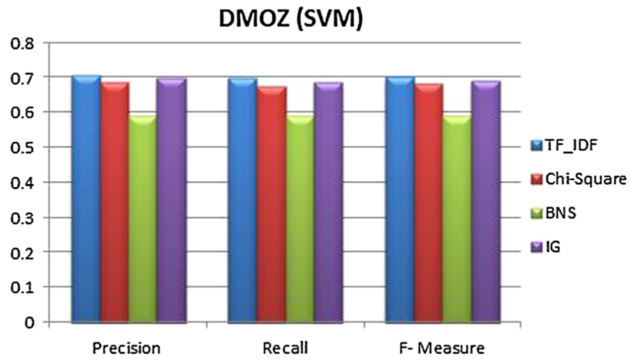

Fig. 6 Performance measurements of different feature selection techniques using SVM

observed the overall F-measure on DMOZ dataset for different classifiers using the four traditional feature selection techniques as follows:

(1) It has been observed from Fig. 6 that LinearSVC using 'TF-IDF' feature selection technique generates good results as can be evident from the overall F-measure of 0.7035 with a feature vector length of 1858. 'IG' is also on a par with 'TF-IDF' having overall F-measure value of 0.6919 with a feature vectors length of 1240 followed by Chi-square with overall F-measure of 0.6820. 'BNS' has shown poor performance having overall F-measure of 0.59. Category-wise performance of LinearSVC for TF-IDF in which it achieved the maximum overall F-measure is given in Table 4 for demonstration purpose.

(2) When ELM has been used as a classifier (shown in Fig. 7), it has been observed that 'TF-IDF' selection techniques have maximum overall F-measure of 0.7055 followed by 'IG' and 'Chi-square' with overall F-measure of 0.6898 and 0.6824, respectively. However, the results when 'BNS' has been used as the selection technique are not impressive as the overall F-measure is 0.5996. In order to demonstrate the category-wise performance of ELM (shown in Table 5), we have shown only TF-IDF feature selection technique in which it achieved the maximum overall F-measure.

(3) Similarly, when ML-ELM has been used as a classifier (shown in Fig. 8), the results get improved, achieving an overall F-Measure of 0.7228 using 'TF-IDF' as feature selection technique followed by IG having 0.7147. The number of hidden nodes is same as used in ELM. But again result of 'BNS' is not impressive, which is 0.6068. Performances of ML-ELM on different categories using different feature selection techniques are given in Table 6, 7, 8 and 9, respectively.

From the above results, it is concluded that using ML-ELM as the classifier, the results obtained are better compared to SVM and ELM techniques. For ML-ELM classifier, the number of hidden layer nodes is set more than the length of the input feature vector (i.e., $\boldsymbol{n} < \boldsymbol{L}$, as discussed in Sect. 3.3.1) in order to represent the training feature set from a lower-dimensional input space to a higher-dimensional (or sparse) feature space. This in turn generated a good performance of ML-ELM on this dataset. Although maximum overall F-measure is achieved by ML-ELM using 'TF-IDF' as the feature selection technique but 'IG' is on a par with 'TF-IDF' in most of the cases. Generalization capability of ELM on different feature selection techniques is either better or almost similar to SVM on this dataset, and the reason may be due the large training dataset which possibly generates overfitting during the training process of ELM.

5.2.1 *Comparison with existing approaches on DMOZ dataset*

1. *Proposed approach versus Heung-Seon Oh et al.*
   We have compared our work with Oh et al. (2011) who have proposed an algorithm consisting of two stages: search and classification. In the search stage, using a search technique, they retrieved several candidate categories from entire hierarchy that are more similar to the input document. Then they collected training data for

Table 3 Different parameters used on 20-Newsgroups Dataset

Feature selection technique	Feature vector length	Hidden nodes	Hidden layers
TF-IDF	2260	2500	3
Chi-square	1510	1600	3
BNS	1756	2000	3
IG	1462	1600	3

Table 4 DMOZ dataset using SVM and 1858 feature vector length for TF-IDF

Category	No. of test documents	Precision	Recall	*F*-Measure
Arts	1396	0.7097	0.7072	0.7084
Business	3384	0.7099	0.7015	0.7057
Computers	1494	0.7102	0.6998	0.7050
Games	5757	0.7105	0.7025	0.7065
Health	1491	0.7107	0.6893	0.6998
Homes	1405	0.6825	0.6626	0.6724
News	1504	0.7113	0.6775	0.6940
Recreation	1410	0.6898	0.6991	0.6944
Reference	1301	0.7118	0.6882	0.6998
Regional	1307	0.6814	0.6945	0.6879
Science	1390	0.7123	0.7337	0.7228
Shopping	6209	0.7126	0.7061	0.7093
Society	1505	0.7128	0.7074	0.7101
Sports	1515	0.7131	0.6894	0.7011
Overall	31068	0.7078	0.6994	0.7035

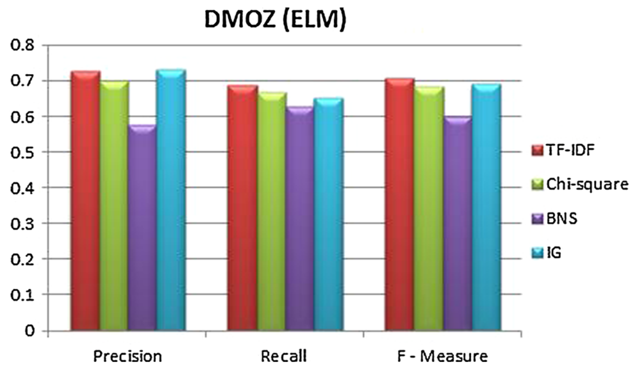

Fig. 7 Performance measurements of different feature selection techniques using ELM

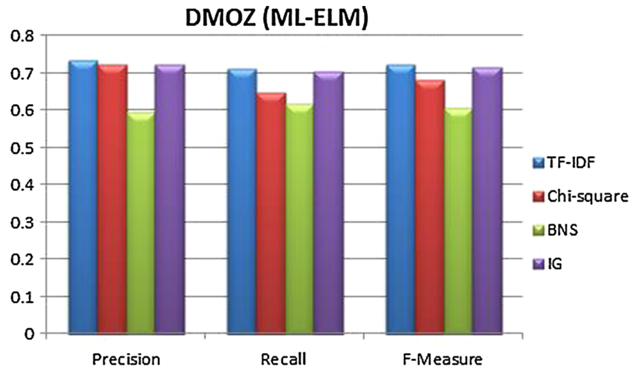

Fig. 8 Performance measurements of different feature selection techniques using ML-ELM

Table 5 DMOZ dataset using ELM and 1858 feature vector length for TF-IDF

Category	No. of test documents	Precision	Recall	*F*-Measure
Arts	1396	0.7165	0.6930	0.7046
Business	3384	0.7395	0.6834	0.7103
Computers	1494	0.6859	0.7149	0.7001
Games	5757	0.7406	0.6960	0.7176
Health	1491	0.7294	0.6633	0.6948
Homes	1405	0.6883	0.7048	0.6965
News	1504	0.7169	0.6723	0.6939
Recreation	1410	0.7661	0.6634	0.7111
Reference	1301	0.6983	0.6590	0.6781
Regional	1307	0.7476	0.6456	0.6929
Science	1390	0.7626	0.6823	0.7202
Shopping	6209	0.7165	0.7060	0.7112
Society	1505	0.7156	0.6654	0.6896
Sports	1515	0.7186	0.6709	0.6939
Overall	31068	0.7263	0.6866	0.7055

Table 6 DMOZ dataset using ML-ELM and 1858 feature vector length for TF-IDF

Category	No. of test documents	Precision	Recall	*F*-Measure
Arts	1396	0.7340	0.7029	0.7181
Business	3384	0.7342	0.7025	0.7180
Computers	1494	0.7345	0.7114	0.7228
Games	5757	0.7348	0.7002	0.7171
Health	1491	0.7350	0.6910	0.7123
Homes	1405	0.7353	0.6880	0.7109
News	1504	0.7356	0.6820	0.7078
Recreation	1410	0.7358	0.7245	0.7301
Reference	1301	0.7361	0.7035	0.7194
Regional	1307	0.7363	0.7199	0.7280
Science	1390	0.7366	0.7225	0.7295
Shopping	6209	0.7369	0.7315	0.7342
Society	1505	0.7371	0.7198	0.7284
Sports	1515	0.7374	0.7204	0.7288
Overall	31068	0.7356	0.7105	0.7228

each candidate from the documents associated with the candidate category (local information) and from top-level categories (global information). Their proposed methods for determining the mixture weights are applied to each category node in modulating the relative contributions of local and global models. The model was tested on DMOZ dataset, and the maximum average F-measure they achieved is 0.3773 which is lesser than our approach.

2. *Proposed approach versus Gui-Rong Xue et al.*
Xue et al. (2008) have suggested a two-stage approach for large-scale hierarchical classification called deep classi-

Table 7 DMOZ dataset using ML-ELM and 1408 feature vector length for Chi-square

Category	No. of test documents	Precision	Recall	F-Measure
Arts	1396	0.6994	0.6523	0.6750
Business	3384	0.6845	0.6331	0.6578
Computers	1494	0.7382	0.6324	0.6812
Games	5757	0.7385	0.7245	0.7314
Health	1491	0.7387	0.6310	0.6807
Homes	1405	0.7215	0.6304	0.6729
News	1504	0.6815	0.6297	0.6546
Recreation	1410	0.7395	0.6290	0.6798
Reference	1301	0.7398	0.6283	0.6795
Regional	1307	0.6945	0.6277	0.6594
Science	1390	0.7403	0.6270	0.6790
Shopping	6209	0.7245	0.6263	0.6718
Society	1505	0.7409	0.6257	0.6784
Sports	1515	0.7411	0.6250	0.6781
Overall	31068	0.7231	0.6475	0.6832

Table 8 DMOZ dataset using ML-ELM and 1789 feature vector length for BNS

Category	No. of test documents	Precision	Recall	F-Measure
Arts	1396	0.5964	0.6334	0.6143
Business	3384	0.5789	0.6165	0.5971
Computers	1494	0.5867	0.5753	0.5809
Games	5757	0.5897	0.6307	0.6095
Health	1491	0.5839	0.6091	0.5962
Homes	1405	0.6162	0.5850	0.6002
News	1504	0.6263	0.5875	0.6063
Recreation	1410	0.6266	0.6754	0.6501
Reference	1301	0.5976	0.5892	0.5934
Regional	1307	0.6345	0.6172	0.6257
Science	1390	0.5873	0.6130	0.5999
Shopping	6209	0.5993	0.6260	0.6124
Society	1505	0.5860	0.6289	0.6067
Sports	1515	0.5654	0.6205	0.5917
Overall	31068	0.5957	0.6188	0.6068

Table 9 DMOZ dataset using ML-ELM and 1240 feature vector length for IG

Category	No. of test documents	Precision	Recall	F-Measure
Arts	1396	0.7217	0.7005	0.7109
Business	3384	0.7219	0.7145	0.7182
Computers	1494	0.7222	0.7128	0.7175
Games	5757	0.7225	0.6878	0.7047
Health	1491	0.7227	0.7023	0.7124
Homes	1405	0.7230	0.6756	0.6985
News	1504	0.7233	0.6905	0.7065
Recreation	1410	0.7235	0.7121	0.7178
Reference	1301	0.7238	0.7012	0.7123
Regional	1307	0.7240	0.7075	0.7157
Science	1390	0.7243	0.7467	0.7353
Shopping	6209	0.7246	0.7191	0.7218
Society	1505	0.7248	0.7204	0.7226
Sports	1515	0.7251	0.7024	0.7136
Overall	31068	0.7233	0.7064	0.7147

fication. In the first stage, they organized the hierarchy of text into flat categories where a search process is conducted on large-scale hierarchies by retrieving the related categories for a given document. Then they ranked the categories and select the most useful categories. In the second stage, to classify the given document on a given small subset, they trained a classification model on that small subset of original hierarchy. They evaluated their deep classification approach on the Open Directory Project with an F-measure of 0.5180, which is lower compared to the maximum overall F-measure of 0.7228 of our approach.

3. *Proposed approach versus Siddharth Gopal et al.* Finally, we compared our work with Gopal and Yang (2013) where they have developed a recursive regularization framework along with a scalable optimization algorithm for large-scale hierarchical classification with hierarchical and graphical dependencies between the class labels. They developed two different variants of their framework using the logistic-loss function and the hinge-loss function. They have used multiple benchmark datasets including DMOZ for experimental purpose and achieved a consistent results. An F-measure of 0.5717 has been achieved while using DMOZ dataset which is significantly lesser than the maximum overall F-measure obtained in our approach.

The comparison results are given in Table 6. Our work with ML-ELM as the classifier acquired an impressive overall F-measure of 0.7228 using 'TF-IDF' feature selection technique on DMOZ dataset which justified the significance of our approach compared to the above existing approaches (Table 10).

5.3 20-Newsgroups dataset

The 20-Newsgroups is one of the most popular datasets used for text classification and has 20 different newsgroups. The Web documents in it are categorized into 7 categories. For

Table 10 Comparison of results with different approaches

Different approaches	Dataset used	*F*-Measure (%)
Heung Seonoh et al.	DMOZ	37.73
Gui-Rong Xue et al.	DMOZ	51.80
Siddharth Gopal et al.	DMOZ	57.17
Our approach (SVM)	DMOZ	70.35
Our approach (ELM)	DMOZ	70.55
Our approach (ML-ELM)	DMOZ	**72.28**

Bold value obtained the highest F-measure compared to other approaches

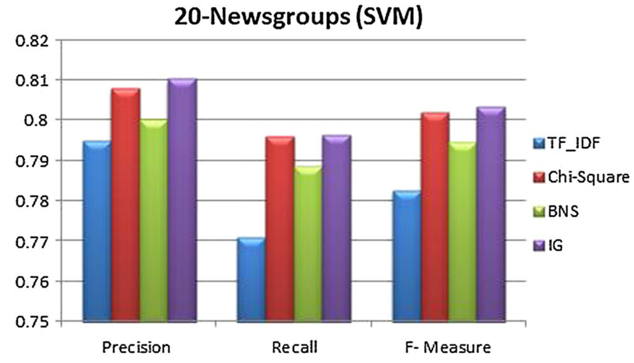

Fig. 9 Performance measurements of different feature selection techniques using SVM

Table 11 20-Newsgroups dataset using SVM and 1462 feature vector length for IG

Category	No. of test documents	Precision	Recall	*F*-Measure
Alt	320	0.8343	0.7914	0.8123
Computers	1952	0.8354	0.7889	0.8115
Miscellaneous	390	0.8309	0.8097	0.8202
Recreation	1590	0.8032	0.7984	0.8008
Science	1580	0.8034	0.8064	0.8049
Social	399	0.7949	0.7945	0.7947
Talk	1297	0.7832	0.7912	0.7872
Overall	7528	0.8104	0.7965	0.8033

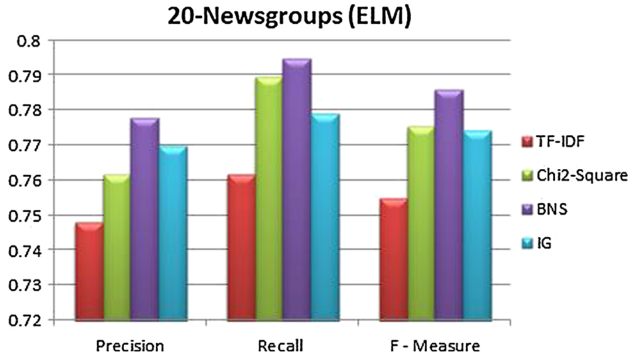

Fig. 10 Performance measurements of different feature selection techniques using ELM

experimental purpose, 18,846 documents have been considered out of which 11,318 documents are used for training purpose and 7528 documents are used for testing purpose.

We observed the following F-measure on 20-Newsgroups dataset for different classifiers with different existing feature selection techniques:

(1) It has been found that LinearSVC using 'IG' as the feature selection technique generates decent result as can be evident from the overall F-measure of 0.8033 with a feature vector length of 1462 (shown in Fig. 9), which is followed by Chi-square having F-measure of 0.8020 with feature vector length of 1510 and then by BNS and TF-IDF having overall F-measure of 0.7945 (feature vector length of 1756) and 0.7827 (feature vector length of 2260), respectively. Table 11 demonstrates the performance of SVM on each category of 20-Newsgroups using 'IG' as the feature selection technique for which it achieved the maximum overall F-measure.

(2) When ELM has been used as a classifier (shown in Fig. 10), it is observed that 'BNS' feature selection technique generates an overall F-measure of 0.7861 followed by 'Chi-square' and 'IG' having overall F-measure of 0.7754 and 0.7744, respectively. ELM using TF-IDF has the lowest performance compared to all other feature selection techniques. Table 12 shows the category-wise performance of ELM using 'BNS' technique in which it achieved the highest overall F-measure.

(3) Similarly, when ML-ELM with hidden layer = 3 has been used as a classifier (shown in Fig. 11), the results improved achieving an F-measure of 0.8153 using 'BNS' as feature selection technique followed by 'Chi-square' and 'IG' having overall F-measure of 0.8139 and 0.8134, respectively. Table 13, 14, 15 and 16 show the category-wise performances of ML-ELM using TF-IDF, Chi-square, BNS and IG, respectively.

From the above results, it is concluded that using ML-ELM as the classifier, the results obtained are better compared to SVM and ELM techniques. The number of hidden layer nodes is set more than the length of the input feature vector (i.e., n $<$ L, as discussed in Sect. 3.3.1) in order to represent the training feature set in a higher-dimensional (or sparse) feature space. This in turn generated good performance of ML-ELM on this dataset. ML-ELM using BNS feature selection technique achieved the highest F-measure compared to other feature selection techniques. SVM showed decent performance compared to ELM in all feature selection techniques on this dataset.

Table 12 20-Newsgroups dataset using ELM and 1756 feature vector length for BNS

Category	No. of test documents	Precision	Recall	*F*-Measure
Alt	320	0.7489	73.08	0.7397
Computers	1952	0.7357	0.8045	0.7686
Miscellaneous	390	0.7775	0.7999	0.7885
Recreation	1590	0.7930	0.7753	0.7841
Science	1580	0.7987	0.8255	0.8119
Social	399	0.7889	0.7687	0.7787
Talk	1297	0.8003	0.7885	0.7944
Overall	7528	0.7777	0.7947	0.7861

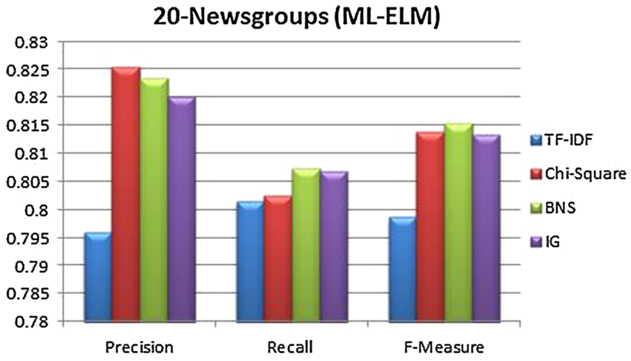

Fig. 11 Performance measurements of different feature selection techniques using ML-ELM

Table 13 20-Newsgroups dataset using ML-ELM and 2260 feature vector length for TF-IDF

Category	No. of test documents	Precision	Recall	*F*-Measure
Alt	320	0.8055	0.8000	0.8027
Computers	1952	0.7875	0.8045	0.7959
Miscellaneous	390	0.7894	0.8256	0.8071
Recreation	1590	0.8199	0.7926	0.8060
Science	1580	0.7547	0.7868	0.7704
Social	399	0.8246	0.8040	0.8142
Talk	1297	0.8214	0.8179	0.8196
Overall	7528	0.7961	0.8015	0.7987

Table 14 20-Newsgroups dataset using ML-ELM and 1510 feature vector length for Chi-square

Category	No. of test documents	Precision	Recall	*F*-Measure
Alt	320	0.8456	0.7996	0.8220
Computers	1952	0.8575	0.7888	0.8217
Miscellaneous	390	0.8647	0.8179	0.8406
Recreation	1590	0.8145	0.8334	0.8238
Science	1580	0.8147	0.7998	0.8072
Social	399	0.8040	0.7886	0.7962
Talk	1297	0.7945	0.7890	0.7917
Overall	7528	0.8256	0.8025	0.8139

Table 15 20-Newsgroups dataset using ML-ELM and 1756 feature vector length for BNS

Category	No. of test documents	Precision	Recall	*F*-Measure
Alt	320	0.8479	0.8035	0.8251
Computers	1952	0.8490	0.8010	0.8243
Miscellaneous	390	0.8445	0.8218	0.8330
Recreation	1590	0.8168	0.8373	0.8269
Science	1580	0.8170	0.8007	0.8088
Social	399	0.7998	0.7825	0.7911
Talk	1297	0.7968	0.7929	0.7948
Overall	7528	0.8236	0.8074	0.8153

Table 16 20-Newsgroups dataset using ML-ELM and 1462 feature vector length for IG

Category	No. of test documents	Precision	Recall	*F*-Measure
Alt	320	0.8440	0.7993	0.8210
Computers	1952	0.8451	0.7968	0.8202
Miscellaneous	390	0.8406	0.8176	0.8289
Recreation	1590	0.8129	0.8062	0.8095
Science	1580	0.8131	0.8142	0.8137
Social	399	0.8046	0.8286	0.8164
Talk	1297	0.7929	0.8056	0.7992
Overall	7528	0.8202	0.8068	0.8134

5.3.1 Comparison with existing approaches on 20-Newsgroups dataset

1. *Proposed approach versus Zhang et al.*
   The obtained results are compared with Zhang et al. (2009) where they have suggested Fuzzy kNN algorithm to classify the Web pages, and preparation of feature vector is done using simple TF-IDF approach. They have tested their approach on 20-Newsgroups dataset and from Table 2 listed in Zhang et al. (2009), we observed that their F-Measure of Fuzzy kNN approach is 0.7638 when the feature vector size is 2500. On the other hand, our approach obtained maximum overall F-measure of 0.8153 using ML-ELM as the classifier.
2. *Proposed approach versus Gongde Guo et al.*
   Guo et al. (2003) have used a simple kNN model-based approach to classify Web pages into different categories. From Table 3 of the Guo et al. (2003), it is evident that their approach has obtained an F-measure of 0.8079 on

Table 17 Comparison of results with different approaches

Different approaches	Dataset used	F-Measure (%)
Zhang et al.	20-Newsgroups	76.38
Gongde Guo et al.	20-Newsgroups	80.79
Weimao Ke et al.	20-Newsgroups	60.30
Yangqiu Song et al.	20-Newsgroups	68.20
Nguyen Cao Truong Hai et al.	20-Newgroups	64.42
Our Approach (SVM)	20-Newsgroups	80.33
Our Approach (ELM)	20-Newsgroups	78.61
Our Approach (ML-ELM)	20-Newsgroups	**81.53**

Bold value obtained the highest F-measure compared to other approaches

the 20-Newsgroups dataset which is lesser than what we achieve using ML-ELM as the classifier.

3. *Proposed approach versus Weimao Ke et al.*
   Ke (2012) have discussed the least information theory (LIT) to quantify meaning of information in probability distributions and derived a new document representation for text classification. LIT offers an information centric approach to weight terms based on probability distributions in the documents versus in the collection. They suggested two term weight quantities in the context of document classification [least information binary (LIB) and least information frequency (LIF)]. They have shown that LIB*LIF weighting scheme outperforms TF*IDF in several experimental settings. For experimental work, 20-Newsgroups dataset has been used and they have claimed an F-measure of 0.6030 which is lower than our results.
4. *Proposed approach versus Yangqiu Song et al.*
   A dataless hierarchical classification approach has introduced by Song and Roth (2014). Their scheme is composed of two steps: a bootstrapping step and semantic similarity step. In the bootstrapping step, they adapt to the specific document collection. In the semantic similarity step, to compute meaningful semantic similarity between a document and a potential label, they embedded both labels and documents in a semantic space. They have justified that their algorithm is competitive with supervised classification algorithms. An F-measure of 0.6820 has been achieved by them using 20-Newsgroups dataset, which is significantly lesser than what we obtained using ML-ELM as the classifier.
5. *Proposed approach versus Nguyen Cao Truong Hai et al.*
   Im Kim and Park (2009) have combined SVD with LDA for text classification. In their method, they first applied SVD on input data which convert it into a rank-specified reduced space and latter they applied LDA on this reduced space for classifying the text. They have used 20-Newsgroups dataset for experimental purpose and achieved an F-measure of 0.6442. It is less than the overall F-measure of 0.8153 received by our approach using ML-ELM.

The comparison results are given in Table 17. The proposed approach with ML-ELM as the classifier obtained an impressive overall F-measure of 0.8153 using ‘BNS’ as the feature selection technique on 20-Newsgroups dataset which signifies the importance of our approach compared to the above existing approaches.

5.4 Comparisons with other traditional classifiers

The overall F-measures of different classifiers using different feature selection techniques on both datasets are given in Table 18. It is observed from the experimental results that by

Table 18 Comparisons of ML-ELM with other state-of-the-art classifiers

Classifier	20- Newsgroups (F-Measure-%)				DMOZ (F-Measure-%)			
	TF-IDF	Chi-square	BNS	IG	TF-IDF	Chi-square	BNS	IG
ELM	75.48	77.54	78.61	77.44	70.55	68.24	59.96	68.98
ML-ELM	**79.87**	**81.39**	**81.53**	**81.34**	**72.28**	**68.32**	**60.68**	**71.47**
SVM (LinearSVC)	78.27	80.20	79.45	80.33	70.35	68.20	59.00	69.19
SVM (Linear kernel)	65.73	72.57	72.00	63.48	67.13	62.39	54.11	64.35
K-Nearest Neighbor ($K = 5$)	53.29	53.45	55.28	48.64	46.38	41.72	38.62	35.45
Multinomial Naive Bayes	71.39	69.50	72.89	67.76	58.95	54.34	54.67	53.40
Gaussian Naive Bayes	55.24	58.53	58.62	59.33	43.56	41.35	36.38	42.63
Bernoulli Naive Bayes	67.45	68.31	69.37	64.25	52.35	48.27	51.66	47.47
Decision trees	58.21	58.23	60.40	57.89	46.87	42.67	42.45	48.85
Random forest (10 classifiers)	68.35	70.46	73.54	69.25	57.34	58.23	54.37	59.85
Extra trees (10 classifiers)	69.27	70.85	73.51	70.64	58.46	56.72	52.33	57.65
Gradient boosting (10 classifiers)	64.58	66.32	68.45	67.65	57.48	57.53	53.77	59.82

Bold values obtained the highest F-measure for each feature selection technique compared to other traditional classifiers representing the training dataset of ML-ELM on both datasets in a higher-dimensional feature space, generates highest F-measure compared to all other traditional classifiers which justified the prominence of deep learning for classifying the text data. From the Table 18, it can be observed that in DMOZ dataset, ML-ELM dominated all other classifiers for all feature selection techniques. Similarly in 20-Newsgroups dataset for all feature selection techniques, ML-ELM dominates all other classifiers with highest overall F-measure of 0.8153.

6 Conclusion

The paper presented an efficient approach for classifying the text data using ML-ELM and other established classifiers. The proposed approach selects the best features (using connected component technique and Wordnet) for preparing an effective training dataset. ML-ELM yields very good results which demonstrates the efficiency of our approach compared to other existing approaches. Various feature selection techniques have been combined individually with connected component to prepare a good feature vector. The experimental results on DMOZ and 20-Newsgroups datasets using SVM, ELM and ML-ELM classifiers are concluded as follows:

(1) Using LinearSVC, SVM gives maximum overall F-measure of 0.7035 on DMOZ dataset with 'TF-IDF' as the feature selection technique and overall F-measure of 0.8033 on 20-Newsgroups dataset with 'IG' as the feature selection technique.
(2) Similarly, using ELM as classifier gives maximum overall F-measure of 0.7055 on DMOZ dataset with 'TF-IDF' as the feature selection technique and overall F-measure of 0.7861 using 20-Newsgroups dataset with 'BNS' as the feature selection technique.
(3) Using ML-ELM as classifier gives maximum overall F-measure of 0.7228 on DMOZ dataset with 'TF-IDF' as the feature selection technique and overall F-measure of 0.8153 on 20-Newsgroups dataset with 'BNS' as the feature selection technique.
(4) The possible reasons for ML-ELM to shows better performance on both datasets compared to other state-of-the-art classifiers can be summarized as follows:

    i. ability of representing the training feature set in a higher- or sparse-dimensional feature space by making nodes of the hidden layer more than the nodes of the input layer.
    ii. layer-wise unsupervised training is set for the weighting parameters of each hidden layer.
    iii. in the deep architecture of ML-ELM, the presence of multiple layers gives multiple nonlinear transformation of the input data which in turn able to generate better representation learning.
    iv. the nature of the deep architecture of ML-ELM can capture higher-level abstraction and every layers in the network can learn a distinct form of the input by performing representation learning using unsupervised learning technique.

(5) The large training dataset of DMOZ compared to 20-Newsgroups yields similar or better results of ELM than SVM possibly due to the occurrences of overfitting during the training phase of ELM. But in 20-Newsgroups dataset, performance of SVM is better than ELM. This indicates that ELM has lower generalization ability compared to SVM when the training dataset is small but for large training dataset, ELM has similar or better performance than SVM.

Our approach with the promising results witnessed the suitability and importance of ML-ELM in the field of text classification compared to other state-of-the-art classifiers. This shows that deep learning has high impact for classification of text data. Toward future work, this approach can be implemented in a distributed environment which will consume less processing time and will help in load balancing. We believe that by combining the feature space of ML-ELM with other state-of-the-art classifiers will further strengthen the results of text classification.

Compliance with ethical standards

Human and animal rights This article does not contain any studies with human participants or animals performed by any of the authors.

Conflict of interest None.

References

Bai Z, Huang G-B, Wang D, Wang H, Westover MB (2014) Sparse extreme learning machine for classification. IEEE Trans Cybern 44(10):1858–1870

Chen R-C, Hsieh C-H (2006) Web page classification based on a support vector machine using a weighted vote schema. Expert Syst Appl 31(2):427–435

Ding S, Xu X, Nie R (2014) Extreme learning machine and its applications. Neural Comput Appl 25(3–4):549–556

Ding S, Zhang N, Xu X, Guo L, Zhang J (2015) Deep extreme learning machine and its application in EEG classification. Math Probl Eng 2015

Forman G (2003) An extensive empirical study of feature selection metrics for text classification. J Mach Learn Res 3:1289–1305

Gomez JC, Moens M-F (2012) Hierarchical classification of web documents by stratified discriminant analysis. In: Multidisciplinary information retrieval. Springer, pp 94–108

Gopal S, Yang Y (2013) Recursive regularization for large-scale classification with hierarchical and graphical dependencies. In: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp 257–265

Guo G, Wang H, Bell D, Bi Y, Greer K (2003) Knn model-based approach in classification. In: On the move to meaningful internet systems, (2003) CoopIS, DOA, and ODBASE. Springer, pp 986–996

Hinton GE, Osindero S, Teh Y-W (2006) A fast learning algorithm for deep belief nets. Neural Comput 18(7):1527–1554

Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. Science 313(5786):504–507

Huang G-B (2003) Learning capability and storage capacity of two-hidden-layer feedforward networks. IEEE Trans Neural Netw 14(2):274–281

Huang G-B, Zhu Q-Y, Siew C-K (2006) Extreme learning machine: theory and applications. Neurocomputing 70(1):489–501

Huang G-B, Zhou H, Ding X, Zhang R (2012) Extreme learning machine for regression and multiclass classification. IEEE Trans Syst Man Cybern Part B Cybern 42:513–529

Im Kim K, Park HR (2009) Svd-lda: a combined model for text classification. J Inf Process Syst 5(1):5–10

Kasun LLC, Zhou H, Huang G-B, Vong CM (2013) Representational learning with extreme learning machine for big data. IEEE Intell Syst 28(6):31–34

Ke W (2012) Least information document representation for automated text classification. Proc Am Soc Inf Sci Technol 49(1):1–10

Klassen M, Paturi N (2010) Web document classification by keywords using random forests. In: Networked digital technologies. Springer, pp 256–261

Liang N-Y, Huang G-B, Saratchandran P, Sundararajan N (2006) A fast and accurate online sequential learning algorithm for feedforward networks. IEEE Trans Neural Netw 17(6):1411–1423

Lingras P, Butz C (2007) Rough set based 1-v-1 and 1-vr approaches to support vector machine multi-classification. Inf Sci 177(18):3782–3798

Li L, Song D, Liao L (2012) Vertical classification of web pages for structured data extraction. In: Information retrieval technology. Springer, pp 486–495

Liu X, Gao C, Li P (2012) A comparative analysis of support vector machines and extreme learning machines. Neural Netw 33:58–66

Manning CD, Raghavan P, Schütze H et al (2008) Introduction to information retrieval, vol 1. Cambridge university press, Cambridge

Mirza B, Kok S, Dong F (2016) Multi-layer online sequential extreme learning machine for image classification. In: Proceedings of ELM-2015 vol 1. Springer, pp 39–49

Oh H-S, Choi Y, Myaeng S-H (2011) Text classification for a large-scale taxonomy using dynamically mixed local and global models for a node. In: Advances in information retrieval. Springer, pp 7–18

Rifkin R, Yeo G, Poggio T (2003) Regularized least-squares classification. Nato Sci Ser Sub Ser III Comput Syst Sci 190:131–154

Rujiang B, Xiaoyue W, Zewen H (2011) A novel web pages classification model based on integrated ontology. In: Software engineering, business continuity, and education. Springer, pp 1–10

Song Y, Roth D (2014) On dataless hierarchical text classification. In: AAAI. pp 1579–1585

Tang J, Deng C, Huang G-B, Hou J (2014) A fast learning algorithm for multi-layer extreme learning machine. In: International Conference on IEEE Image Processing (ICIP). pp 175–178

Wan CH, Lee LH, Rajkumar R, Isa D (2012) A hybrid text classification approach with low dependency on parameter by integrating k-nearest neighbor and support vector machine. Expert Syst Appl 39(15):11 880–11 888

Xue G-R, Xing D, Yang Q, Yu Y (2008) Deep classification in large-scale text hierarchies. In: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval. ACM, pp 619–626

Yang Y, Pedersen JO (1997) A comparative study on feature selection in text categorization. ICML 97:412–420

Yang Y, Wu Q (2015) Multilayer extreme learning machine with subnetwork nodes for representation learning. IEEE Trans Cybern 99:1–14

Zhang J, Niu Y, Nie H (2009) Web document classification based on fuzzy k-nn algorithm," In: CIS'09. International conference on computational intelligence and security, 2009, vol 1. IEEE, pp 193–196