

RESEARCH

Open Access



Sentiment analysis of online product reviews using DLMNN and future prediction of online product using IANFIS

P. Sasikala^{1*}  and L. Mary Immaculate Sheela²

*Correspondence:
shakthesasi@gmail.com

¹ Department of Computer Science, Mother Teresa Women's University, Kodaikanal, India
Full list of author information is available at the end of the article

Abstract

A major task that the NLP (Natural Language Processing) has to follow is Sentiments analysis (SA) or opinions mining (OM). For finding whether the user's attitude is positive, neutral or negative, it captures each user's opinion, belief, and feelings about the corresponding product. Through this, needed changes can well be done on the product for better customer contentment by the companies. Most of the existent techniques on SA aimed at these online products have extremely low accuracy and also encompassed more time amid training. By employing a Deep learning modified neural network (DLMNN), a technique is proposed aimed at SA of online products review; in addition, via Improved Adaptive Neuro-Fuzzy Inferences System (IANFIS), a technique is proposed aimed at future prediction of online products to trounce the above-stated issues. Firstly, the data values are separated into Contents-based (CB), Grades-based (GB), along with Collaborations based (CLB) setting as of the dataset. Then, each setting goes via review analysis (RA) by employing DLMNN, which renders the results as negative, positive, in addition to neutral reviews. IANFIS carry out a weighting factor and classification on the product for upcoming prediction. In the experimental assessment, the proposed work gave an enhanced performance compared to the existing methods.

Keywords: Sentiment analysis, Sentiment analysis of online product reviews, Review analysis, Deep learning modified neural network (DLMNN), Adaptive neuro-fuzzy inference system (ANFIS), Improved ANFIS (IANFIS)

Introduction

Since, there are numerous brands that are present in the market; selecting one will be a tough task for a consumer. The advancement of E-Commerce influences the buying routine of customers. Buyers make the desired decision centered on the reviews present in E-commerce (i.e. the ratings and summary of relevant text about the items can well be utilized aimed at decision making [1]). Along with it, the reviews for the product can as well be seen in Social networking sites [2]. In present years, social networks have turned very popular; so there is a chance that because of those sites, the expansion of data can be uncontrollable in the future [3, 4]. Since, everyone is posting comments every day, which led to the immense continuous augmentation in the online data and

online information. Thus, it is really hard to accurately extract relevant information as of the internet [5]. The customers together with manufacturers will attain as of analyzing the positive along with the negative sentiments regarding every product that can well be attained via SA. SA stands as chief tasks in NLP [6–8]. By employing SA, the mood or attitude of the critic can well be determined as negative or positive [9]. In SA, all product reviews to be summarized and sentiments are to be classified.

SA stands as a field that evaluates the people's opinions, evaluations, sentiments, attitudes, appraisals, as well as the emotion that they encompass on entities cherish products, organizations, services, and people [10, 11]. The links between SA and product design stay comparatively uncharted regardless of the swift advancements of SA in other fields. The primary aim of SA is to recognize the data's polarity on the Web and then to classify them. SA is text centered analysis; however there are particular challenges to discover the precise polarity of the sentence. The SA encompasses '3' stages: (i) document, (ii) sentence and (iii) aspect-level [12]. In sentence level, the document SA checks the entire document and categories the opinion as negative, positive or neutral (i.e. a single document might encompass manifold opinions even concerning the same entities [13]). The aspect level finds the target of the opinion indicating that every opinion has a target. The bag-of-words for SA wherein the relationships among words were not considered and a document is nothing but a compilation of words [14]. There are '2' sorts of methods for SA: (i) semantic orientation, (ii) statistical Machine Learning (ML) approach. The former approach ascertains the document's sentiment grounded upon the extracted sentimental words as well as phrases. In addition, the latter focuses upon the document's sentiment grounded on the extracted sentimental features as well as the ML. The statistical one overcomes the semantic scheme regarding SA accuracy [15]. Semantic word spaces are extremely valuable; however, it can't articulate the meaning of lengthy sentences on an moral method [16]. In order to interpret as well as understand human emotions in addition to feelings, the machines have to be dependable and efficient.

Most SA is grounded on supervised ML [17]. An vital role is played via the Feature extraction (FE) in addition to classifier design of texts. Term frequency, Term Occurrence, Binary term occurrence, and Terms Frequency-Inverse Documents Frequency (TF-IDF) are the disparate methods for Feature Selections (FS) [18]. The TF-IDF usually uses the sentiment lexicon to choose the feature words as well as calculate weights that were broadly applied to traditional NLP tasks [19].

Several methods for SA were proposed in the precedent decades, most of which are centered on computational linguistic approach and ML approach, for instance Naive Bayes (NB) Maximum Entropy along with Supports Vectors Machines (SVM) [20]. These models are trained on the feature vectors derived as an output as of the Latent Dirichlets Allocation (LDA) and the sentiment in the text is classified as positive or negative. From several methods, it can be stated that the ML approach exhibit higher performance than the computational linguistic approach [21].

As Deep Learning (DL) showed remarkable outcomes for an assortment of NLP tasks, it has captured the researchers' attention [22]. DL is a division of ML and it encompasses manifold layers of perceptron that is stimulated by means of the brain [23]. Numerous DL models are present, for instance deep neural networks (DNN), convolutions neural networks (CNN) [24], deep CNN [25], deep Restricted Boltzmanns Machine (RBM), etc.

Although, there are many propitious outcomes with DL in NLP, it hasn't overcome the prevailing issues [26]. The below-mentioned restrictions affect the existing work that is centered on SA. The limitations include (1) the knowledge of hierarchical connections of product aspects is not completely used. (2) Sentences or reviews stating many aspects associated with complex sentiments aren't dealt very well [27]. Also, the overall appraisal of the SA using ML hasn't provided better accuracy and efficient training time. So, this paper proposed an efficient SA of online products reviews. The chief contribution of the proposed methodology is enlisted as follows,

- The SA is done in '3' scenarios: grade based, content-based and collaboration based.
- The review is analyzed for the '3' scenarios using the DLMNN classifier, which gives enhanced accuracy and low training time.
- The future prediction of the product is done utilizing the IANFIS system.

The draft of this paper is prearranged as Sect. "Literature review" reviews the associated works concerning the technique proposed. Section "Proposed methodology", proffers a concise discussion concerning the proposed methodology, Sect. "Results and discussion", elucidates the Investigational outcome and Sect. "Conclusion" briefly concludes the paper.

Literature review

Feilong Tang et al. [28] suggested '2' generative model, MaxEnt-JABST as well as JABST, that extracted typically the fine-grained opinions along with aspects as of reviews (online). The JABST model extracted particular and general opinions and aspects together with the sentiment polarity (SP). In addition, the MaxEnt-JABST design added a maximal entropy classifier for separating aspects or opinion words more precisely. Those designs were assessed on review regarding restaurants and electronic devices quantitatively as well as qualitatively. The experiential outcomes evinced that the designs outperformed existent baselines and were competent to recognize fine-grained aspects and opinions but the improvement was still needed.

Rajkumar et al. [29] rendered a '2' ML approaches say Naïve Bayes (NB) and SVM for performing SA on reviews of a specific product. In those approaches, the dataset was gathered as of Amazon, which comprised reviews regarding Laptops, Cameras, Mobiles, Tablets, video surveillance, and TVs. Subsequently, stemming, stop word removal, and also punctuation marks removal were executed and it was transmuted into a bag of words. This dataset was contrasted to opinion lexicons, that is, 4783 negative and 2006 positive words with sentiment scores intended for every sentence were evaluated. Utilizing score and disparate features, the NB along with SVM were employed and diverse accurateness was computed. The ML approaches proffered the good outcomes to categorize product reviews. NB got 98.170% accuracy and SVM got 93.54% accuracy for Camera—related Reviews. The approach utilizes the SVM, which encompasses several key parameters that are required to be set properly for attaining the best classification outcomes. Thus, the SVM renders lower accuracy in classification.

Sumbal Riaz et al. [30] recommended an approach termed text mining for examining customer reviews to ascertain the customers' opinions and executed the SA on the

massive dataset of product (6 sorts) reviews proffered by disparate customers on the internet. In this approach, SA was employed at the phrase level instead of document-level for computing every term's SP. Then key graph keyword extraction was used aimed at extracting keywords as of each document with high-frequency terms and the intensity of SP by gauging its strength was evaluated. The k-means clustering was utilized for grouping data on the base of sentiment strength value. Those values were contrasted to the star rating of the same data and the excellent and neutral sentiment toward products was examined. The approach uses the clustering which may bring about over clustering.

Satuluri Vanaja and Meena Belwal [31] rendered an Aspect-level SA, which was attained by Identification, aggregation, and Classification. The preprocessing includes Parts-of-Speech tagging to every word in each sentence, extracting frequently used words, removing stopping or unwanted words and adjective extraction from the sentences. The classification was executed utilizing '2' ML algorithms, NB and SVM classification algorithm and the performance were contrasted centered on Recall, F1 measure, and Precision. The outcomes evinced that it attained more accurateness from the NB when weighed against the SVM. The SVM approach was not apt for large datasets.

Wei Zhang et al. [32] propounded an emotion classification algorithm grounded on SVM as well as latent-SA (LSA). Primarily, Psychology and NLP were integrated to divide the emotions in the online reviews onto '4' categories: a. happiness, b. hope, c. disgust, and d. anxiety. Subsequently, the LSA approach was utilized for optimizing the text feature extraction and employed the SVM as a classifier for ameliorating the emotional classification accurateness and computational efficacy. The experiential outcomes evinced that the model could effectually compute online reviews. Context meanings of data with DL algorithms were utilized for combining the reviews theme, sentiment classification and product characteristics for further enhancing the multiple class emotional detection accurateness. The approach employs only a less number of data for analyzing, which is not efficient.

Barkha Bansal and Sangeet Srivastava [33] rendered a Hybridized Attribute-centric Sentiment Classification (HABSC) for infusing domain-specific knowledge and collecting the implicit word relations. This approach found the utmost frequent bi-gram as well as tri-gram in the corpus, followed by POS tagging for retaining opinion words and aspect descriptions. Subsequently, it has deployed TFIDF for signifying every document, followed by automatic extraction of an optimal topic. All the adverbs and adjectives were labeled utilizing pre-existing lexicon and domain-related knowledge. The method's efficacy was tested utilizing datasets. The outcomes evinced that the classification accurateness of HABSC exceeded for disparate methods and also evinced less computational time as contrasted to distributed vectorization frameworks. The approach was not effective in detecting the attribution, and it has a high computational time.

Chonghui Guo et al. [34] examined a ranking approach via online reviews grounded on diverse aspects of variant products that integrated the subjective as well as objective sentiment values. Primarily, the product's sentiment value was evaluated by ascertaining the weights of those aspects with the LDA topic design. At the time of this process, the realistic meaning of every single aspect was as well summarized. Subsequently, consumers' personalized preferences were regarded whilst evaluating the total scores of variant products. Meanwhile, comparative superiority in-between every '2' products also added

into final scores. By utilizing the Page Rank algorithm, the attained final score of every product was evaluated as of the constructed graph. The outcome elucidated that whilst regarding only objective sentiment values of the product, the ranking outcome attained by this approach had a good correlation with the primary sales orders. But the system used the LDA, which was sensitive to overfit, and validation of LDA models was at least problematic.

Jian-Wu Bi et al. [35] offered a method for demonstrating the SA outcomes relied on the interval type-2 fuzzy numbers that considered the accuracy rates. Initially, the work showed the demonstration of SA outcomes attained from a large amount of online analyses as interval type-2 fuzzy numbers that were the initial effort to present the SA outcomes by considering its accuracy rates, and the first to get the decision data in a big data environment. This method had a clear logic as well as provided with a good foundation for conducting many types of management decision investigation that were relied on online reviews. Subsequently, the pertinent theoretical investigation was offered for the constructed interval type-2 fuzzy numbers. Finally, the system's performance was estimated that rendered better outcome.

Proposed methodology

An imperative manner through which the meaningful detail is mined from online is by SA of the reviews (online). The human's sentiments, emotions, opinions, et cetera concerning the products are expressed in the form of customer reviews as well as star ratings, which are being analyzed by a machine i.e. ML approach. The RA of online products will improve product quality as well as influence the purchase decisions of the consumers. Therefore, product review analysis is a universally accepted platform in which the consumer could effortlessly be conscious of their needs. For the SA of online products, though several ML techniques were suggested in the past, those techniques have only encompassed limited features and also didn't focus on the future prediction of online products by considering the user review comments. To trounce such drawbacks, two methodologies are proposed i.e. DLMNN and IANFIS for RA and future prediction of online product. The FRD is employed for the proposed SA. The data values as of the dataset are separated into '3' scenarios such as GB, CB, and CLB. Afterward, they are separated as positive, negative, as well as neutral with the help of DLMNN. Firstly, the efficient RA stage is performed for these three scenarios. For the initial one i.e. GB, centered on the score value of user ratings, the polarity score (PS) is computed followed by which it is classified as positive, negative, and neutral using DLMNN. After that, for CB and CLB, Preprocessing, FE, FS, and Review Classification are performed separately for both. In CB, only the comments given by the user are considered, whereas, CLB, user ratings along with comments are considered for the RA. The future prediction of the products is performed subsequent to these classifications (i.e. 3 scenarios). For which, the weighting factor of the pre-processed data is estimated by performing '5' operations, like keyword frequency, identification positive and negative word, computation of support, computation of confidence, as well as entropy estimation. And finally, the future prediction of the product is done with the help of IANFIS. The proposed method's architecture is exhibited in Fig. 1.

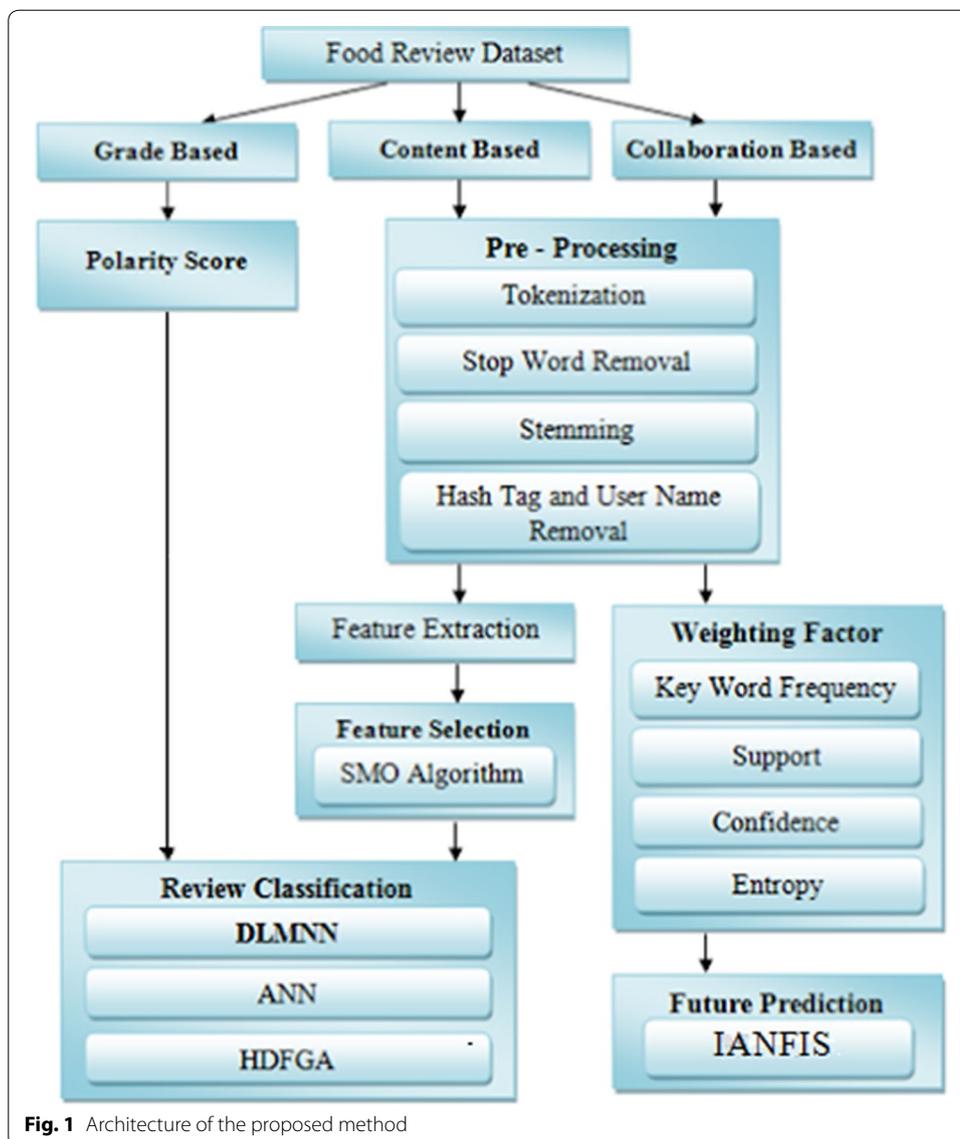


Fig. 1 Architecture of the proposed method

Food review dataset

The FRD is taken as the input in the case of the proposed technique. And this dataset has around 5 lakhs records that are openly available online. The dataset in them consists of features like Product Id, Profile Name, User Id, Helpfulness Denominator, Helpfulness Numerator, and Score Time Summary Text. 80% of data is given for training and the remaining data (i.e. 20%) is taken for testing. The RA is done centered on three scenarios (i.e. are GB, CB, and CLB) of the dataset for the online product. The GB encompasses only the information regarding the ratings of the online products, which varies for each product in the dataset. The CB encompasses the information relating to the comments of the products written by the customer. Finally, the CLB encompasses ratings as well as customer written reviews. The product’s review was analyzed by utilizing these three. In the proposed technique, the RA is done

separately for these three scenarios and inputted to the DLMNN for review classification. A meticulous elucidation of the scenarios is exhibited below.

GB scenario

In the GB scenario, the PS is computed for every product and later-on given to DLMNN for classification. Centered on the ratings (stars) that are clicked by the buyer for the certain product, the PS for that product is computed. If 4 or 5 stars are given to a product, then the PS will be 1; if it is 3, then the PS will be 0.5 and if it is 2, then the PS will be 0. The sentiment of the product is classified based on these PS. These computed PS of the product is given as input to the DLMNN classifier. DLMNN classifies it into (i) positive, (ii) negative, and (iii) neutral. A positive outcome indicates that the product encompasses lots of positive stars and also a good one. Negative results imply that the product has hordes of negative stars and is a bad one. Finally, neutral indicates that the product has positive as well as negative stars and is a neutral one.

CB scenario

For the RA of CB, the data values in the dataset are processed in four steps, such as (i) pre-processing, (ii) feature extraction, (iii) feature selection (iv) classification, which are elucidated briefly in the below subsection.

Preprocessing

In this stage, the comments of the online products stored in the dataset are pre-processed by performing tokenization, stemming, stop word removal, and hashtag and username removal. First, the pre-processing step carried out the tokenization, which is fundamentally the procedure of splitting the text into a compilation of meaningful pieces (tokens). For instance, a chunk of text is split into words or sentences. Then, the common words that are not useful for learning will be removed, which is denoted as the stop word removal. A word, like “the” or “and”, can be removed by comparing text with a stop words list. Next, stemming process is taken. Stemming stands as a process in which the words are decreased to its root by means of eradicating the inflection via dropping redundant characters that is typically a suffix. Finally, URL, Hashtag and username are removed because Hashtags are particularly used in social networks to mark keywords in messages, which make it simple to find. In this phase, the hashtags marked with a number sign (#) in front of a word are removed from an entered text. Next, if any URL or username is present, then that will also be removed and finally, the elongated word is trimmed.

Feature extraction

Here, the features, for instance, positive emoticon count, negative emoticon count, exclamation mark count, question mark count, positive gazetteer words occurrence count, negative gazetteer words occurrence count, unigrams, bigrams, trigrams, n-grams, and part-of-speech tag from the preprocessed data are extracted.

Feature selection

After FE, by utilizing the Spider Monkey Optimization Algorithm (SMOA), the essential features are chosen centered on the fitness values of extracted features. The brief explanation about the SMOA is given as follows,

SMOA is a population-centered stochastic algorithm. SMOA is enthused by the smart food foraging behavior of the spider monkeys (SM). The SM's foraging behavior shows that these SM comes under the class of fission–fusion social structure (FFSS) centered animals. Thus, the optimization algorithm centered on the SM's foraging behavior could well be explained better with reference to FFSS. The subsequent are the chief features of the FFSS.

1. Female monkeys lead a pack of 40–50 members to forage food. The group leader is liable for searching the food sources.
2. And if the group leader couldn't able to find adequate food for the entire group members, then she splits them into subgroups of 3 to 8 monkeys each to decrease the competition. These small groups search for food separately.
3. These Sub-groups will also be guided by a female (i.e. local leader). This local leader will make the decision for planning an effectual foraging route every day.
4. To exchange data on the subject of food availability as well as territorial boundaries, the sub-groups communicate with one another.

This algorithm executes six stages. They are local leader stage (LLS), global leader stage (GLS), global leader learning stage (GLLS), local leader learning stage (LLLS), local leader decision stage (LLDS), global leader decision stage (GLDS). Before executing these, the SM population is initialized. A meticulous explanation is provided below, Initialization: The solution of every monkey is a D -dimensional vector in which D implies the number of parameters (variables) to be optimized. The SM is initialized with the equivalently distributed values betwixt the minimum and maximum values by the SMO.

$$sm_{ij} = sm_{mnj} + rd(0, 1) * (sm_{mxj} - sm_{mnj}) \quad (1)$$

Where, sm_{ij} is the j th dimension of the i th spider monkey, sm_{mxj} and sm_{mnj} signifies the maximum and minimum limits of that free parameter.

Local leader phase (LLP): The SM centered on the experience of the local leader and the monkeys belonging to the same sub-group update their values. The new solution's fitness is calculated. If it is more than the original solution, then the monkey updates its solution. The position update of i th SM that is the part of m th group is stated as

$$sm_{newij} = sm_{ij} + rd[0, 1] * (ll_{mj} - sm_{ij}) + rd[-1, 1] * (sm_{rj} - sm_{ij}). \quad (2)$$

Where sm_{ij} denotes i th SM in j th direction, ll_{mj} denotes ' j th' dimension of the m th local group leader position. sm_{rj} specifies the ' j th' dimension of the r th SM that is randomly selected from m th group in order that $r \neq i$ in j th dimension.

Global leader phase (GLP): Subsequent to LLP, the SM centered on the GL along with local group member's experience updates their positions. This is performed for better convergence. It is estimated by utilizing the below equation.

$$sm_{newij} = sm_{ij} + rd(0, 1) \times (gl_j - sm_{ij}) + rd(-1, 1) \times (sm_{rj} - sm_{ij}). \quad (3)$$

Where gl_j denotes the GL location in 'jth' dimension and $j \in \{1, 2, \dots, D\}$ specifies the arbitrarily selected index. The SM probability is selected for updating their position, which is proportional to its fitness value in order that the best monkey encompasses more chance of being selected and improving itself. The probability is computed utilizing its fitness, which is shown as.

$$p_{ri} = 0.9 \times \frac{F_i}{F_{\max}} + 0.1 \quad (4)$$

Global leader learning (GLL): The global leader position is updated by searching all the solutions. The position of SM with the highest fitness function is taken as the global leader's position. Also, it is checked that the global leader's position has changed or not. If the position didn't change, then the Global Limit Count's value is increased by one.

Local leader learning (LLL): Here, the greedy search is performed within the sub-groups. The SM with the uppermost fitness function is picked as the local leader. The old along with updated position of the local leader is also verified. If it isn't changed, then the Local Limit Count is augmented by one.

Local leader decision (LLD): This phase aids in the re-initialization of a group if its local leader does not update its position of the specified local leader limit. Re-initialization of the group might cause a member of the group to enter into the infeasible region from a feasible region. The LL position is computed utilizing the formula.

$$sm_{Newij} = sm_{ij} + rd(0, 1) \times (gl_j - sm_{ij}) + rd(-1, 1) \times (sm_{ij} - ll_{mj}). \quad (5)$$

Global leader decision (GLD): Here, the whole swarm is bifurcated into groups if the GL isn't updated for the specified GL limit. Amid this phase, local leaders are decided for newly generated sub-groups utilizing the LLL process.

DLMNN classifier

After feature selection, rank the chosen features (words) utilizing the sentiwordnet dictionary. SentiWordNet, derived as of the WordNet database, is basically an opinion lexicon where every term is related to numerical scores signifying positive sentiment and negative sentiment information. The feature word encompasses positive and even negative ranking. These score values of the products are inputted to DLMNN and are classified as positive, neutral and negative. Positive signifies that the product is a good product and has more positive comments as of the customers. Negative signifies that the product is not a good one and receives lots of negative comments and the product has fewer stars. Also, neutral indicates that the product is an average product that received positive together with negative comments as of the customer.

Each input is sent to a discrete node existent in the input area of the DLMNN. The randomly assigned values called weights are associated to each input. The nodes in the hidden layer (HL) termed hidden nodes perform the function of adding the product of the input value and weight vector of all the input nodes which are connected to it. In DLNN, the weight values are effectively optimized utilizing the Hybrid Dragonfly-Genetic Algorithm (HDF-GA) which is called DLMNN. Random weight value gives a more back-propagation

process to achieve the result, and hence the optimization is performed in the proposed method. The activation operation is then implemented and this layer's output is transported to the consecutive layer. The steps in DLMNN classification are expounded below,

Step 1: Initially assign the ranked data values and their respective weights utilizing Eq. (6) and (7).

$$R_i = \{R_1, R_2, R_3 \dots R_n\}. \quad (6)$$

$$w_i = \{w_1, w_2, w_3 \dots w_n\}. \quad (7)$$

where R_i signifies the n number of ranked data's like $R_1, R_2, R_3 \dots R_n$ and w_i denotes the weight value of R_i which includes ' n ' number of weights namely $w_1, w_2, w_3 \dots w_n$ for corresponding $R_1, R_2, R_3 \dots R_n$.

Step 2: Multiply the ranked input data with the randomly chosen weight vectors and then summate those values, and it is expressed as,

$$S_m = \sum_{i=1}^n R_i w_i \quad (8)$$

where S_m indicates the summed value.

Step 3: Evaluate the activation function Af_i utilizing the Eq. (9),

$$Af_i = S_i \left(\sum_{i=1}^n R_i w_i \right) \quad (9)$$

and

$$B_i = e^{-R_i^2} \quad (10)$$

where B_i specifies the exponential of R_i .

Step 4: Compute the next HL's output utilizing Eq. (11).

$$Y_i = b_i + \sum B_i w_i \quad (11)$$

where, b_i –Bias value, w_i –Weight between the input layer and HLs.

Step 5: Perform the steps from 2 to 4 for every layer of DLMNN. Lastly, summate all the existent input signals' weights for attaining the output layer neurons' value which is expressed as.

$$U_i = b_i + \sum O_i w_j. \quad (12)$$

where, O_i –Value of the layer that precedes the output one, w_j –Weights of the HL, U_i –Output unit.

Step 6: Contrast the network output to the target value and find the error signal which is the difference of those '2' values. This value is mathematically signified utilizing (13),

$$e_s = A_i - U_i. \quad (13)$$

where, e_s –Error signal, A_i –aimed target output.

Step 7: Here, the output unit is weighted against the targeted value. Determine the related error. Compute δ_i centered on this error.

$$\delta_i = e_s[f(A_i)]. \quad (14)$$

Step 8: assess the weight correction employing the backpropagation methodology utilizing Eq. (15).

$$C_i = \lambda \delta_i(R_i). \quad (15)$$

where C_i implies the weight correction, λ signifies the momentum term, and δ_i implies the error that is distributed in the network. The weight values are optimized utilizing an HDF-GA algorithm. The elucidation for HDF-GA is proffered below.

The Dragon fly (DF) algorithm is mainly inspired as of the natural static as well as dynamic swarming behaviors of DFs. The '2' necessary phases of optimization like (i) exploration and (ii) exploitation are intended by designing the social interaction of DFs during navigation, food search, and avoidance of enemies while swarming. Here, '3' main rules (nature of swarms) are followed by DFs that are, separation (S_p) –The DFs avoid one another on account of collision in a stationary position as of the neighborhood, Alignment (A_l) –Each DF's velocity coordinates with another one in neighborhood and Cohesion (C_h) –The DFs fly towards the center of a group of neighborhoods.

The normal dragonfly algorithm encompasses the convergence speed problem. So in the proposed work, the crossover and mutation process of the Genetic Algorithm (GA) is hybridized with the conventional DF algorithm to attain effective optimization outcomes to form HDF-GA. The proposed hybridization algorithm renders a better optimum result with better convergence speed. The process of the HDF-GA are expounded below.

Initially assign the size of n DFs as N and the i th DF's position is formulated as,

$$M_i = (m_i^1, m_i^d, \dots, m_i^N). \quad (16)$$

where, $i = 1, 2, 3 \dots N$, m_i^d –Position of the i th DF in d th dimension of the search spaces N –Number of search agents.

After initialization, the separation, alignment, cohesion, attraction and distraction are computed. First, Compute S_p utilizing Eq. (17).

$$S_p(i, t) = - \sum_{j=1}^N M(i, t) - M(j, t) \quad (17)$$

where, $S_p(i, t)$ –Separation motion for i th individual at the t th iteration $M(i, t)$ –Current individual ' i 's position at the t th iteration $M(j, t)$ –Neighboring individual j 's position at the t th iteration.

Second, compute the velocity matching of individuals amongst the neighbourhood utilizing Eq. (18).

$$A_l(i, t) = \frac{\sum_{j=1}^N v(j, t)}{N} \quad (18)$$

where, $A_l(i, t)$ –Alignment motion of the i at the iteration t , $v(j, t)$ –Velocity of j at the t th iteration.

Third, compute the cohesion of individuals in the direction of the center of mass of the neighbourhood utilizing Eq. (19).

$$C_h(i, t) = \frac{\sum_{j=1}^N M(j, t)}{N} - M(i, t) \quad (19)$$

Fourthly, compute the motion of attraction towards the food source $A_r(i, t)$ using Eq. (20).

$$A_r(i, t) = M(f, t) - M(i, t) \quad (20)$$

Here, $M(f, t)$ indicates the food source position at the iteration t . Individual DFs with the best objective function (OF) up to the current iteration would be regarded as food.

Fifthly, compute the motion of distraction $D_r(i, t)$ against the enemy utilizing (21).

$$D_r(i, t) = M(e, t) - M(i, t) \quad (21)$$

where, $M(e, t)$ signifies the enemy distraction motion of i at iteration t . Individual DFs with the worst OF up to the current iteration would be regarded as an enemy.

Then, 2 genetic operators “Crossover and mutation” are added and are applied when the DF does not have at least ‘1’ neighboring DF. This brings effective optimization. Here, the 2-point crossover type is used and is executed utilizing the crossover points.

$$c_1 = \frac{|X_{t+1}|}{3} \quad (22)$$

$$c_2 = c_1 + \frac{|X_{t+1}|}{2} \quad (23)$$

where c_1 and c_2 indicates the selected ‘2’ crossover points. During mutation, replace the number of genes as of every chromosome with new genes. The replaced genes are arbitrarily created genes without any repetition in the chromosome. The chromosomes indicate a parameter set that gives the solution in this proposed system.

After crossover, mutation update the DFs’ position in a search space and simulate their movements utilizing ‘2’ vectors like position vector Y and step vector ΔX . The direction of the DFs’ movement is specified by ΔX and is formulated as,

$$\Delta X_{t+1} = (sS_p + aA_l + cC_h + fA_r + eD_r) + w\Delta M_t. \quad (24)$$

where, t is the current iteration, s is the separation weight, S_p symbolizes the separation of the i th individual, a indicates the alignment weight, A_l alludes to the alignment of the i th individual, c cohesion weight, C_h signifies the cohesion of the i th individual, f is the food factor, A_r specifies the food source of the i th individual, e is the enemy factor, D_r ,

signifies the position of the enemy of the i th individual, indicates the weight of inertia and t signifies the iteration counter. At last, the position vector could be evaluated as of the value of the step vector. It is computed as,

$$Y_{t+1} = Y_t + \Delta Y_{t+1} \tag{25}$$

where, ΔY is the Step Vector, t is the current iteration and Y is the Position Vector. The HDF-GA pseudo-code is proffered in Fig. 2.

CLB scenario

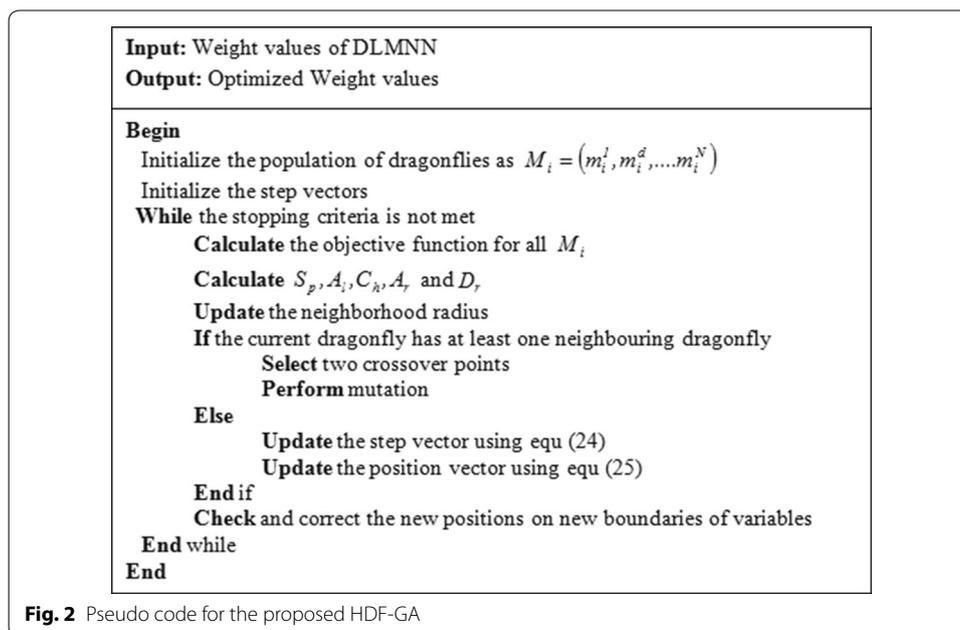
Here, the RA of online products is performed by collaborating GB with CB scenarios. Here, both grades and reviews given by the customer for a specific product are regarded. Contrasting the outcomes of both GB and CB, the CLB scenario is analyzed.

Future prediction

After RA, the future prediction of the online products is done with the help of the IAN-FIS technique. Firstly, the weighting factor of the online product has been gauged by performing five steps, such as, keyword frequency, identification, support, confidence and entropy. First, identify the frequent keyword that means the number of occurrences of a particular keyword on the dataset and is expressed as,

$$K_{f_i} = \{K_{f_1}, K_{f_2}, K_{f_3}, \dots, K_{f_n}\} \tag{26}$$

After the extraction of frequent keywords, the positive and negative keywords are identified as of the previously found frequent keywords utilizing sentiwordnet dictionary. After that, the support, confidence and entropy values of those identified words are computed. The support of the keyword is extracted. The support signifies the percentage



of transactions in the database which comprises every keyword on the database. For an association rule $(K_1 \rightarrow K_2)$ (K_1, K_2 -keywords), the support is formulated as,

$$S_p = \text{Support}(K_1 \rightarrow K_2) = P(K_1 \cup K_2) \quad (27)$$

Then, the confidence is extracted for the keywords, which specify the percentage of transactions on the database with the keywords (K_1, K_2). The confidence is evaluated utilizing the conditional probability which is further expressed in respect of keyword support and is proffered as,

$$C_f = \text{Confidence}(K_1 \rightarrow K_2) = P(K_1/K_2) = \frac{P(K_1 \cup K_2)}{P(K_1)} \quad (28)$$

where, $P(K_1 \cup K_2)$ —Number of transactions with the keywords K_1 and K_2 , $P(K_1)$ —Number of transactions containing a keyword K_1 .

Finally the entropy, which indicates the average of the information attribute. The information quantity of every attribute generates an arbitrary variable, which may be the anticipated or average value and is regarded as entropy. Entropy calculates as,

$$E_i = - \sum_{p,f=1}^n S_p \log_2 C_f \quad (29)$$

where, S_p denotes support and C_f denotes the confidence. These computed entropy values are inputted to the IANFIS classifier for future prediction.

IANFIS

The IANFIS algorithm is employed for classifying the future prediction of the product into high and low. High indicates that the future demand is high for the product, whereas, low symbolizes that the future demand is low for the product. IANFIS takes the entropy values of frequent positive keywords and negative keywords as input. The positive keyword with the maximal entropy value signifies that the product has higher future demand. Similarly, the negative keyword with the maximal entropy value signifies that the product has lower future demand. Grounded on this future demand of a specific product, all high-quality reviews of that product are visible on the top to the customers and also that product is suggested for them on the top of the existent product category list. The IANFIS is briefly explained below.

A sort of artificial neural network termed ANFIS (Adaptives Neuro-Fuzzy Inferences System) is centered on the Takagi–Sugenos fuzzy inference system. As it incorporates the neural networks as well as fuzzy logic principles, it comprises the potential of capturing their merits into a single model. Its inference system agrees with a compilation of “fuzzy IF–THEN rules” that have the learning competency of approximating nonlinear functions. In its architecture, five layers could be seen. Each layer is explained below.

First layer: It is the fuzzification layer, which takes the input and finds the membership functions (MFs) that belong to them.

Second layer: It is the rule layer, which is accountable for producing the Firing Strengths (FS) aimed at the rules.

Third layer: It normalizes the computed FSs by diving each value with the total FS.

Fourth layer: here, the normalized values are given as input (consequence parameter set) and the defuzzificated values returns as the output.

Fifth layer: It regards the defuzzificated values as input for returning the final output.

The conventional ANFIS model utilizes the bell MF. The bell MF has a symmetric form and cannot easily calculate the operations. So the proposed technique uses the Gaussian kernel Membership Function (GMF) for ameliorating the rule generation process's performance. The GMF has the smoothness along with concise notation. The '2' basic rules of IANFIS are,

Rule 1: If E_1 is P_i as well as E_2 is Q_i then,

$$Rules_i = sE_i + t_iE_{i+1} + u_i \tag{30}$$

Rule 2: If E_1 is P_{i+1} as well as E_2 is Q_{i+1} then,

$$Rules_{i+1} = s_{i+1}E_i + t_{i+1}E_{i+1} + u_{i+1} \tag{31}$$

where P_i, Q_i, P_{i+1} and Q_{i+1} specifies the fuzzy sets. E_i and E_{i+1} values represent the disparate entropy values attained as of the former step. $s_i, t_i, u_i, s_{i+1}, t_{i+1}$ and u_{i+1} values indicates the parameter set. The layers in IANFIS are expounded below individually.

Layer 1: Each adaptive node of layer-1 has a node function and they are adapted to a function parameter. It is formulated as,

$$IA_{1,i} = \mu_{f_i}(E_i) \tag{32}$$

Here, E_i -Input to 'i'node.

The input of MF gives the degree of member-ship value as the output for each node. The Gaussian kernel MF is the MF used in the proposed work and is specified in the succeeding equation.

$$\mu_{f_i} = \exp\left(-\frac{\|s_i - t_i\|^2}{2u_i^2}\right) \tag{33}$$

where, s_i, t_i and also u_i indicates the MFs' parameters that could alter the shape of the MF. The parameters are alluded to as the premise parameters.

Layer 2: Its fixed nodes produce the output in the form of the product of all existent incoming signals

$$IA_{2,i} = K_i = \mu_{f_i}(E_i) \times \mu_{f_i}(E_{i+1}). \tag{34}$$

The FS of a rule is the output for $IA_{2,i}$.

Layer 3: Its fixed nodes are tagged as N. The i th node evaluates the ratio of the i th rule's FS to the total of all rules' FSs.

$$IA_{3,i} = \bar{K}_i = \frac{K_i}{K_i}, \quad i = 1, 2, \dots, 6 \tag{35}$$

The outputs are termed normalized FSs for convenience.

Layer 4: It comprises adaptive nodes with node function and is formulated as,

$$IA_{4,i} = \bar{K}_i \cdot Rules_i \quad (36)$$

Here, \bar{K}_i implies the normalized FS as of the preceding layer and $Rules_i$ signifies the rule of the system. The parameters that are employed are termed as succeeding parameters.

Layer 5: It has a single fixed node and it produces the overall output by summing the existing incoming signals. Here, the circle node is labeled as \sum .

$$IA_{5,i} = \sum_i K_i Rules_i = \frac{\sum_i K_i Rules_i}{\sum_i K_i} \quad (37)$$

Results and discussion

The proposed DLMNN for SA of online product reviews and IANFIS for future predictions are implemented in the working platform of JAVA. Here, the three scenarios like GB, CB, and CLB, which are utilized for RA of online product-reviews, are contrasted during performance analysis. The IANFIS approach used for future prediction of online product is weighed against the existing ANFIS. For both RA and future predictions, the ML approaches are implemented. The performance metrics like precision (p_s), f-score (f_s), recall (r_k), and accuracy (a_c) is used to contrast the proposed schemes' performance.

Performance analysis of GB, CB, and CLB using DLMNN

Here, the performance analysis of GB, CB, and CLB scenarios using DLMNN is made in respect of the performance measures say p_s , r_k , f_s , and a_c which is evinced in Table 1.

Table 1 compares the performance of the GB, CB, CLB scenarios of RA using DLMNN regarding (a) p_s , and r_k , and (b) f_s , and a_c . For comparing those three scenarios, 5000 data are regarded. For 1000 data, the GB and CB gives the p_s of 86.3266 and 87.9988, r_k of 92.4555 and 92.1125, f_s of 90.3324 and 91.0455, and a_c of 96.8543 and 89.7764, whereas, the CLB scenario gives the values of 95.4534, 96.1128, 95.0455, and 95.7764 for p_s , r_k , f_s , and a_c respectively, which are higher than GB and CB scenarios. The chief aim of the proposed methodology is to ameliorate the SA system's accuracy, and in this way, the system can attain higher accuracy centered on the CLB scenario. Likewise, for 5000 data, the CLB scenario gives the values of 96.4555, 96.8322, 96.8543, and 96.6654 for p_s , r_k , f_s , and a_c respectively but the GB and CB scenarios gives 88.3447 and 88.8455 for p_s , 91.8876 and 90.8322 for r_k , 91.6355 and 90.8541 for f_s , and 86.67,845 and 96.6654 for a_c respectively, which are lower than CLB scenario. Contrasted to GB, the CB scenario attains good outcomes for all performance measures. Similarly, for the remaining number of data (i.e. 2000, 3000 and 4000), the proposed system gives higher accuracy centered on the CLB scenario. Thus, it inferred that the proposed system attains better performance centered on the CLB scenario. The Table 1 is graphically evinced in Fig. 3.

Performance analysis of IANFIS

Here, the IANFIS is used for future forecast of online product. For performance analysis, the proposed IANFIS is contrasted to the existing ANFIS, Artificial Neural Network (ANN), as well as Deep Neural Network (DNN) regarding p_s , r_k , f_s , and a_c which is displayed in Fig. 4.

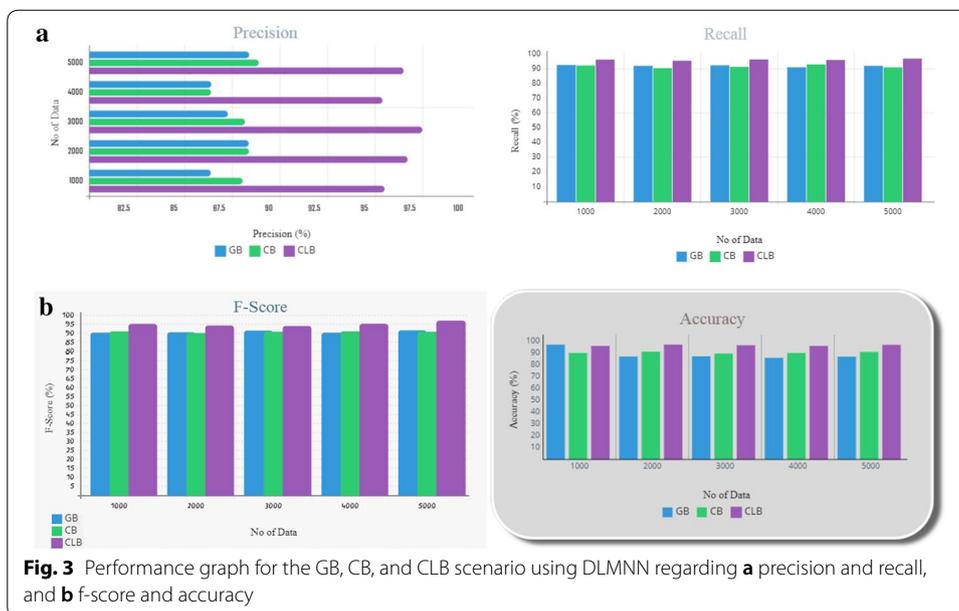
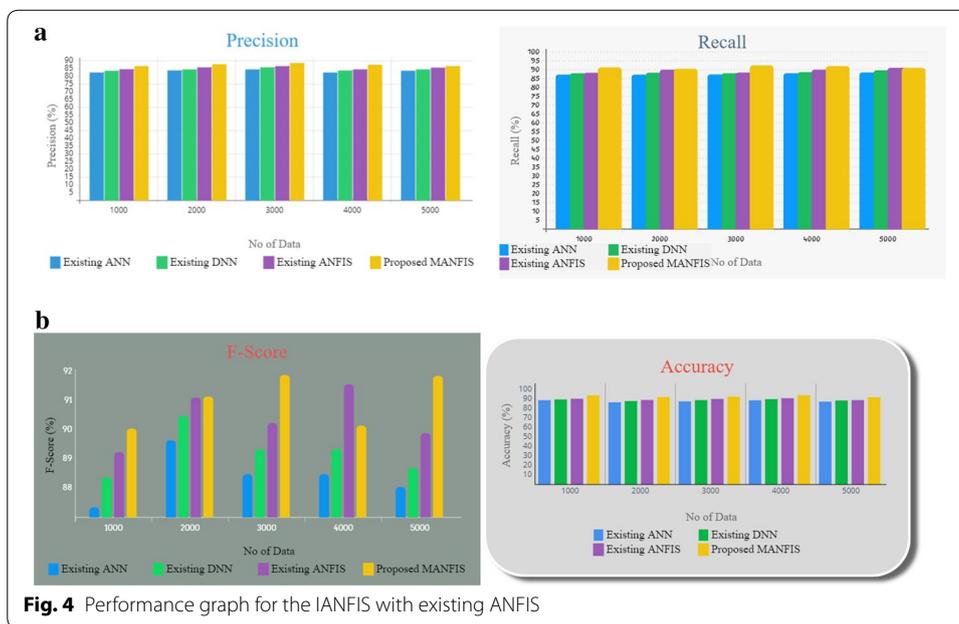


Table 1 Performance comparison of GB, CB, and CLB

No of data	Performance measures					
	p_s			r_k		
	GB	CB	CLB	GB	CB	CLB
a						
1000	86.3266	87.9988	95.4534	92.4555	92.1125	96.1128
2000	88.3226	88.3322	96.6644	91.8857	90.3534	95.3534
3000	87.2247	88.1244	97.4354	92.2333	91.2324	96.2324
4000	86.3577	86.3365	95.3422	90.8837	92.8323	95.8323
5000	88.3447	88.8455	96.4555	91.8876	90.8322	96.8322
No of data	Performance measures					
	f_s			a_c		
	GB	CB	CLB	GB	CB	CLB
b						
1000	90.3324	91.0455	95.0455	96.8543	89.7764	95.7764
2000	90.5688	90.1454	94.1454	86.76454	90.8845	96.8875
3000	91.4788	90.8807	93.8875	86.97552	89.3445	96.3447
4000	90.3357	91.1043	95.1435	85.67556	89.7764	95.7761
5000	91.6355	90.8541	96.8543	86.67845	90.6654	96.6654

Figure 4 evinces the performance comparison graph for the proposed IANFIS, ANFIS, ANN and the DNN concerning the performance measures say (a) p_s , and r_k and (b) f_s , and a_c . Totally 5000 data are taken for performance comparison of the proposed and existing methods. The graph is plotted for 1000–5000 data. For 1000 data, the proposed IANFIS gives 86.4534 of p_s , 91.1128 of r_k , 90.0455 of f_s , and 93.7734 of a_c . But the existing systems, such as ANFIS, ANN, and DNN have poor performance than the proposed system centered on the recall, precision, f-measure as well as accuracy metrics. And also



for the remaining 2000, 3000, along with 4000 data, the IANFIS attains the best results. Hence, from the comparison, it is deduced that the proposed IANFIS attains pre-eminent performance for the future prediction of online products when weighed against the existing ANFIS.

Conclusion

A DLMNN methodology is proposed aimed at SA of online products review and an IANFIS methodology is proposed aimed at future prediction of online product. The performance of both the proposed methodologies is analyzed. The proposed DLMNN is employed for three scenarios (GB, CB, and CLB) of RA. The comparison of those three scenarios for disparate numbers of data (from 1000 to 5000) concerning the performance measures of p_s, r_k, f_s , and a_c , is done. While comparing the ‘3’ scenarios, the CLB scenario attain the best outcomes for product RA. And, while contrasting the IANFIS for future prediction against the existing ANFIS, the proposed IANFIS attains the highest values for p_s, r_k, f_s , and a_c . Hence, from the performance analysis, the paper infers that the proposed CLB scenario and IANFIS performed-well for SA and future prediction of online products. The system has a shortcoming such that the keyword processing only identifies the sentiment reflected in a particular word; it typically fails at providing all of the elements necessary to understand the complete context of the entire piece. In the future, the proposed system can be extended by solving the keyword processing problem and improve the performance using a hybridization algorithm in the future prediction process.

Abbreviations

CB: Content based; CLB: Collaboration based; DLMNN: Deep learning modified neural network; DNN: Deep neural network; GB: Grade based; IANFIS: Improved adaptive neuro fuzzy inference system; NLP: Natural language processing; OM: Opinion mining; RA: Review analysis; SA: Sentiment analysis.

Acknowledgements

The authors thank the anonymous reviewers for their helpful suggestions and comments.

Authors' contributions

SK designed the idea, prepared the draft of manuscript and performed experiments. SK and MIS wrote the article. Both authors read and approved the final manuscript.

Funding

The research received no external funding.

Availability of data and materials

The datasets used and/or analysed during the current study are available from the corresponding author on reasonable request.

Competing interests

The authors declare that they have no competing interests.

Author details

¹ Department of Computer Science, Mother Teresa Women's University, Kodaikanal, India. ² FESAC, Information Technology, Pentecost University College, Accra, Ghana.

Received: 27 December 2019 Accepted: 7 May 2020

Published online: 19 May 2020

References

1. Verma JP, Patel B, Patel A. Big data analysis: recommendation system with Hadoop framework. *IEEE*. 2015. <https://doi.org/10.1109/CICT.2015.86>.
2. Choudhary M, Choudhary PK. Sentiment analysis of text reviewing algorithm using data mining, In *International Conference on Smart Systems and Inventive Technology (ICSSIT)*, pp. 532–538, 2018.
3. Khuc VN, Shivade C, Ramnath R, Ramanathan J. Towards building large-scale distributed systems for twitter sentiment analysis. *Proceedings of the 27th annual ACM symposium on applied computing*, ACM, 2012.
4. Vijayakumar S, Vijayakumar V, Logesh R, Indragandhi V. Unstructured data analysis on big data using map reduce. *Procedia Computer Sci*. 2015;50:456–65.
5. Gondaliya V, Mandaviya K. An improved approach for online trending forum detection based on sentiment analysis. *Int Conference Inventive Comput Technol (ICT)*. 2016;3:1–5.
6. Fang Xing, Zhan Justin. Sentiment analysis using product review data. *J Big Data*. 2015;2(1):5.
7. Vinodhini G, Chandrasekaran RM. Sentiment analysis and opinion mining: a survey. *Int J*. 2012;2(6):282–92.
8. Severyn A, Moschitti A. Twitter sentiment analysis with deep convolutional neural networks. *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, 2015.
9. Kumar KS, Desai J, Majumdar J. Opinion mining and sentiment analysis on online customer review. In *IEEE International Conference on Computational Intelligence and Computing Research (ICCR)*, pp. 1–4, 2016.
10. Jagbir Kaur, and Meenakshi Bansal, "Multi-layered sentiment analytical model for product review mining", In *Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, pp. 415-420, 2016.
11. Singla Z, Randhawa S, Jain S. Statistical and sentiment analysis of consumer product reviews. In *8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1–6, 2017.
12. Hemmatian F, Sohrabi MK. A survey on classification techniques for opinion mining and sentiment analysis. *Artificial Intelligence Review*, pp. 1–51, 2017.
13. Feldman Ronen. Techniques and applications for sentiment analysis. *Commun ACM*. 2013;56(4):82–9.
14. Trupthi M, Suresh P, Narasimha G. Sentiment analysis on twitter using streaming API. In *IEEE 7th International Advance Computing Conference (IACC)*, pp. 915–919, 2017.
15. Zheng Lijuan, Wang Hongwei, Gao Song. Sentimental feature selection for sentiment analysis of Chinese online reviews. *Int J Mach Learning Cybernetics*. 2018;9(1):75–84.
16. Richard S et al. Recursive deep models for semantic compositionality over a sentiment treebank. *Proceedings of the conference on empirical methods in natural language processing*, 2013.
17. Xavier G, Antoine B, Yoshua B. Domain adaptation for large-scale sentiment classification: A deep learning approach. *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011.
18. Rajalaxmi H, Seema S. Aspect based feature extraction and sentiment classification of review data sets using Incremental machine learning algorithm. In *Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEIICB)*, pp. 122–125, 2017.
19. Saleh MR, Martín-Valdivia MT, Montejo-Ráez A, Ureña-López LA. Experiments with SVM to classify opinions in different domains. *Expert Syst Appl*. 2011;38(12):14799–804.
20. Krishna MH, Rahamathulla K, Akbar A. A feature based approach for sentiment analysis using SVM and coreference resolution. In *International Conference on Inventive Communication and Computational Technologies (ICICT)*, IEEE, pp. 397–399, 2017.
21. Rana S, Singh A. Comparative analysis of sentiment orientation using SVM and Naive Bayes techniques. In *2nd International Conference on Next Generation Computing Technologies (NGCT)*, IEEE, pp. 106–111, 2016.
22. Vamshi KB, Pandey AK, Siva KA. Topic model based opinion mining and sentiment analysis. In *International Conference on Computer Communication and Informatics (ICCCI)*, IEEE, pp. 1–4, 2018.
23. Du X, Cai Y, Wang S, Zhang L. Overview of deep learning. In *31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, IEEE, pp. 159–164, 2016.

24. Yoon K. Convolutional neural networks for sentence classification, arXiv preprint arXiv, pp. 1408.5882, 2014.
25. Dos Santos C, Gatti M. Deep convolutional neural networks for sentiment analysis of short texts. Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, 2014.
26. Asad A, Siti MS, Shafaatunnur H, Jalil P. Deep learning-based sentiment classification of evaluative text based on multi-feature fusion. *Inf Process Manage*. 2019;6(4):1245–59.
27. Wei W, Gulla JA. Sentiment learning on product reviews via sentiment ontology tree. Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2010.
28. Tang Feilong, Luoyi Fu, Yao Bin, Wenchao Xu. Aspect based fine-grained sentiment analysis for online reviews. *Inf Sci*. 2019;488:190–204.
29. Jagdale RS, Shirsat VS, Deshmukh SN. Sentiment analysis on product reviews using machine learning techniques. In *Cognitive Informatics and Soft Computing*, Springer, Singapore, pp. 639–647, 2019.
30. Riaz S, Fatima M, Kamran M, Nisar MW. Opinion mining on large scale data using sentiment analysis and k-means clustering. *Cluster Comput*. 2019;22(3):7149–64.
31. Vanaja S, Belwal M. Aspect-level sentiment analysis on e-commerce data. In: international conference on inventive research in computing applications (ICIRCA), IEEE, pp. 1275–1279, 2018.
32. Zhang W, Kong SX, Zhu YC. Sentiment classification and computing for online reviews by a hybrid SVM and LSA based approach. *Cluster Computing*. 2018. pp. 1–14
33. Bansal Barkha, Srivastava Sangeet. Hybrid attribute based sentiment classification of online reviews for consumer intelligence. *Appl Intell*. 2019;49(1):137–49.
34. Guo Chonghui, Zhonglian Du, Kou Xinyue. Products ranking through aspect-based sentiment analysis of online heterogeneous reviews. *J Syst Sci Syst Eng*. 2018;27(5):542–58.
35. Bi Jian-Wu, Liu Yang, Fan Zhi-Ping. Representing sentiment analysis results of online reviews using interval type-2 fuzzy numbers and its application to product ranking. *Inf Sci*. 2019;504:293–307.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

Submit your next manuscript at ▶ [springeropen.com](https://www.springeropen.com)
