Sentiment Analysis of Online Product Reviews Using DLMNN and Future Prediction of Online Product Using IANFIS

Sasikala p (✉ shakthesasi@gmail.com)

Government Science College <https://orcid.org/0000-0002-0222-0459>

Mary Immaculate Sheela

Pentecost University College

Research

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)

Posted Date: January 2nd, 2020

DOI: <https://doi.org/10.21203/rs.2.19872/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Sentiment Analysis of Online Product Reviews Using DLMNN and Future Prediction of Online Product Using IANFIS

Sasikala P^{#1} , L.Mary Immaculate Sheela ^{*2}

^{#1} Department of Computer science, Mother Teresa Women's University, Kodaikanal, India.
shakthesasi@gmail.com

^{*2} Pentecost University College, Accra, Ghana. drsheela09@gmail.com

Abstract: Sentiment analysis or opinion mining is one of the major tasks of NLP (Natural Language Processing). It captures the user's opinion, feelings, and belief regarding the respective product especially to determine whether the user's attitude is positive, negative, or neutral. This analysis greatly helps the companies to make necessary changes in their product which in return can overcome the flaws that the product is facing and targets better customer satisfaction. Existing techniques for the sentiment analysis of online product reviews obtained low accuracy and also took more time for training. To overcome such issues in this paper, a DLMNN is proposed for sentiment analysis of online product review and IANFIS is proposed for future prediction of online product. Here, the sentiment analysis and future predictions are done on the products taken from the food review dataset. First, from the dataset, the data values are partitioned into GB, CB, and CLB scenarios and then the review analysis for each scenario is performed separately using DLMNN and they give the result as positive, negative, and neutral reviews for the product. After the process of review classification based on these three scenarios, the future prediction of the products is done by performing weighting factor and classification using IANFIS. Experimental results are compared with some existing techniques and the results show that the proposed method outperforms other existing algorithms.

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).*

1. INTRODUCTION

There are a huge number of brands available in the market, thus selecting one from many is a difficult job for a consumer. The buying pattern of customers is very much affected by the evolution of E-Commerce. Reviews that are available on Ecommerce are used by the buyers to make appropriate decisions. Social networking sites also provide a platform to write reviews for the product [1]. So people are posting more and more comments, this online data and online information are growing rapidly every day. So, extracting information from the internet is very difficult [2]. Analysis of positive and negative sentiments about each topic of the product is very useful to the customers and manufacturers. This can be achieved with the help of Sentiment Analysis (SA). SA is used to determine the mood of the writer or attitude of the speaker; it may be either positive or negative towards the product [3]. In SA, all product reviews to be summarized and sentiments are to be classified.

SA additionally known as opinion mining is the field of study that analyzes people's opinions, sentiments, evaluations, appraisals, attitudes, and emotion towards entities cherish products, services, organizations, and people [4,5]. SA basically has three stages: document, sentence and aspect level [6]. The document-level sentiment analysis checks the entire document and opinion is determined as positive, negative or neutral during the sentiment analysis process. The aspect level sentiment analysis finds the target of the opinion indicating that every opinion has a target. The bag-of-words method for sentiment analysis in which the relationships between words were not at all considered and a document is represented as just a collection of words [7]. There are two kinds of methods for sentiment analysis: semantic orientation approach and statistical machine learning approach. The semantic approach determines the sentiment of a document based on extracted sentimental words and phrases. The statistical approach determines the sentiment of a document based on extracted sentimental features and the machine learning approach. The statistical approach outperforms the semantic approach concerning the accuracy of sentiment analysis [8].

Most SA is based on supervised machine learning methods. Feature extraction and classifier design of texts play important roles in the methods. The different methods for feature selection includes Term Occurrence, Binary term occurrence, Term frequency and TF-IDF (term frequency-inverse document frequency) [9]. The TF-IDF method usually uses the sentiment lexicon to select feature words and calculate weights, which has been extensively applied to traditional natural language processing tasks [10].

In the past decades, several methods have been proposed for sentiment analysis. Most of those methods are based on the computational linguistic approach and machine learning approach such as Naive Bayes (NB) Maximum Entropy (ME) and Support Vector Machines (SVMs) [11]. These models are trained on the feature vectors derived as an output from the Latent Dirichlet Allocation (LDA) model and the sentiment in the text is classified as positive or negative. Although all of them obtained a good result, several methods in literature have shown that the machine learning approach exhibit higher performance than the computational linguistic approach [12].

However, in more recent years, the use of a new technique known as deep learning has attracted the attention of researchers, since it has obtained remarkable results for various NLP tasks [13]. Deep Learning is a part of machine learning architecture including multiple layers of perceptron inspired by the human brain [14]. There are various deep learning models such as deep neural networks (DNN), convolution neural networks (CNN), deep Restricted Boltzmann Machine (RBM), etc. However, while the use of deep learning models in NLP has illustrated very desirable results for many tasks, it seems that they have not yet reached the level to overcome the existing problems and there is room for improvement [15].

The draft structure for this paper is systematized as: Section 2 surveys the associated works regarding the method proposed. Sections 3, proffers a brief discussion about the proposed methodology, section 4, explores the Investigational outcome and section 5 briefly concludes the paper.

2. LITERATURE REVIEW

Feilong Tang *et al* [16] presented two generative models namely JABST and MaxEnt–JABST that extracted fine-grained aspects and opinions from online reviews. The JABST model extracted general and specific aspects and opinions jointly, with their sentiment polarity. Furthermore, the MaxEnt–JABST model added a maximum entropy classifier to separate opinion and aspect words more accurately. Those models were evaluated on reviews of electronic devices and restaurants qualitatively and quantitatively. The experimental results showed that the models outperformed state-of-the-art baselines and were able to identify fine-grained aspects and opinions.

Rajkumar S. Jagdale *et al.* [17] presented a two machine learning techniques say Vector machine (SVM) and Naïve Bayes (NB) for performing sentiment analysis on product reviews. In these techniques, the dataset was collected from Amazon, which contained reviews of Cameras, Laptops, Mobile phones, tablets, TVs, video surveillance. Then, stop word removal, stemming and punctuation marks removal were performed and it was converted into a bag of words. This dataset was compared with opinion lexicons i.e. 2006 positive words and 4783 negative words and sentiment scores for every sentence were calculated. Using score and different features, the SVM and NB were applied and different accuracy measurements were calculated. The machine learning techniques gave the best results to classify product reviews. NB got an accuracy of 98.17% and SVM got an accuracy of 93.54% for Camera Reviews.

Sumbal Riaz *et al.* [18] suggested a technique of text mining for analyzing customer reviews to find the customers' opinions and performed the sentiment analysis on the large scale dataset of product (6 categories) reviews given by various customers on the internet. In this approach, sentiment analysis was applied at the phrase level rather than document-level to calculate every term's sentiment polarity. Then key graph keyword extraction technique was used for extracting keywords from each document with high-frequency terms and the intensity of sentiment polarity by measuring its strength was calculated. The k-means clustering algorithm was used to group the data on the basis of sentiment strength value. These values were compared with the star rating of the same data and the better and neutral sentiment toward products was analyzed.

Satuluri Vanaja and Meena Belwal [19] presented an Aspect-level Sentiment Analysis, which was accomplished by Identification, Classification, and aggregation. The classification was done using two machine learning algorithms, SVM and NB classification algorithm and the performance were compared based on Precision, Recall, and F1 measure. The results showed that it obtained more accuracy from the Naïve Bayes algorithm than the Support Vector Machine algorithm.

Wei Zhang *et al* [20] presented an emotion classification algorithm based on latent semantic analysis and SVM. Firstly, natural language processing and psychology were combined to divide the emotions in the online reviews into four categories: happiness, hope, disgust, and anxiety. Then, the latent semantic analysis method was used to optimize the text feature extraction process and utilized the SVM as a classifier to improve the emotional classification accuracy and computational efficiency. The experimental results showed that the model can effectively calculate online reviews. Context meanings of information with deep learning algorithms were used to combine the reviews theme, product characteristics and sentiment classification method to further enhance the multiple class emotional detection accuracy.

Barkha Bansal and Sangeet Srivastava [21] presented a Hybrid Attribute-Based Sentiment Classification (HABSC) to capture the implicit word relations and incorporating domain-specific knowledge. This approach found the most frequent bigrams and trigrams in the corpus, followed by POS tagging to retain aspect descriptions and opinion words. Then, it was employed TFIDF (term frequency-inverse document frequency) to represent each document, followed by automatically extracting an optimal number of topics. All the adjectives and adverbs were labeled using domain-specific knowledge and pre-existing lexicon. The efficiency of that method was tested using datasets. Results showed that the classification accuracy of HABSC significantly exceeded for various state-of-the-art methods and also exhibited less computational time as compared to distributed vectorization schemes.

Chonghui Guo *et al* [22] investigated a ranking method through online reviews based on different aspects of the alternative products, which combined both objective and subjective sentiment values. Firstly, the objective sentiment value of the product was calculated by determining the weights of these aspects with the LDA topic model. During this process, the realistic meaning of each aspect was also summarized. Then, consumers' personalized preferences were taken into consideration while calculating the total scores of alternative products. Meanwhile, comparative superiority between every two products also contributes to their final scores. By using the Page Rank algorithm, the final score of each product was computed from the constructed graph model. The result demonstrated that while considering only objective sentiment values of the product, the ranking result obtained by this method had a strong correlation with the actual sales orders.

3. PROPOSED METHODOLOGY

Sentiment analysis of online reviews is an important way of mining useful information from the Internet. It is a machine learning approach in which machines classify and analyze the human's sentiments, emotions, opinions, etc. about the products which are expressed in the form of customer reviews and star rating. The review analysis of online products not only improves product quality but also influence the purchase decisions of the consumers. Thus, product review analysis is a widely accepted platform where the consumer can easily be aware of their requirements. Several machine learning techniques were proposed for the sentiment analysis of online products. But in existing techniques, review analysis was done by using some limited features only and they didn't concentrate on the future prediction of online products using user review comments. To overcome such drawbacks in this paper, two methodologies are proposed i.e. Deep Learning Modified Neural Network (DLMNN) and Improved Adaptive Neuro-Fuzzy Inference System (IANFIS) for review analysis and future prediction of online product. For the proposed sentiment analysis, the Food review dataset is used. From the dataset, the data values are partitioned into three scenarios and then classified into positive, negative, and neutral using DLMNN. They are Grade based (GB), Content-based (CB), and Collaboration based (CLB). Firstly, the efficient review analysis stage is performed for these three scenarios. For Grade based technique, the polarity score is calculated depending upon the score value of user ratings and then classified as positive, negative, and neutral using DLMNN. Then for CB and CLB, the following steps are carried out: Preprocessing, Feature extraction, Feature Selection, and Review Classification. Here for each CB and CLB scenario, the above-mentioned steps are performed

separately. In CB, the comments given by the user for the particular product are used for review analysis whereas, in the CLB scenario, both user ratings and comments are used for review analysis. After the process of review classification based on these three scenarios, the future prediction of the products is done. For this, the weighting factor of the pre-processed data is estimated by performing five operations such as keyword frequency, identification positive and negative word, computation of support, computation of confidence, and entropy estimation, and then the future prediction of the product is done using IANFIS. The architecture of the proposed method is shown in figure 1.

Figure 1: Architecture of the proposed method

3.1 Food Review Dataset

The dataset used for the proposed method is the food review dataset. This is the publicly available dataset that has 5 lakhs of records and the dataset contains features like Product Id, User Id, Profile Name, Helpfulness Numerator, Helpfulness Denominator, and Score Time Summary Text. From this dataset, 80% of data is used for training and 20% of data is used for

testing. The review analysis of the online product is done by considering three scenarios of the dataset. They are GB, CB, and CLB. The GB encompasses the information about ratings of the online products. The rating varies for each product in the dataset. The CB encompasses the information about the comments of the products that were written by the customer. Then the CLB encompasses both rating and customer written reviews of the product. From these three, the review of the product has been analyzed. In the proposed method, the review analysis is done separately for these three scenarios and inputted to the DLMNN for review classification. The step by step explanation of each scenario is given below.

3.2 GB Scenario

For the GB scenario, the polarity score is calculated for each product and then given to DLMNN for classification. The polarity score for the product is computed based on the ratings (stars) clicked by the customer for the particular product. If the star for the product is 4 and 5, the polarity score for the product is 1, if the star clicked by the customer for the online product is 3, then the polarity score is 0.5, and if the star for the product is 2 and 1 means then the polarity score of the product is 0. Based on these polarity scores, the sentiment of the product is classified. These computed polarity scores of the product are given as input to the DLMNN classifier. The DLMNN classifies the polarity score values into positive, negative, and neutral. That means the result of positive indicates that the product is a good one and has lots of positive stars from the customer, negative indicates that the product is a bad product which receives lots of negative stars and neutral indicates the product is a neutral one which has both positive and negative stars.

3.3 CB Scenario

For the review analysis of CB, the data values in the dataset undergo the following processes.

- Pre-processing
- Feature extraction
- Feature selection
- Classification

3.3.1 Preprocessing

In this stage, the comments of the online products stored in the dataset are pre-processed by performing tokenization, stop word removal, stemming, and hashtag and username removal.

- **Tokenization:** Tokenization is the process of dividing the text into a set of meaningful pieces. These pieces are called tokens. For example, a chunk of text is divided into words or sentences.
- **Stop word removal:** Removal of commonly used words that are unlikely to be useful for learning. A majority of the words in a given text are the connecting parts of a sentence

rather than subjects, objects or intent. A word like "the" or "and" can be removed by comparing text to a list of stop words.

- **Stemming:** Much of natural language machine learning approaches are about the sentiment of the text. Stemming is a process where words are reduced to root by removing inflection through dropping unnecessary characters, usually a suffix.
- **URL, Hashtag and Username removal:** Hashtags are particularly used in social networks to mark keywords in messages, which make it easier 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.

3.3.2 Feature Extraction

In this phase, the features like 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.

3.3.3 Feature Selection

After feature extraction, the necessary features are selected by using the Spider Monkey Optimization Algorithm (SMOA). In this step, the algorithm selects the necessary features depending upon the fitness values of extracted features. The brief explanation about the SMOA is given as follows

3.3.3.1 SMOA

SMOA is basically a population-centered stochastic algorithm. SMOA is enthused by the smart food foraging behavior. The foraging behavior of spider monkeys shows that these monkeys fall in the category of fission-fusion social structure (FFSS) based animals. Thus, the optimization algorithm, which is based on the foraging behavior of spider monkeys can be explained better in terms of FFSS. The following are the key features of the FFSS.

1. The monkeys forage for food in a group of 40 to 50 members, which will be led by the group leader (female monkey). A group leader generally leads the group and is responsible for searching the food sources.
2. If the group leader is not able to find sufficient food for all the group members, then she divides the group into subgroups consisting of 3 to 8 monkeys to reduce the competition. These groups now search for food independently.
3. Sub-groups are also supposed to be led by a female (local leader) who becomes a decision-maker for planning an efficient foraging route each day.
4. Subgroups communicate with each other to exchange information about the availability of food and territorial boundaries.

The 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 stages, the population of the Spider Monkeys (SM) is initialized. The step by step by explanation is provided below

- **Initialization:** The solution of each monkey is a D-dimensional vector where D is the number of parameters (variables) to be optimized. The SMO initializes the spider monkeys with the uniformly distributed values between the maximum and minimum values.

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

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 spider monkeys (SM) update their values based upon the experience of the local leader of the subgroup and the monkeys in the same sub-group. The fitness of the new solution is calculated. If the fitness of the new solution is more than the original solution, then the monkey updates its solution. The position update of i th SM, which 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 which is randomly selected from m th group in order that $r \neq i$ in j th dimension.

- **Global Leader Phase (GLP)**

After LLP, the SM update their positions based on the experience of a GL and local group member. This is performed for better convergence. It is evaluated using the below equation

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

Where gl_j denotes the GL location in ' j th' dimension and $j \in \{1, 2, \dots, D\}$ specifies arbitrarily selected index. The probability of spider monkeys is selected for updating their position, which is proportional to its fitness value so that the best monkey has more chance of being selected and improving itself. The probability is computed using its fitness. The Probability is computed as

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

➤ **Global Leader Learning (GLL)**

The global leader position is updated by searching all the solutions. The position of spider monkey with the highest fitness function is taken as the position of the global leader. Also, it is checked that the position of the global leader has changed or not. If the position didn't change, then the value of the Global Limit Count is incremented by one.

➤ **Local Leader Learning (LLL)**

Here, the greedy search is done within the subgroups. The spider monkey with the highest fitness function is now chosen as the local leader. The old and updated position of the local leader is also checked. If it has not changed, then the Local Limit Count is incremented by one.

- **Local Leader Decision (LLD)** This phase helps 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.

3.3.4 DLMNN classifier

Then after feature selection, the selected feature words are ranked using sentiwordnet dictionary. SentiWordNet is an opinion lexicon derived from the WordNet database where each term is associated with numerical scores indicating positive and negative sentiment information. Then the feature word encompasses both positive and negative ranking. These score values of the products are given as an input to DLMNN and are classified into positive, negative, and neutral. The classification results of positive indicates that the product is a good product and has more positive comments from the customers, negative indicates that the product is not a good one and receives lots of negative comments and low stars for the product, and neutral indicates that the product is average product that received both positive and negative comments from the customer.

In DLMNN, each input is provided to a discrete node prevailing in the input region of the classifier. The weights are the arbitrarily assigned values and are linked with each input. The subsequent one is the hidden layer. The nodes of this layer are termed as hidden nodes. These nodes perform the function of adding the product of the input value and the weight vector of all the input nodes that are linked to it. In DLNN, the weight values are optimized using the HDF-GA which is called DLMNN. Random weight value gives a more backpropagation process to achieve the result. So the optimization is performed in the proposed method. The activation operation is then applied and this layer's output is transported to the consecutive layer. The steps in DLMNN classification are explained as follows,

Step 1: Initialize the ranked data values and their equivalent weights using equ (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 indicates the n number of ranked data's as $R_1, R_2, R_3, \dots, R_n$ and w_i denotes the weight value of R_i which includes the 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 input ranked data with the randomly selected weight vectors and then sum up that values totally, which is expressed as below.

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

Where S_m denotes the summed value.

Step 3: Determine the 'activation function' (AF) using the below equation (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 Af_i specifies the activation function and B_i specifies the exponential of R_i .

Step 4: Compute the next hidden layer's output using the below equation (11)

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

Where, b_i denotes the bias value, w_i specifies the weight betwixt the input and the hidden layers.

Step 5: Perform the above 3 steps for each layer of DLMNN. Finally, evaluate the output by summing all the input signals' weights to attain the output layer neurons' value which is expressed as

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

Where, O_i denotes the value of the layer that precedes the output one, w_j specifies the weights of the hidden layer, in addition, U_i indicates the output unit.

Step 6: Compare the network output with the target value. The difference betwixt these two values is called the error signal. This value is mathematically signified using (13) as follows,

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

Where, e_s implies the error signal, A_i specifies the aimed target output.

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

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

Step 8: Evaluate the weight correction utilizing the backpropagation methodology using equation ()

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

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 using an HDF-GA algorithm. The explanation for HDF-GA is given in the below section.

3.3.4.1 HDF-GA

The main inspiration of the DA algorithm originates from the static and dynamic swarming behaviors of dragonflies in nature. Two essential phases of optimization, i.e. exploration and exploitation, are designed by modeling the social interaction of dragonflies in navigating, searching for foods, and avoiding enemies when swarming dynamically or statistically. Here, three main rules are followed by one another (nature of swarms)

- Separation (S_p) - The dragonflies avoid one another due to collision in a stationary position from the neighborhood.
- Alignment (A_l) - Each dragonfly's velocity coordinates with another one in neighborhood
- Cohesion (C_h) - The dragonflies fly towards the midpoint of the group of the neighborhood.

Here in the proposed method, the GA's crossover and mutation operations are hybridized with the convention dragonfly algorithm to get effective optimization results. So the proposed technique is named HDF-GA. The steps embraced in the HDF-GA are explained below

Step 1: Initialize the size of n dragonflies as N and the i th dragonfly position is denoted as follows

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

Where $i = 1, 2, 3, \dots, N$, m_i^d denotes the position of the i th dragonfly in d th dimension of the search space and N denotes the number of search agents.

Step 2: Calculate S_p using the below equation

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

Where $S_p(i, t)$ denotes the separation motion for an i th individual at the t th iteration, $M(i, t)$ denotes the position of the current individual i at the t th iteration, and $M(j, t)$ specifies the position of the neighboring individual j at the t th iteration.

Step 3: Evaluate the velocity matching of individuals among the neighbourhood using the below equation

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

Where $A_i(i, t)$ is the alignment motion of the i at the iteration t , $v(j, t)$ denotes the velocity of j at the t th iteration.

Step 4: Compute the cohesion of individuals towards the centre of the mass of neighbourhood using the below equation

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

Step 5: Compute the motion of attraction towards the food source using equation (20)

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

Where $A_r(i, t)$ is the food attraction motion, $M(f, t)$ is the food source position at the iteration t . Individual dragonflies with the best objective function up to the current iteration will be considered as food.

Step 6: Compute the motion of distraction against the enemy using (21)

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

Where $D_r(i, t)$ is the enemy distraction motion, $M(e, t)$ is the enemy distraction motion of i at iteration t . Individual dragonflies with the worst objective function up to the current iteration will be considered as an enemy.

Step 7: Then, 2 genetic operators are added. Crossover and mutation are applied when the dragonfly does not have at least one neighboring dragonfly. This makes optimization more effective. The type of crossover that is used is the two-point crossover. This is done using 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 are the two points that are chosen as crossover points.

Step 8: Mutation is carried out by replacing the number of genes from each chromosome with new genes. The replaced genes are randomly generated genes without any repetition within the chromosome. In the proposed system, the chromosomes are the set of parameters that define the solution.

Step 9: Update the position of the dragonflies in a search space and simulate their movements using two vectors such as step and position vector. The step vector specifies the direction of the movement of the dragonflies and is represented as,

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

Where, ΔX signifies the Step Vector, t is the current iteration, s is the separation weight, S_p denotes the separation of the i -th individual, a indicates the alignment weight, A_l refers 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 denotes the iteration counter.

Step 4: The position vector can be calculated from the value of the step vector. It is calculated as follows.

$$Y_{t+1} = Y_t + \Delta Y_{t+1} \quad (25)$$

Where, ΔY is the Step Vector, t is the current iteration and Y is the Position Vector. The pseudo-code for the proposed HDF-GA is depicted in below figure 2.

Input: Weight values of DLMNN Output: Optimized Weight values
Begin Initialize the population of dragonflies as $M_i = (m_i^l, m_i^d, \dots, m_i^N)$ Initialize the step vectors While the stopping criteria is not met Calculate the objective function for all M_i Calculate S_p, A_i, C_h, A_r and D_r Update the neighborhood radius If the current dragonfly has at least one neighbouring dragonfly Select two crossover points Perform mutation Else Update the step vector using equ (24) Update the position vector using equ (25) End if Check and correct the new positions on new boundaries of variables End while End

Figure 2: Pseudocode for the proposed HDF-GA

3.4 CLB Scenario

In the CLB scenario, the review analysis of online products is done by collaborating GB with CB scenarios. Here, both grades and reviews given by the customer for a particular product have been considered. Comparing both results of GB and CB, the CLB scenario is analyzed.

3.4 Future Prediction

After the process of review analysis, the future prediction of the online products is done with the help of the IANFIS technique. Firstly, the weighting factor of the online product has been computed by performing five steps. They are

Keyword Frequency: The frequency of keyword is the number of occurrences of a particular keyword on the dataset. The keyword frequency is expressed as,

$$K_{f_i} = \{K_{f1}, K_{f2}, K_{f3}, \dots, K_{fn}\} \quad (26)$$

Identification: After finding the frequent keywords in the dataset, the positive and negative keywords from the frequent keywords are identified using sentiwordnet dictionary. After that, support, confidence and entropy values are computed for these positive and negative words.

Support: The support is the percentage of transactions in the database that contains every keyword in the database. Here two keywords (K_1, K_2) are considered. The support of an association rule $S \rightarrow T$ is given by,

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

Confidence: The confidence is the percentage of transactions in the database with the keywords S and T . The confidence is calculated using the conditional probability which is further expressed in terms of keyword support. The equation for the confidence is given by,

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

Where, $P(K_1 \cup K_2)$ denotes the number of transactions containing the keyword K_1 and K_2 , and $P(K_1)$ is the number of transactions containing the keyword K_1 .

Entropy: Entropy is the average of the information attribute. The information quantity of each attribute generates an arbitrary variable which may be the anticipated or average value and is regarded as entropy. Entropy computes as follows,

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

Where, S_p denotes support and C_f denotes the confidence. These computed entropy values are then given as the input to the IANFIS classifier for future prediction.

3.4.1 IANFIS

In this step, the entropy values of positive and negative frequent keywords are given as an input to the IANFIS algorithm for classifying the future prediction of the product into high and low. The classification result of high indicates that the future demand for the product is high and low indicates that the future demand for the product is low. These classification is done based on the given entropy values of positive and negative frequent keywords. If the product gets the highest entropy value for the positive keyword, then the future demand for the product is also high and if the product gets the highest entropy value for the negative keyword, then the future demand for the product is low, which can be done using IANFIS. Based on this future demand of the particular product, all high-quality reviews of the particular product is visible on the top to customers and also that product is suggested on the top list of the product category. The brief explanation of IANFIS is given as follows

An adaptive neuro-fuzzy inference system (ANFIS) is a kind of artificial neural network that is based on Takagi–Sugeno fuzzy inference system. Since it integrates both neural networks and fuzzy logic principles, it has the potential to capture the benefits of both in a single framework. Its inference system corresponds to a set of fuzzy IF-THEN rules that have the learning

capability of approximating nonlinear functions. The architecture of ANFIS is composed by five layers. The role of each layer is explained as follows

- ✓ The first layer is called as fuzzification layer. It takes the input values and determines the membership functions belonging to them.
- ✓ The second layer is specified as the rule layer, which is responsible for generating the firing strengths for the rules.
- ✓ Then, the third layer normalizes the computed firing strengths, by dividing each value for the total firing strength.
- ✓ Afterward, the normalized values are given to the fourth layer (consequence parameter set).
- ✓ The values returned by the fourth layer are the defuzzificated ones and those values are passed to the fifth layer for returning the final output.

The conventional ANFIS model uses the bell membership function (MF). But the proposed method uses Gaussian kernel MF (GMF) to improve the performance of the rule generation process. So the ANFIS is called improved ANFIS (IANFIS). The two basic rules of IANFIS are specified in the below equations.

Rule 1: If E_1 is P_i and E_2 is Q_i then,

$$Rules_i = sE_i + t_iE_{i+1} + u_i \quad (30)$$

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

$$Rules_{i+1} = s_{i+1}E_i + t_{i+1}E_{i+1} + u_{i+1} \quad (31)$$

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 different entropy values obtained from previous step. $s_i, t_i, u_i, s_{i+1}, t_{i+1}$ & u_{i+1} values are the parameter set. The layers in IANFIS are explained as follows

Layer 1: Every node in this layer is an adaptive node with a node function

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

Here, E_i is the input to node i . Each node adapts to a function parameter. The output from each node is a degree of member-ship value that is given by the input of the member-ship function (MF). The MF that is used in the proposed work is Gaussian kernel MF. This is specified in the succeeding equation.

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

Where, s_i , t_i and also u_i are the MFs parameters that can change the shape of the MF. The parameters are alluded to as the premise parameters.

Layer 2: Every node in this layer is a fixed node whose output is the product of all the incoming signals.

$$IA_{2,i} = K_i = \mu_{f_1}(E_i) \times \mu_{f_2}(E_{i+1}) \quad (34)$$

The output of this layer $IA_{2,i}$ represents the firing strength of a rule.

Layer 3: Every node in this layer is a fixed node labeled N. The i th node calculates the ratio of the i th rule's firing strength to the sum of all rules' firing strengths.

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

For convenience, outputs of this layer are called normalized firing strengths.

Layer 4: Every node in this layer is an adaptive node with a node function.

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

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

Layer 5: The single node in this layer is a fixed node, which computes the overall output as the summation of all incoming signals. In this layer, the circle node is labeled as $\sum$

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

4. RESULT AND DISCUSSION

The proposed sentiment analysis of online product reviews using DLMNN and future predictions using IANFIS are implemented in the working platform of JAVA. Here, the performance analysis of three scenarios such as GB, CB, and CLB, which are used for review analysis of online product reviews are compared and the IANFIS technique used for future prediction of online product is compared with the existing ANFIS. For both review analysis and future prediction, the machine learning approaches are used. So the performance measures such as precision (p_s), recall (r_k), f-score (f_s), and accuracy (a_c) are used to compare the proposed schemes' performance.

4.1 Performance analysis of GB, CB, and CLB using DLMNN

In this phase, the performance analysis of GB, CB, and CLB scenarios using DLMNN is compared in terms of the performance measures like p_s , r_k , f_s , and a_c which is shown in table 1.

Table 1: Performance comparison of GB, CB, and CLB
(a)

No of Data	Performance measures					
	p_s			r_k		
	GB	CB	CLB	GB	CB	CLB
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

(b)

No of Data	Performance measures					
	f_s			a_c		
	GB	CB	CLB	GB	CB	CLB
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

Table 1 shows the performance comparison for the GB, CB, CLB scenarios of review analysis using DLMNN regarding (a) p_s , and r_k , and (b) f_s , and a_c . For comparing three scenarios, 5000 numbers of data are considered. For 1000 numbers of data, the GB and CB gives the values as 86.3266 and 87.9988 for p_s , 92.4555 and 92.1125 for r_k , 90.3324 and 91.0455 for f_s , and 96.8543 and 89.7764 for a_c 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 which are higher than GB and CB scenarios. Likewise, for 5000 numbers of 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 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 86.67845 and 96.6654 for a_c which are lower than CLB scenario. Compared to GB, the CB scenario achieves good results for all performance measures. But the CLB scenario achieves better results for all performance measures compared to GB and CB scenario. The graphical illustration of the above table 1 is plotted in figure 3.

Figure 3: Performance graph for the GB, CB, and CLB scenario using DLMNN regarding (a) precision and recall, and (b) f-score and accuracy

4.2 Performance analysis of IANFIS

Here, the performance analysis of IANFIS used for future prediction of online product is compared with the existing ANFIS regarding p_s , r_k , f_s , and a_c which is shown in figure 4.

Figure 4: Performance graph for the IANFIS with existing ANFIS

The above figure 4 shows the performance comparison graph for the proposed IANFIS with the existing ANFIS concerning the performance measures say (a) p_s , and r_k and (b) f_s , and a_c . Totally 5000 numbers of data are taken for performance comparison of the proposed data. The graph is plotted for 1000-5000 numbers of data. For 1000 numbers of data, the existing ANFIS gives the values of 84.4523, 88.1252, 89.2344, and 90.1284 for p_s , r_k , f_s , and a_c whereas 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 which are higher than existing ANFIS. Likewise, for large numbers of data (5000), the proposed IANFIS gives 86.4555, 90.8322, 91.8544, and 91.6654 for p_s , r_k , f_s , and a_c but the ANFIS gives 85.4555, 90.8876, 89.8875, and 88.6787 for p_s , r_k , f_s , and a_c which are lower than IANFIS. And also for the remaining data (2000, 3000, and 4000), the IANFIS obtains the best results. So, from the comparison, it is concluded that the proposed IANFIS achieves superior performance for the future prediction of online products when compared with existing ANFIS.

5. CONCLUSION

In this paper, a methodology for sentiment analysis of online product review is proposed using DLMNN and a methodology for future prediction of online product is proposed using IANFIS. The performance analysis of these two proposed methodologies is done. The proposed DLMNN is applied for three scenarios (GB, CB, and CLB) of review analysis and the comparison is done for these three scenarios concerning the performance measures of p_s , r_k , f_s , and a_c , these measures are compared for a number of data (from 1000 to 5000). In comparing the three scenarios, the CLB scenario achieves the best results for product review analysis. And the IANFIS for future prediction is compared with the existing ANFIS. The proposed IANFIS achieves the highest values for p_s , r_k , f_s , and a_c when compared with existing ANFIS. So, from the performance analysis, the paper concludes that the proposed CLB scenario and IANFIS work well for sentiment analysis and future prediction of online products.

List of abbreviations

DLMNN : Deep Learning Modified Neural Network

ANFIS : Adaptive Neuro-Fuzzy Inference System

IANFIS: Improved Adaptive Neuro-Fuzzy Inference System

SA : Sentiment Analysis

GB: Grade Based

CB : Content Based

CLB : Collaborative Based

SA : Sentiment Analysis

SMOA : Spider Monkey Optimization Algorithm

Declarations

Availability of data and materials

This dataset consists of reviews of fine foods from amazon. The data span a period of more than 10 years, including all ~500,000 reviews. Reviews include product and user information, ratings, and a plain text review. It also includes reviews from all other Amazon categories.

J. McAuley and J. Leskovec. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. WWW, 2013.

Competing interests

The authors declare that they have no competing interests.

Funding

The research received no external funding.

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.

Author information

Affiliations

Sasikala P^{#1} , L.Mary Immaculate Sheela^{*2}

^{#1} Research Scholar, Department of Computer science, Mother Teresa Women's University, Kodaikanal, India. shakthesasi@gmail.com

^{*2} Professor & Dean – Fesac, Pentecost University College, Accra, Ghana.
drsheela09@gmail.com

Corresponding author

Correspondence to Sasikala P

Acknowledgements

Not applicable.

REFERENCES

1. Minu Choudhary, and Prashant Kumar Choudhary, "Sentiment analysis of text reviewing algorithm using data mining", In International Conference on Smart Systems and Inventive Technology (ICSSIT), pp. 532-538, 2018.
2. Vaidehi Gondaliya, and Krupa Mandaviya, "An improved approach for online trending forum detection based on sentiment analysis", In International Conference on Inventive Computation Technologies (ICICT), vol. 3, pp. 1-5, 2016.
3. KL Santhosh Kumar, Jayanti Desai, and Jharna Majumdar, "Opinion mining and sentiment analysis on online customer review", In IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), pp. 1-4, 2016.
4. 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.
5. Zeenia Singla, Sukhchandan Randhawa, and Sushma Jain, "Statistical and sentiment analysis of consumer product reviews", In 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1-6, 2017.
6. Fatemeh Hemmatian, and Mohammad Karim Sohrabi, "A survey on classification techniques for opinion mining and sentiment analysis", Artificial Intelligence Review, pp. 1-51, 2017.
7. Trupthi, M., Suresh Pabboju, and Narasimha G, "Sentiment analysis on twitter using streaming API", In IEEE 7th International Advance Computing Conference (IACC), pp. 915-919, 2017.

8. Lijuan Zheng, Hongwei Wang, and Song Gao, "Sentimental feature selection for sentiment analysis of Chinese online reviews", *International journal of machine learning and cybernetics*, vol. 9, no. 1, pp. 75-84, 2018.
9. Rajalaxmi Hegde, and 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 (AEEICB)*, pp. 122-125, 2017.
10. Rushdi Saleh M., Maria Teresa Martín-Valdivia, Arturo Montejo-Ráez, and Ureña-López L. A., "Experiments with SVM to classify opinions in different domains", *Expert Systems with Applications*, vol. 38, no. 12, pp. 14799-14804, 2011.
11. Hari Krishna M, Rahamathulla K, and Ali Akbar, "A feature based approach for sentiment analysis using SVM and coreference resolution", In *International Conference on Inventive Communication and Computational Technologies (ICICCT)*, IEEE, pp. 397-399, 2017.
12. Shweta Rana, and Archana Singh, "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.
13. Krishna Vamshi, B., Ajeet Kumar Pandey, and Kumar AP Siva, "Topic model based opinion mining and sentiment analysis", In *International Conference on Computer Communication and Informatics (ICCCI)*, IEEE, pp. 1-4, 2018.
14. Xuedan Du, Yinghao Cai, Shuo Wang, and Leijie Zhang, "Overview of deep learning", In *31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, IEEE, pp. 159-164, 2016.
15. Asad Abdi, Siti Mariyam Shamsuddin, Shafaatunnur Hasan, and Jalil Piran, "Deep learning-based sentiment classification of evaluative text based on Multi-feature fusion", *Information Processing & Management*, vol. 56, no. 4, pp. 1245-1259, 2019.
16. Feilong Tang, Luoyi Fu, Bin Yao, and Wenchao Xu, "Aspect based fine-grained sentiment analysis for online reviews", *Information Sciences*, vol. 488, pp. 190-204, 2019.
17. Rajkumar Jagdale, S., Vishal S. Shirsat, and Sachin N. Deshmukh, "Sentiment analysis on product reviews using machine learning techniques", In *Cognitive Informatics and Soft Computing*, Springer, Singapore, pp. 639-647, 2019.
18. Sumbal Riaz, Mehvish Fatima, M. Kamran, and M. Wasif Nisar, "Opinion mining on large scale data using sentiment analysis and k-means clustering", *Cluster Computing*, vol. 22, no. 3, pp. 7149-7164, 2019.

19. Satuluri Vanaja, and Meena Belwal, "Aspect-level sentiment analysis on e-commerce data", in international conference on inventive research in computing applications (ICIRCA), IEEE, pp. 1275-1279, 2018.
20. Wei Zhang, Sui-xi Kong, and Yan-chun Zhu, "Sentiment classification and computing for online reviews by a hybrid SVM and LSA based approach", Cluster Computing, pp. 1-14, 2018.
21. Barkha Bansal, and Sangeet Srivastava, "Hybrid attribute based sentiment classification of online reviews for consumer intelligence", Applied Intelligence, vol. 49, no. 1, pp. 137-149, 2019.
22. Chonghui Guo, Zhonglian Du, and Xinyue Kou, "Products ranking through aspect-based sentiment analysis of online heterogeneous reviews", Journal of Systems Science and Systems Engineering, vol. 27, no. 5, pp. 542-558, 2018.

Figures

Figure 1

Architecture of the proposed method

Input: Weight values of DLMNN

Output: Optimized Weight values

Begin

Initialize the population of dragonflies as $M_i = (m_i^l, m_i^d, \dots, m_i^N)$

Initialize the step vectors

While the stopping criteria is not met

Calculate the objective function for all M_i

Calculate S_p, A_i, C_h, A_r and D_r

Update the neighborhood radius

If the current dragonfly has at least one neighbouring dragonfly

Select two crossover points

Perform mutation

Else

Update the step vector using equ (24)

Update the position vector using equ (25)

End if

Check and correct the new positions on new boundaries of variables

End while

End

Figure 2

Pseudocode for the proposed HDF-GA

(a)

(b)

Figure 3

Performance graph for the GB, CB, and CLB scenario using DLMNN regarding (a) precision and recall, and (b) f-score and accuracy

(a)

(b)

Figure 4

Performance graph for the IANFIS with existing ANFIS