Sādhanā (2020)45:123
https://doi.org/10.1007/s12046-020-01342-0

Chronological-brain storm optimization based support vector neural network for sentiment classification using map reduce framework

M POONGOTHAI[1,*] and M SANGEETHA[2]

[1]Institute of Road and Transport Technology, Erode, India
[2]Coimbatore Institute of Technology, Coimbatore, India
e-mail: poongothaim31@gmail.com

MS received 20 November 2019; revised 9 March 2020; accepted 11 March 2020

Abstract. Sentiment classification plays a dominant role in day-to-day life including the political events, production areas, and commercial activities. The need for the accurate and instant classification of the user emotions is a hectic task to be solved. The traditional methods fail to address the classification of dynamic data and huge volumes of documents. Moreover, to assure the classification accuracy and deal with huge volumes of data, the proposed method employs the MapReduce Framework. The proposed sentiment analysis involves two processes, such as feature extraction and classification that is performed in the MapReduce framework using the mapper and reducer functions. The feature vector is based on the sentence-specific features, SentiWordNet-based features, and statistical features corresponding to the individual reviews that are classified as positive and negative reviews using the proposed Chronological-Brain Storm Optimization based Support Vector Neural Network (CBSO-SVNN). The analysis is progressed using four datasets obtained from the movie review database that confirms that the proposed method outperformed the existing methods in terms of the accuracy, sensitivity, and specificity. The accuracy, specificity, and sensitivity of the proposed method are 0.8714, 0.9027, and 0.8714, respectively.

Keywords. Brain storm optimization; support vector neural network; chronological; MapReduce framework; sentiment classification.

1. Introduction

Big data analysis gains remarkable interest in Computer Science and it defines the king-size of data that progress freely in the social media, web, remote sensing data, medical records, and so on which can be structured, unstructured or semi-structured [1]. One of the examples for big data is social media and the idea behind the social media is to ensure the communication and information sharing throughout the world with the aim to connect people of similar communal interest towards achieving cost-effective information sharing among them. Social media depends on multiple principle sources and receivers in order to assure better reachability and usability when compared with the traditional media that operates based on the concept of single source and multiple users. In addition, it offers a raised area for online shopping, e-commerce, and so on in order to widespread the business all over the world [2]. On the other hand, the political parties use the social media for organizing their campaigns and communicate their views and ideas to the society. Websites such as Facebook and Twitter [3] offer a constant option of sharing and re-tweeting digital images, texts, and videos to keep sprouting effectively. The social media imposes highly bad impacts on the society in the form of virility, harmful comments, and Cyber bullying. The demerit is that the contents established at these sites may impose a tedious risk on the government that imposed the inability to watch these digital contents traversing at these sites [2]. To analyze these contents and extract the useful information out of it, for any purpose, sentiment analysis plays a significant role [4].

Sentiment analysis, a significant section of Big Data, is also referred as sentiment classification, sentiment mining, opinion analysis or opinion mining [5–7]. It is a method of text classification that is a significant area of research interest at the early 2000s as a result of the large number of prejudiced texts available in the social forums, blogs, and media [8, 9]. The sentiment analysis plays a most important role in opinion polls, e-commerce, agronomy [10, 11], speech recognition [12], and education [13, 14]. Sentiment analysis stands as a platform to explore the opinions of the users through the analysis of the posts and comments in twitter blogs regarding a product, policies, and topics [15]. From the analysis of the reviews of the users, one can obscure the attitude of the user regarding any event or topic

[*]For correspondence

Published online: 14 May 2020

[9, 16]. Sentiment analysis ensures any company to have a quick overview over the reviews of the customer over their product in social media that is updated regularly. On the other hand, researchers convey that there exists a relationship among the sentiment analysis of stock values and social media such that the stock price in future could be predicted based on the sentiment analysis carried out on blogs [14, 17].

A sentiment lexicon possesses words and phrases that convey either negative or positive sentiments. The opposite orientation of words in various domains raises a need for a better sentiment classification in spite of the presence of sentiment lexicon [14]. The sentiment classification by the machine learning methods analyzes the online texts of various users and extracts the properties of the users at the time of classification. The number of words in a review is limited to minimum characters in the tweets; hence, users utilize simple words in conveying their feelings about any product or event. These simple words are rich in information essential for performing the sentiment classification. However, these words are not employed by all the users. For illustration, let us consider the extroverts, who convey their emotions directly and they engage in using very concise words for sentiment expression [18], such as “HB” that is the short form of “Happy Birthday” and is not habitually employed by the other users available in the reviews. The conventional machine learning based techniques unsheathe the general sentiment features available in online texts, but they did not succeed in distinguishing the significant features and thereby causing the poor performance of the sentiment classifiers. This makes the need for the fine-grained method to perform sentiment classification, mainly in microblog data [4].

The primary intention of the research relies on developing the technique for sentiment classification using the MapReduce framework that is inbuilt with the proposed Chronological-Brain Storm Optimization based Support Vector Neural Network (CBSO-SVNN). Initially, the movie review from the database is pre-processed to remove the insignificant words from the reviews such that the complexity associated with the feature extraction is reduced. The pre-processed reviews are split into subsets that enter the map-reduce framework, which is comprised of two modules, such as mapper and reducer modules. The mapper module performs the feature extraction that presents the feature vector for individual movie reviews for effective classification. The proposed classifier, called CBSO-SVNN, performs the classification in the reducer module. The importance of map-reduce framework is that it manages the big data. The CBSO algorithm that tunes the optimal weights for tuning SVNN is developed through the introduction of chronological concept into Brain Storm Optimization (BSO) algorithm.

The contribution of the paper:

Chronological-Brain Storm Optimization based Support Vector Neural Network (CBSO-SVNN): In CBSO-SVNN, the optimal weights of the SVNN classifier are selected by the proposed CBSO algorithm.

Chronological-Brain Storm Optimization (CBSO): The CBSO algorithm is developed by integrating the chronological concept in the standard BSO algorithm.

The structure of the paper: Section 1 displays the background of the sentiment classification and section 2 discusses the existing works in sentiment classification and the challenges associated with the sentiment classification. The proposed sentiment classification approach is deliberated in section 3. The results of the proposed sentiment classification approach are discussed in section 4 and section 5 summarizes the research work.

2. Literature survey

This section deliberates the review of the existing sentiment classification methods with their merits and demerits.

Vo Ngoc Phu *et al.* [19] developed a sentiment classification approach based on the Fuzzy C-Means (FCM) to process the big data with millions of English documents. The implementation time of this method was less to extract the sentiment features. However, in distributed systems, this method required more time for implementation and this method was expensive. Han Liu and Mihaela Cocea [20] modelled a fuzzy information granulation approach that was proficient for dealing with the textual data, and also facilitated the interpretability. This method was not applicable for multi-granularity processing of sentiment data. Xiaojia Pu *et al.* [21] developed a method, named as Structural SVM, which offers high accuracy and requires the minimum training time. This method was not applicable for deep learning schemes. Andreas Kanavos *et al.* [22] developed a sentiment analysis classification framework that was efficient, robust and scalable, but the method could not handle complex semantic issues. Apostolos Filippas, and Theodoros Lappas *et al.* [23] developed a Bigcounter algorithm that effectively delivers valuable insight on the connection between data size and performance, but the method could not mine actionable insights from Big Data using the effective algorithms. Arulmurugan *et al.* [16] developed a method, named as unique emotional modeling methodology, which increased the classification accuracy of sentiments. In case of large documents, the performance of this method was poor. Farhan Hassan Khan *et al.* [7] modelled a method, named as Sentiment Knowledge Base (SKB) approach, that increased the efficiency associated with the extraction of the semantic features. The method could not handle other kinds of sentiment lexicons. Dandan Jiang *et al.* [24] developed Word emotion computation algorithm that computed the news event sentiment exactly, but the method could not consider word emotion pattern and emotion distance.

2.1 *Challenges*

- Sentiment classification impacts a prominent role in daily life mainly, in the political fields, in events associated with the production commodity, and commercial activities. It is a hectic challenge to classify the emotions accurately and within a short period [19].
- It is the need to understand and characterize the feelings of the users in the highly complex and sensitive environment that stays in the sensible discussion between most of the psychologists [16].
- The main problem associated with the affective processing is that the naming of the emotion is based on the habitual conditions of the population that is under review and based on the application setting [25]. On the other hand, it is a tedious process to maintain the vital databases as a mirror of large number of feelings and at the same time, representing social and relevant contrasts is a hectic demerit [16].
- Existing methods of sentiment classification fail due to the problems associated with the unavailability of the data, sparsity, dependence of the data with respect to the domain, and poor adequate performance of classification [7].
- It is unable to adopt the existing semantic classification strategies directly for any of the events that are new for classification because of the following reasons: (1) The existing methods concentrated on the adjectives and adverbs at the time of emotion classification such that the semantics availing in the documents are missed out from getting recognized. (2) Due to two extremes, one highly relies on standard emotion thesaurus and the other completely abandons standard emotion thesaurus [24].
- The existing sentiment classification techniques fail to handle the big data sets efficiently. A large number of machine learning techniques are employed for improving the classification accuracy, but took large time for training and hence not applicable for big data [1].
- The lexicon based methods require more labours and have poor flexibility issues. Hence, the lexicon based methods are not suitable in open-domain environments like microblog [4].

3. Proposed method

Sentiment classification, a text classification strategy extracts the opinions, emotions, and attitudes of the persons available in the form of the written language such that the issues associated with accuracy in fixing the polarities of the text available in the reviews are handled effectively. Sentiment analysis finds valuable application in the business and social domain such that the behaviours and impact on the users are retrieved. However, the growth of the online reviews is growing exponentially such that the existing traditional systems fail to extract the user attitudes because of the huge computational complexity. To tackle the issues associated with the existing systems and release the burden of computation, map-reduce framework is employed for the sentiment classification. The intention of the research is to carry out the sentiment classification of the movie reviews using the map-reduce framework. The schematic representation of the proposed CBSO-SVNN in sentiment classification of movie reviews is portrayed in figure 1. The map-reduce framework comprises two phases that use the feature extraction methodology in the mapper phase and proposed CBSO-SVNN algorithm for the classification in the reducer phase. The feature extraction in the mappers is progressed at the sentence level and word level such that the features ensure the accuracy of the classification. The classification at the reducer phase computes the class of the review using the proposed algorithm. In this paper, movie reviews are employed for sentiment classification as there are a large collection of the reviews that summarize the views of the reviewers regarding the movie, its goodness and badness. It is the criticism of the movies regarding the overall quality of the movie and it decides if the movie is worth recommending and these reviews are conveyed through various social media. The reviews are in the form of star indicators such that there is no need for the hand-level indicators. The need for sentiment classification of the movie reviews is that the opinions of the reviewers are extracted well to enhance the decision-making to the customers insisting them to like or dislike the movie from watching. Let us consider the movie review database, denoted as *M*, that carries the reviews of the individual users.

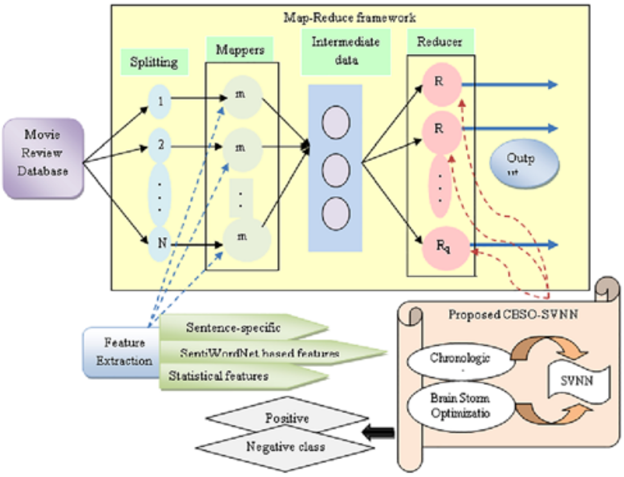

Figure 1. Schematic representation of the proposed CBSO-SVNN in sentiment classification of movie reviews.

3.1 *Pre-processing*

The movie review is pre-processed to remove the insignificant words as these words add no meaning to the classification. These insignificant words are referred as stop words and the examples of stop words include “in”, “the”, and so on. The pre-processed reviews are fed to the map-reduce framework such that the complexity associated with the processing is reduced.

3.2 *Sentiment classification*

Movie reviews give the overview of the film regarding its overall quality, objective of the movie, analysis of the film formal techniques, and the cinematic experience of the reviewer. The need for the map-reduce framework is to manage the movie reviews obtained from the distributed sources such that map-reduce platform enables a platform to store the distributed data across the available servers. Moreover, map-reduce is an inexpensive platform for performing the parallel processing of the data, and it offers better scalability. The map-reduce framework performs parallel processing and returns a flexible solution with high speed, and above all, map-reduce framework has high tolerance towards failure. The map-reduce framework comprises the mapper and reducer phases, each of which aim at two major functions. The map-reduce accepts a set of inputs and generates a set of outputs based on the algorithms employed. The mappers extract the features of the reviews and present it to the reducers for the classification that classifies as positive and negative reviews.

3.2.a *Feature extraction in the Mapper phase* The function of the mapper is to map the input data to form an intermediate data that is not essentially to be the same type as that of the input data. The main role of the mapper phase is to develop the features out of the big movie reviews given as input to them such that the mapper phase aims at the feature reduction that minimizes the complexity associated with the classification. Feature selection is a vital data pre-processing method in classification problems. The feature selection [26–28] enhances the classification accuracy of the train reviews. As the beginning of the sentiment classification, the movie reviews in the form of the big data are split into sub-data line by line such that several reviews enter the individual mappers of the map-reduce framework. Thus, for a big data, there are thousands of mappers operating parallel such that the effective extraction of the features is enabled from all the movie reviews. The reviews from the data centers are read by the mappers and are converted as feature vector such that the features, such as sentence-specific features [29], SentiWordNet-based features and statistical features are obtained from the reviews. The number of mappers in the mapper phase is given as

$$m = \{m_1, m_2, \ldots, m_i, \ldots, m_N\} \tag{1}$$

where, N denotes the total number of mappers that process the data from the distributed systems. The review in the i^{th} mapper is given as

$$m_i = \left\{t_1, t_2, \ldots, t_r, \ldots, t_p\right\} \tag{2}$$

The dimension of the data in the i^{th} mapper is given as $(1 \times p)$and p is the total number of reviews running in a single mapper. The dimension of the movie review M is given as, $(N \times p)$. Let us assume that there are v number of words in the r^{th} review that is given as,

$$t_r = \left\{S_r^1, S_r^2, \ldots, S_r^x, \ldots, S_r^v\right\} \tag{3}$$

Once the movie reviews are obtained by the mappers, the features of the individual reviews are extracted in the mappers as given below.

The better extraction of the movie features improves the classification accuracy and enhances the effectiveness of classification. The feature vector of the r^{th} review is given as

$$g^r \in g_r^j; (1 \leq j \leq 11) \tag{4}$$

where, g_r^j is the j^{th} feature of the r^{th} review in the database containing big data.

i) *Sentence-specific features:* Sentence-specific features [29] are the basic features of the sentiment classification that enables the extraction of the textual features from the train reviews. It is essential to frame the polarity of the sentence from the movie reviews. The sentence-specific features include the hash tags, punctuations, and non-dictionary words. The hash tag feature is the feature that counts the number of hash tags present in the individual reviews of the big movie review database. The hash tag feature is given as

$$g_r^1 = |H(t_r)| \tag{5}$$

where, $H(t_r)$ is the total number of the hash tags in the r^{th} review. The feature based on the punctuation is the count of the number of dots, question marks, and exclamation marks present in the individual reviews of the movie review database that is given as

$$g_r^2 = |P(t_r)| \tag{6}$$

where, $|P(t_r)|$ denotes the total number of punctuations present in the r^{th} review. The third feature is based on the non-dictionary words that is given as

$$g_r^3 = |D(t_r)| \tag{7}$$

where, $|D(t_r)|$ is the total number of the non-dictionary words present in the r^{th} review.

ii) *SentiWordNet based features:* SentiWordNet [30] is the lexical resource that assigns a value to the individual words in the review such that the value can be a non-zero value. The SentiWordNet is the polarity score decider that estimates both the positive and negative scores of the individual words in the movie review.

$$\left[g^{\rho}_{r,x}, g^{N}_{r,x}\right] = sentinet\left[S^{x}_{r}\right] \tag{8}$$

where, $g^{\rho}_{r,x}$ is the positive score of the x^{th} word in the r^{th} review and $g^{N}_{r,x}$ is the negative score for the x^{th} word in the r^{th} review. The positive score of the individual movie review is computed as the sum of the positive scores obtained for all of the words in the reviews that is given as

$$g^{4}_{r} = \sum_{e=1}^{|\rho|} g^{e}_{r,x} \tag{9}$$

where, $|\rho|$ refers to the total number of the positive scores for all the significant words in the r^{th}review. The negative score for an individual movie review is the summation of the negative scores of the words present in the individual movie review that is given as

$$g^{5}_{r} = \sum_{e=1}^{|\mathrm{N}|} g^{e}_{r,x} \tag{10}$$

where, $|\mathrm{N}|$ refers to the total number of negative scores for all the significant words in the r^{th} review.

iii) *Statistical features*: The statistical features include mean, variance, and standard deviation such that these features ensure the texture interpretation of the words in the reviews to improve the classification accuracy. The mean, variance, and standard deviation for the positive score and negative score of the words are computed separately.

Statistical features using the Positive score of the individual movie review: The mean of the positive score for the pre-processed review is computed by taking the average of the positive score of every word in the pre-processed review. The mean of the r^{th} review is given as

$$\mu^{\rho}_{r,x} = \frac{1}{|\rho|} \sum_{\substack{e=1 \\ e\in\rho}}^{|\rho|} g^{e}_{r,e} \tag{11}$$

where, $g^{e}_{r,e}$ specify the positive score of the r^{th} review. The variance of the positive score of the individual review is computed using the negative score of the individual word in the review and the mean of the positive score of the r^{th} review. The variance of the r^{th} review using the positive score is given as

$$V^{\rho}_{r,x} = \frac{1}{|\rho|} \sum_{\substack{e=1 \\ e\in\rho}}^{|\rho|} \left(g^{e}_{r,e} - \mu^{\rho}_{r,x}\right)^2 \tag{12}$$

where, $\mu^{\rho}_{r,x}$ is the mean of the r^{th} review. The standard deviation of the positive score of the r^{th} review is given as the square root of the variance of the positive score of the words in the r^{th} review as given as

$$SD^{\rho}_{r,x} = \sqrt{V^{\rho}_{r,x}} \tag{13}$$

where, $V^{\rho}_{r,x}$ is the mean of the r^{th} review using the positive score.

Statistical features using the Negative score of the individual movie review: The statistical features using the negative score of the movie reviews are depicted in this section. The mean of the r^{th} review is the average of the negative score of the words present in the pre-processed review, given as

$$\mu^{\mathrm{N}}_{r,x} = \frac{1}{|\mathrm{N}|} \sum_{\substack{e=1 \\ e\in\mathrm{N}}}^{|\mathrm{N}|} g^{e}_{r,e} \tag{14}$$

where, $g^{e}_{r,e}$ specify the negative score of r^{th} review. The variance of the r^{th} review for the negative score is computed using the negative score of the individual words in the pre-processed review and the mean of the negative score of the r^{th} review is given as

$$V^{\mathrm{N}}_{r,x} = \frac{1}{|\mathrm{N}|} \sum_{\substack{e=1 \\ e\in\mathrm{N}}}^{|\mathrm{N}|} \left(g^{e}_{r,e} - \mu^{\mathrm{N}}_{r,x}\right)^2 \tag{15}$$

where, $\mu^{\mathrm{N}}_{r,x}$ is the mean of the r^{th} review using the negative score. The standard deviation of the r^{th} review using the negative score is computed as the square root of the variance of the r^{th} review as follows

$$SD^{\mathrm{N}}_{r,x} = \sqrt{V^{\mathrm{N}}_{r,x}} \tag{16}$$

where, $V^{\mathrm{N}}_{r,x}$ is the mean of the r^{th} review using the negative score. The feature vector of the r^{th} review is denoted as

$$g^{r} = \left\{g^{r}_{j}\right\}; (1 \le j \le 11) \tag{17}$$

where, g^{r}_{j} is the j^{th} feature vector of the r^{th} review. The features are extracted in the individual mappers that present the dimensionally reduced data such that the classification of movie reviews becomes less complex and offers better classification accuracy. Thus, the output from the individual mappers is aggregated together to form the intermediate data such that the input to the reducer is the intermediate data. The reducers, which perform classification using the proposed algorithm, process the intermediate data that holds the feature vector of the individual movie reviews to classify the polarity of the reviews from the users. The derivation of the class label using the proposed classifier paves a platform to enhance the quality the movie.

3.2.b *Classification of reviews using the proposed algorithm*. The classification of the reviews based on the polarity is progressed in the reducer phase using the proposed CBSO-SVNN. The number of reducers is given as

$$R = \left\{R_1, R_2, \ldots, R_u, \ldots, R_q\right\} \tag{18}$$

where, q is the total number of reducers in the reducer phase. Each of the reducers receives the intermediate data and acts upon the data to derive the class label such that the label describes the movie that has got a good response from the public or a negative response from the public.

The objective of the proposed CBSO is to derive the optimal weights to tune the SVNN classifier that intends to derive the class label of the movie reviews as positive or negative. The proposed CBSO fuses both the BSO [31–34] and chronological idea to find out the global optimal weights for tuning the SVNN.

BSO algorithm offers a promising platform for most of the theoretical and practical engineering applications and the main ability of BSO is that the optimization possesses the tendency to convert the functions in solution space to the functions in objective space. It is clearly known that the BSO algorithm depends on the human brainstorming activities. The BSO performs best at global exploration and fails in case of the local exploration as like other global optimization algorithms. The only way to ensure the performance of BSO is to bring a proper balance among the local exploitation and global exploration phases.

The BSO algorithm is an amalgamation of swarm intelligence and data mining methods. Each individual in the BSO algorithm is a solution to the problem and a data point to expose the background of the problem. Moreover, the BSO algorithm can solve multiobjective optimization problems. The BSO algorithm uses the objective space information directly; hence, it is better than the traditional multiobjective optimization techniques. In addition, BSO algorithm solves numerous real-world problems. Hybrid algorithms play an important role in improving the search capability of swarm intelligence algorithms. Hybridization has the benefits of every algorithm, at the same time aims to reduce the disadvantage of the algorithms. Hence, we use BSO algorithm together with the chronological concept to train the SVNN.

BSO algorithm depends on the iteration of the individual population at the individual iteration. Initially, the total BSO population is subdivided into two major groups, namely normal group and elite group. The generation of the new individual depends on the random generation of the individual either through the selection of one or two individuals from each of the groups and combining the selected individuals in such a way that a random number is added to the selected individual. The next step is the evaluation of the fitness in which the feasibility for the generated individual is verified. If the recently generated individual possesses better fitness than the already existing individual, the new individual is updated. This process is continued till the global optimal solutions are obtained and the existing populations are replaced with a new population. The demerit of BSO regarding the proper balance between the local and global exploitation is resolved by using the local solver. The population update ensures the restoration of best solution and the balancing parameter appears to be numerically insensitive. The global convergence is enhanced using the chronological concept.

The proposed CBSO is the integration of chronological idea in the BSO algorithm. The integration of the chronological idea in the BSO algorithm avoids the difficulty in BSO for converging to the global optimal solution. In BSO, the new individuals are generated depends on the random parameter and the randomly selected individual, while in the proposed CBSO, the new individuals are generated based on the individuals in the previous iterations. The CBSO algorithm inherits the advantages of both the BSO and chronological concept such that the merits and demerits are balanced among themselves.

Architecture of SVNN classifier: SVNN [35] consist of three layers, such as input layer, hidden layer, and output layer as shown in figure 2. The input to the SVNN is the features extracted from the individual reviews present in the database. The input is processed using the weights and bias of the hidden and output layers to generate the predictive results.

The output from the SVNN is given as

$$O_r^{SVNN} = F_2 \times \log sig\left[\left(\sum_{\substack{j=1\\ j\in r}}^{11} g_j^r \times F_j^1\right) + B_1\right] + B_2 \tag{19}$$

where, B_1 and B_2 are the biases of the input and the output layers, the j^{th} feature of the r^{th} review is denoted as g_j^r, F_j^1 denotes the weight of the j^{th} input neuron present in the input layer of SVNN and F_2 symbolizes the weights of the

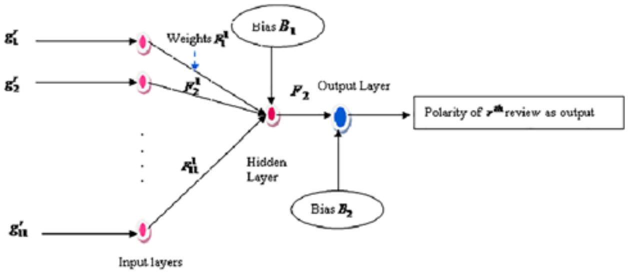

Figure 2. Architecture of SVNN.

hidden layer. The output of the SVNN decides the class label either the review is a positive review or negative review. Thus, the output from SVNN is given as,

$$O_r^{SVNN} = \begin{cases} 1; & Positive\ class \\ 0; & Negative\ class \end{cases} \tag{20}$$

The CBSO algorithm determines the weights of the SVNN classifier based on the following steps.

a) *Initialization:* The population initialization pictures the total number of individuals engaged in the optimization process. Let there be l number of populations in the search space.

b) *Formation of clusters:* In this step, the individuals are grouped under c number of clusters.

c) *Fitness calculation:* The fitness of the individuals present in the clusters is evaluated such that the best individual in the clusters is determined. The fitness of the individuals is computed using the formula given as,

$$F = \delta_{\max} + \delta_{\min} + \frac{D}{p}\sum_{r=1}^{p}\left|O_r - O_r^*\right| \tag{21}$$

where, D denotes the regularization factor, p is the total number of reviews, the output of the reducer for the r^{th} review is denoted as O_r and the ground truth of the r^{th} review is denoted as O_r^*. The factors $\delta_{\max}$ and $\delta_{\min}$ are computed using the maximum and minimum of the Eigen values of the weights that is given as

$$\delta = Eigen\left(X \times X^T\right) \tag{22}$$

$$\delta_{\max} = \max(\delta) \tag{23}$$

$$\delta_{\min} = \min(\delta) \tag{24}$$

d) *Ranking the individuals:* Once the fitness of the individuals is evaluated, the individuals in a cluster are ranked such that the individual that ranked highest is marked as the best individual.

e) *Random selection of the cluster center:* For the selection of the cluster center, generate a random value such that the selected random value ρ_1 does not exceed the pre-determined probability P_1. If the condition is satisfied, the cluster center is selected randomly and it is replaced with the selected individual.

f) *Generation of the new individuals:* The generation of the individuals is based on four probabilities P_2, P_3, P_4, and P_5 and three random numbers ρ_2, $\rho_3$and ρ_4. The generation of the new population is based on the following formula in standard BSO.

$$F_{z+1}^K = F_z^K + \lambda \times G(\mu, \sigma) \tag{25}$$

where, F_z^K is the K^{th} dimension of the selected individual, λ is the weight coefficient corresponding to the Gaussian random value, $G(\mu, \sigma)$ specifies the Gaussian random value, and F_{z+1}^K is the newly generated individual or the new weights corresponding to the $(z+1)^{th}$ iteration.

$$\lambda = \log sig\left[\left(\frac{(0.5 \times z_{\max}) - z_{current}}{s}\right)\right] \times rand() \tag{26}$$

where, $\log sig()$ is the logarithmic sigmoid transfer function, $z_{\max}$ belongs to the maximum number of the iterations, $z_{\min}$ is the current iteration, and s is the slope of $\log sig()$ and $rand()$ belongs to the random number that vary in the interval [0, 1]. Since the standard BSO suffers to converge with global solution, the chronological concept is integrated in BSO; hence, the derivation of the update equation for the proposed CBSO is as follows: The new individual generated in the previous iteration z is based on the $(z-1)^{th}$ iteration that is given as

$$F_z^K = F_{z-1}^K + \lambda \times G(\mu, \sigma) \tag{27}$$

Substituting Eq. (27) into Eq. (25),

$$F_{z+1}^K = F_{z-1}^K + \lambda \times G(\mu, \sigma) + \lambda \times G(\mu, \sigma) \tag{28}$$

$$F_{z+1}^K = F_{z-1}^K + 2 \times \lambda \times G(\mu, \sigma) \tag{29}$$

Hence, the update equation is given as

$$F_{z+1}^K = \frac{F_{z+1}^K + F_{z+1}^K}{2} \tag{30}$$

On substituting Eqs. (25) and (29) in Eq. (30),

$$F_{z+1}^K = \frac{F_z^K + \lambda \times G(\mu, \sigma) + F_{z-1}^K + 2 \times \lambda \times G(\mu, \sigma)}{2} \tag{31}$$

$$F_{z+1}^K = \frac{F_z^K + F_{z-1}^K + 3 \times \lambda \times G(\mu, \sigma)}{2} \tag{32}$$

The above equation forms the update rule of the proposed CBSO algorithm. The population update is progressed based on a condition. Initially, generate a random number ρ_2 that lies between 0 and 1 and check if ρ_2 lies below the pre-determined probability P_2. If $(\rho_2 < P_2)$, select a random cluster that possesses the probability P_3. If the condition $(\rho_2 < P_2)$ is satisfied and if the random number ρ_3 lies below the probability P_4, generate a random number ρ_3. If the condition $(\rho_3 < P_4)$ is satisfied, select the random cluster center, and add with a random value as in Eq. (32) to generate a new individual. If the condition $(\rho_3 < P_4)$fails, select the random individual from the cluster to produce a new individual based on Eq. (32). On the other hand, if the condition $(\rho_2 < P_2)$fails, select two cluster centers and generate a random number ρ_4. Check if $(\rho_4 < P_5)$ is attained, if yes, combine the selected two cluster centers and follows Eq. (32) to update the individual. If the condition $(\rho_4 < P_5)$ fails, two individuals from the selected two cluster centers are selected, combined and applied to Eq. (32) for the individual update. Thus, the generated solution obtained is the global optimal solution of CBSO. However, the feasibility of the new solutions is verified in the next step whether to allow the individual update. The above equation forms the update rule of the proposed CBSO algorithm. The population update is progressed based on a condition. Initially, generate a random number ρ_2 that lies between 0 and 1 and check if ρ_2 lies below the predetermined probability P_2. If $(\rho_2 < P_2)$, select a random cluster that possesses the probability P_3. If the condition $(\rho_2 < P_2)$ is satisfied and if the random number ρ_3 lies below the probability P_4, generate a random number ρ_3. If the condition $(\rho_3 < P_4)$ is satisfied, select the random cluster center, and add with a random value as in Eq. (32) to generate a new individual. If the condition $(\rho_3 < P_4)$ fails, select the random individual from the cluster to produce a new individual based on Eq. (32). On the other hand, if the condition $(\rho_2 < P_2)$ fails, select two cluster centers and generate a random number ρ_4. Check if $(\rho_4 < P_5)$ is attained, if yes, combine the selected two cluster centers and follows Eq. (32) to update the individual. If the condition $(\rho_4 < P_5)$ fails, two individuals from the selected two cluster centers are selected, combined and applied to Eq. (32) for the individual update. Thus, the generated solution obtained is the global optimal solution of CBSO. However, the feasibility of the new solutions is verified in the next step whether to allow the individual update.

g) *Fitness calculation of the new solution:* The fitness of the new solution (individual) is evaluated based on Eq. (21) and if the new solution is found to be better than the existing solution then, the existing solution is replaced by the new solution. Otherwise, the existing solution is retained.

h) *Stopping criterion*: The steps b) to g) are continued until every individual or solution is replaced. Once l number of replacements is completed, the algorithm has been terminated.

4. Results and discussion

This section deliberates the experimental results of the proposed method and the comparative discussion with the existing sentiment classification methods.

4.1 *Experimentation setup*

The implementation of the proposed method is done in the personal computer with Intel core processor, 2 GB RAM, and Windows 10 Operating System. The implementation tool used in this work is JAVA.

4.2 *Dataset description*

The dataset employed is the movie review database [36], from which four datasets are employed for the analysis. The movie-review data is a collection of movie-review text documents, which are labeled, based neither on the overall sentiment polarity in terms of positive or negative nor on the subjective rating in stars.

Dataset 1: Polarity dataset v1.0: This is dataset 1 that possess 700 positive and negative reviews.
Dataset 2: Polarity dataset v1.1: Dataset 2 consists of 700 positive and negative reviews that is presented after processing.
Dataset 3: Scale dataset v1.0: This is a collection of documents that are labeled with a rating scale.
Dataset 4: Subjectivity dataset v1.0: Dataset 4 consists of 5000 subjective and objective processed sentences.

4.3 *Performance metrics*

The performance of the proposed sentiment classification method is analyzed using three metrics, namely accuracy, sensitivity and specificity.
4.3.a *Accuracy:* It specifies the correctness of the classification, which is denoted in Eq. (33).

$$ACC = \frac{TP + TN}{TP + FP + FN + TN} \tag{33}$$

where, TP, TN, FN, and FP represent the true positive, true negative, false negative, and false positive, respectively

4.3.b *Sensitivity:* It is also called True Positive Rate (TPR) specifies the number of positives, which are identified correctly.

$$TPR = \frac{TP}{TP + FN} \tag{34}$$

4.3.c *Specificity:* It is also called True Negative Rate (TNR) specifies the number of negatives, which are identified correctly.

$$TNR = \frac{TN}{TN + FP} \tag{35}$$

Table 1. Simulation parameters.

Parameters	Values
CBSO	
Mapper size	4
Maximum number of iteration	10
Population size	10
BSO	
Maximum number of iteration	10
Population size	10
FCM	
Cluster size	5

4.4 *Competing methods*

The performance of the proposed method is compared with the existing methods, such as Fuzzy C-Means (FCM) [19], Support Vector Machine (SVM) [21], SentiWordNet [7], and Brain Storm Optimization (BSO) [31]. The parameters used for the experimentation are provided in table 1.

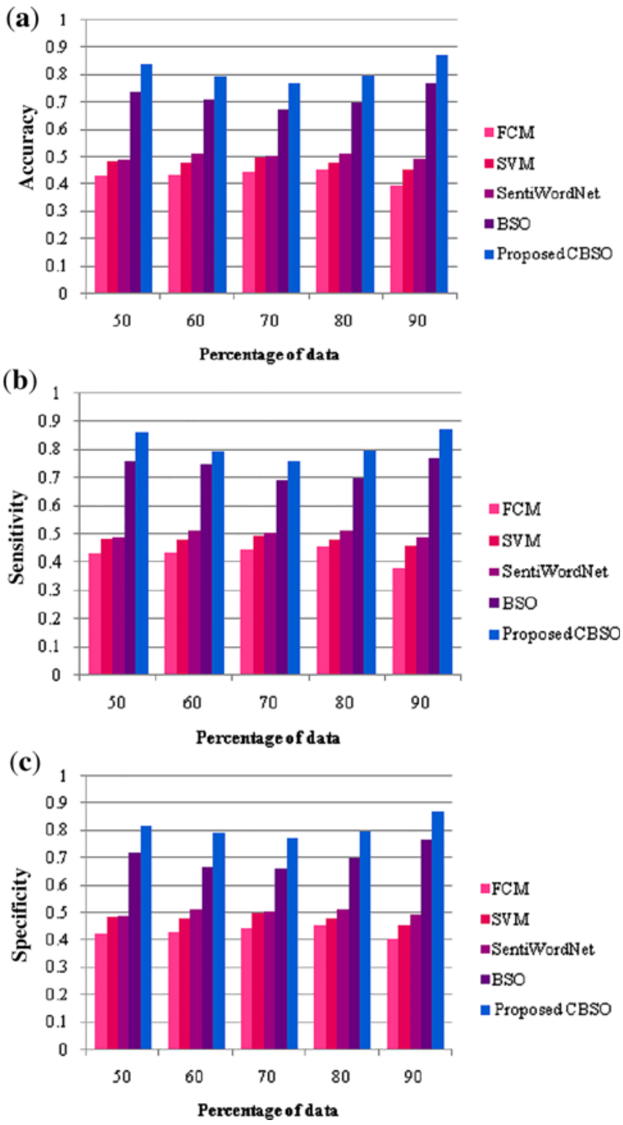

Figure 3. Comparative analysis using dataset 1 (a) accuracy (b) sensitivity (c) specificity.

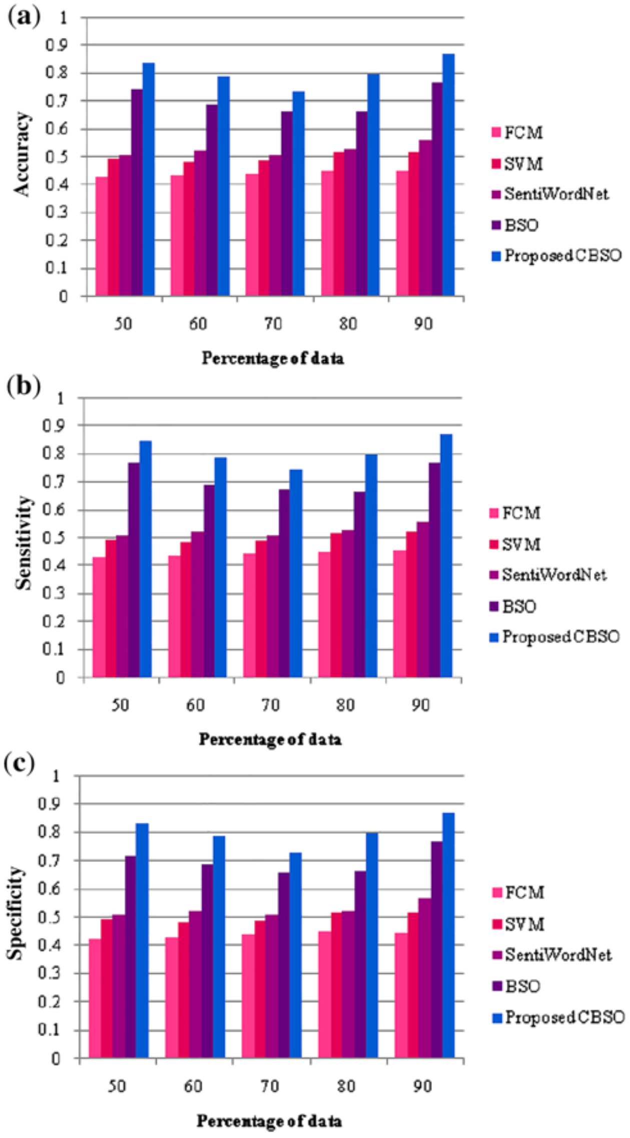

Figure 4. Comparative analysis using dataset 2 (a) Accuracy (b) Sensitivity (c) Specificity.

4.5 *Comparative analysis*

The comparative analysis is done using four datasets taken from the movie review database based on the percentage of training data.

4.5.a *Using dataset 1:* Figure 3 depicts the comparative analysis using dataset 1. The analysis based on accuracy is depicted in figure 3a. Figure 3a depicts that the increase in the percentage of training increases the accuracy of the methods. However, the proposed method outperforms the existing methods. At 90% training data, the accuracy of methods, such as FCM, SVM, SentiWordNet, BSO, and the proposed CBSO is 0.3958, 0.4583, 0.4930, 0.7714, and 0.8714, respectively. Figure 3b depicts the sensitivity analysis using dataset 1. With 90% training data, the sensitivity of methods, FCM, SVM, SentiWordNet, BSO, and the proposed CBSO is 0.3809, 0.4615, 0.4925, 0.7714, and 0.8714, respectively. Figure 3c shows the specificity analysis using dataset 1. With 90% as training data, the specificity of methods, FCM, SVM, SentiWordNet, BSO, and proposed CBSO is 0.4074, 0.4545, 0.4935, 0.7714, and 0.8714, respectively.

4.5.b *Using dataset 2:* Figure 4 shows the analysis using dataset 2 and the analysis of the accuracy is depicted in figure 4a. At 90% training data, the accuracy of methods, FCM, SVM, SentiWordNet, BSO, and proposed CBSO is 0.4513, 0.5208, 0.5625, 0.7714, and 0.8714, respectively. Figure 4b illustrates the sensitivity analysis using dataset 2. With 90% training data, the sensitivity of methods, like FCM, SVM, SentiWordNet, BSO, and proposed CBSO is

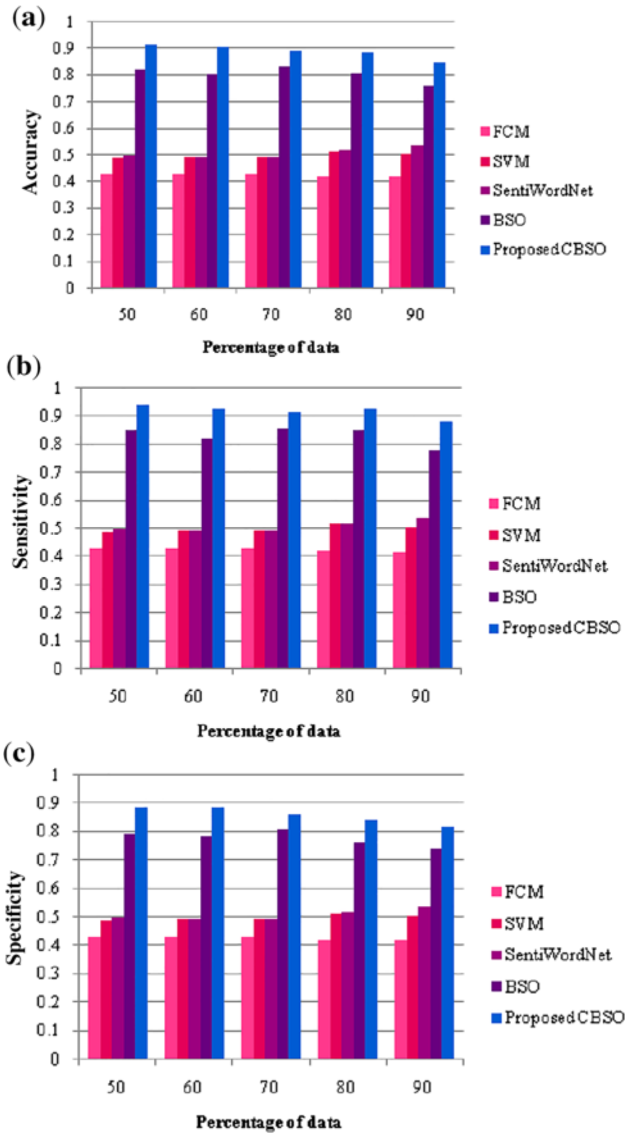

Figure 5. Comparative analysis using dataset 3 (a) accuracy (b) sensitivity (c) specificity.

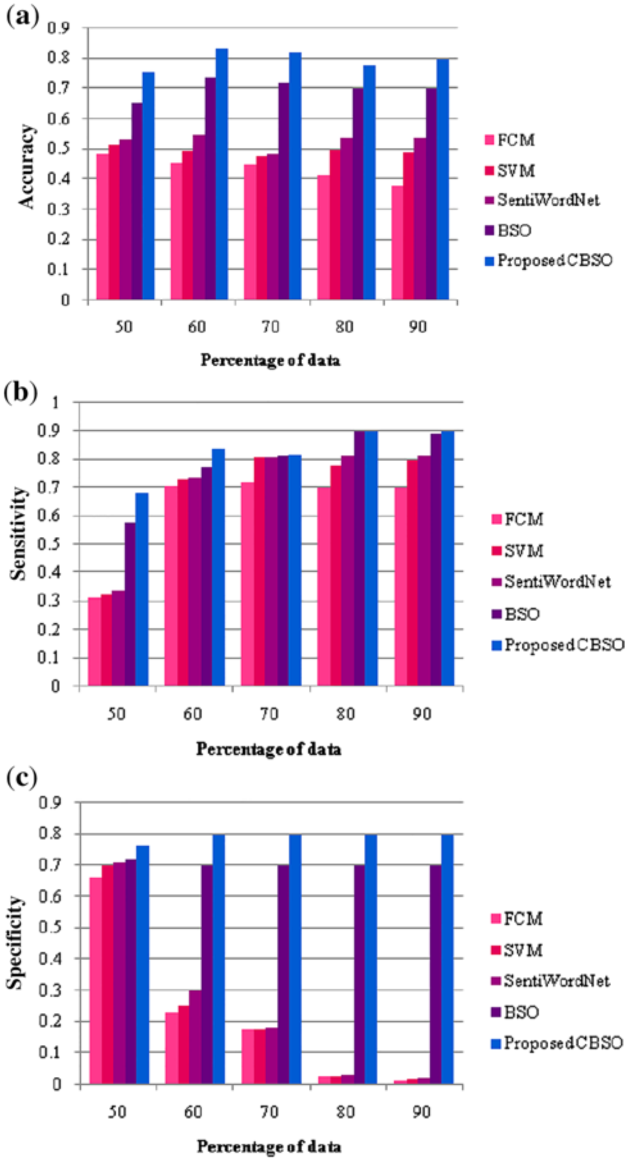

Figure 6. Comparative analysis using dataset 4 (a) accuracy (b) sensitivity (c) specificity.

Table 2. Comparative discussion of the sentiment classification methods.

Methods	FCM	SVM	SentiWordNet	BSO	Proposed CBSO
Accuracy	0.4574	0.4976	0.5625	0.8322	**0.8714**
Sensitivity	0.7208	0.8088	0.8156	0.9014	**0.9027**
Specificity	0.4577	0.7	0.7093	0.8066	**0.8714**
Computational time (s)	9	8.5	8	6.5	**5**

0.4556, 0.5230, 0.5569, 0.7714, and 0.8714, respectively. Figure 4c illustrates the specificity analysis using dataset 2. With 90% training data, the specificity of FCM, SVM, SentiWordNet, BSO, and proposed CBSO is 0.4461, 0.5189, 0.5693, 0.7714, and 0.8714, respectively.

4.5.c *Using dataset 3:* Figure 5 shows the analysis using the dataset 3 and the analysis of accuracy is depicted in figure 5a. At 90% training data, the accuracy of methods, FCM, SVM, SentiWordNet, BSO, and proposed CBSO is 0.4193, 0.5049, 0.5388, 0.76, and 0.85, respectively. Figure 5b depicts the sensitivity analysis using dataset 3. With 90% as training data, the sensitivity of methods, FCM, SVM, SentiWordNet, BSO, and proposed CBSO is 0.4188, 0.5051, 0.5395, 0.78, and 0.88, respectively. Figure 5c depicts the specificity analysis using dataset 3. With 90% as training data, the specificity of the methods, FCM, SVM, SentiWordNet, BSO, and proposed CBSO is 0.4198, 0.5048, 0.5381, 0.74, and 0.82, respectively.

4.5.d *Using dataset 4:* Figure 6 shows the analysis using dataset 4 and the analysis of accuracy is depicted in figure 6a. At 90% training data, the accuracy of FCM, SVM, SentiWordNet, BSO, and proposed CBSO is 0.3796, 0.4907, 0.5370, 0.7, and 0.8, respectively. Figure 6b portrays the sensitivity analysis using dataset 4. With 90% training data, the sensitivity of methods, FCM, SVM, SentiWordNet, BSO, and proposed CBSO is 0.7, 0.8, 0.8156, 0.8911, and 0.9027, respectively. Figure 6c depicts the specificity analysis using dataset 4. With 90% training data, the specificity of the methods, FCM, SVM, SentiWordNet, BSO, and proposed CBSO is 0.014, 0.018, 0.02, 0.7, and 0.8, respectively.

4.6 *Comparative discussion*

The comparative discussion of the sentiment classification methods based on the maximum performance is depicted in table 2. The accuracy, specificity, and sensitivity of the proposed CBSO are 0.8714, 0.9027, and 0.8714, respectively. The maximum accuracy of the existing FCM, SVM, SentiWordNet, and BSO is 0.4574, 0.4976, 0.5625, and 0.8322, respectively. When the maximum sensitivity of the existing FCM, SVM, SentiWordNet, and BSO is 0.7208, 0.8088, 0.8156, and 0.9014, the specificity of FCM, SVM, SentiWordNet, and BSO is 0.4577, 0.7, 0.7093, and 0.8066, respectively. In addition, the computational time of the proposed method is minimum than that of the existing methods. Thus, it is clear that the proposed method performs best compared with the existing methods of sentiment classification.

5. Conclusion

In this work, the sentiment classification is progressed using the MapReduce framework that involves two main functions. The MapReduce framework is capable of dealing with the high dimensional data and one of the main functions is feature extraction. In the feature extraction, which is carried out in the mapper phase, the sentence level and word level features are extracted. The second function is the sentiment classification that is carried out in the reducer phase of the MapReduce framework. The classification is performed using the SVNN classifier and the weights of SVNN classifier are tuned optimally using the proposed CBSO algorithm that is designed based on the chronological concept and BSO. The SVNN classifier based on the proposed CBSO algorithm derives the polarity of the movie reviews such that the emotion of the individual reviewers is extracted from the sentences available online. The extraction of the sentiments buried in the sentences is essential in improving the quality and effectiveness of movies and attracts the customers with a good view about the movie. The analysis using four datasets taken from the movie review database reveals that the accuracy, sensitivity, and specificity of the proposed CBSO are 0.8714, 0.9027, and 0.8714, respectively. The proposed method ensures the effective classification of movie reviews when compared with the existing methods of sentiment classification.

References

[1] Kurian D D M K, Vishnupriya S, Ramesh R, Divya G, Divya D, Kurian M K, Vishnupriya S, Ramesh R, Divya G and Divya D 2015 Big data sentiment analysis using hadoop. *Int. J. Innov. Res. Sci. Technol.* 1(11): 92–96

[2] Matwankar S H and Shinde S K 2015 Sentiment analysis for Big Data using data mining algorithms. *Int. J. Eng. Res. Technol. (IJERT)* 4(09): 962–965

[3] Anjaria M and Guddeti R M R 2014 Influence factor based opinion mining of Twitter data using supervised learning. In: *Proceedings of 2014 Sixth IEEE International Conference on*

Communication Systems and Networks (COMSNETS), pp. 1–8

[4] Lin J, Mao W and Zeng D D 2017 Personality-based refinement for sentiment classification in microblog. *Knowl. Based Syst.* 132: 204–214

[5] Phu V N, Chau V T N, Tran V T N, Duy D N and Duy K L D 2017 A valence-totaling model for Vietnamese sentiment classification. *Evolv. Syst.* 1–47

[6] Khan F H, Qamar U and Bashir S 2016 SentiMI: introducing point-wise mutual information with SentiWordNet to improve sentiment polarity detection. *Appl. Soft Comput.* 39: 140–153

[7] Khan F H, Qamar U and Bashir S 2017 A semi-supervised approach to sentiment analysis using revised sentiment strength based on SentiWordNet. *Knowl. Inf. Syst.* 51(3): 851–872.

[8] Xia R, Zong C and Li S 2011 Ensemble of feature sets and classification algorithms for sentiment classification. *Inf. Sci.* 181(6): 1138–1152

[9] Liu B 2012 Sentiment analysis and opinion mining. *Synth. Lect. Hum. Lang. Technol.* 5(1): 1–167

[10] Freitas L C S R, Tinôco I F F, Baêta F C, Barbari M, Conti L, Júnior C T, Cândido M, Morais C, and Sousa F 2017 Correlation between egg quality parameters, housing thermal conditions and age of laying hens. *Agron. Res.* 15(3): 687–693

[11] Andrade R R, Tinôco I F F, Baêta F C, Barbari M, Conti L, Cecon P R, Cândido M G L, Martins I T A and Teles Junior C G S 2017 Evaluation of the surface temperature of laying hens in different thermal environments during the initial stage of age based on thermographic images. *Agron. Res.* 15(3): 629–638

[12] Darekar R V and Dhande A P 2019 Emotion recognition from speech signals using DCNN with hybrid GA-GWO algorithm. *Multimed. Res.* 2(4): 12–22

[13] Quan C and Renm F 2014 Unsupervised product feature extraction for feature-oriented opinion determination. *Inf. Sci.* 272: 16–28.

[14] Catal C and Nangir M 2017 A sentiment classification model based on multiple classifiers. *Appl. Soft Comput.* 50: 135–141

[15] Pang B and Lee L 2008 Opinion mining and sentiment analysis. *Found. Trends Inf. Retriev.* 2(12): 1–135

[16] Arulmurugan R, Sabarmathi K R and Anandakumar H 2017 Classification of sentence level sentiment analysis using cloud machine learning techniques. *Clust Comput* 1–11

[17] Smailovic J, Grcar M, Lavrac N and Znidarsic M 2014 Stream-based active learning for sentiment analysis in the financial domain. *Inf. Sci.* 285: 181–203

[18] Hirsh J B and Peterson J B 2009 Personality and language use in self-narratives. *J. Res. Person.* 43(3): 524–527

[19] Phu V N, Dat N D, Tran V T N, Chau V T N and Nguyen T A 2017 Fuzzy C-means for English sentiment classification in a distributed system. *Appl. Intell.* 46(3): 717–738

[20] Liu H and Cocea M 2017 Fuzzy information granulation towards interpretable sentiment analysis. *Granular Comput.* 1–14

[21] Pu X, Wu G and Yuan C 2017 Exploring overall opinions for document level sentiment classification with structural SVM. *Multimed. Syst.* 1–13

[22] Kanavos A, Nodarakis N, Sioutas S, Tsakalidis A, Tsolis D and Tzimas G 2017 Large scale implementations for Twitter sentiment classification. *Algorithms* 10(1): 33

[23] Filippas A and Lappas T 2017 Strength in numbers: using Big Data to simplify sentiment classification. *Big Data* 5(3): 256–271

[24] Jiang D, Luo X, Xuan J and Xu Z 2017 Sentiment computing for the news event based on the social media Big Data. *IEEE Access* 5: 2373–2382

[25] Anandakumar H and Umamaheswari K 2017 Supervised machine learning techniques in cognitive radio networks during cooperative spectrum handovers. *Clust. Comput.* 1–11.

[26] Zhang Y, Gong D and Cheng J 2017 Multi-objective particle swarm optimization approach for cost-based feature selection in classification. *IEEE/ACM Trans. Comput. Biol. Bioinf.* (*TCBB*) 14(1).

[27] Zhang Y, Gong D, Gao X and Tian T 2019 Binary differential evolution with self-learning for multi-objective feature selection. *Inf. Sci.*

[28] Zhang Y, Cheng S, Shi Y, Gong D and Zhao X 2019 Cost-sensitive feature selection using two-archive multi-objective artificial bee colony algorithm. *Expert Syst. Appl.* 137 46–58

[29] Tang D, Qin B, Wei F, Dong L, Liu T and Zhou M 2015 A joint segmentation and classification framework for sentence level sentiment classification. *IEEE/ACM Trans. Audio Speech Lang. Process.* 23(11): 1750–1761

[30] Esuli A and Sebastiani F 2006 SENTIWORDNET: a publicly available lexical resource for opinion mining. In: *Proceedings of the 5th Conference on Language Resources and Evaluation* (*LREC'06*), pp. 417–422

[31] Shi Y 2011 Brain storm optimization algorithm. In: *International Conference in Swarm Intelligence, Advances in Swarm Intelligence*, pp. 303–309

[32] Dai C and Lei X 2019 A multiobjective brain storm optimization algorithm based on decomposition. *Complexity* 2019: 11

[33] Cheng S, Sun Y, Chen J, Qin Q, Chu X, Lei X and Shi Y 2017 A comprehensive survey of brain storm optimization algorithms. In: *Proceedings of the IEEE Congress on Evolutionary Computation (CEC), San Sebastian, Spain*

[34] Cheng S and Shi Y 2019 Brain Storm Optimization Algorithms: Concepts, Principles and Applications. Springer, Berlin

[35] Ludwig O, Nunes U and Araujo R 2014 Eigenvalue decay: a new method for neural network regularization. *Neurocomputing* 124: 33–42

[36] Movie review data taken from http://www.cs.cornell.edu/people/pabo/movie-review-data/. Accessed on 2018