Adaptive Rider Feedback Artificial Tree Optimization-Based Deep Neuro-Fuzzy Network for Classification of Sentiment Grade

Sireesha Jasti and G.V.S. Raj Kumar

Department of Computer Science and Engineering, GITAM School of Technology, GITAM (Deemed to be University), Visakhapatnam, Andhra Pradesh, India

https://doi.org/10.26636/jtit.2023.165222

Abstract — Sentiment analysis is an efficient technique for expressing users' opinions (neutral, negative or positive) regarding specific services or products. One of the important benefits of analyzing sentiment is in appraising the comments that users provide or service providers or services. In this work, a solution known as adaptive rider feedback artificial tree optimization-based deep neuro-fuzzy network (RFATO-based DNFN) is implemented for efficient sentiment grade classification. Here, the input is pre-processed by employing the process of stemming and stop word removal. Then, important factors, e.g. SentiWordNet-based features, such as the mean value, variance, as well as kurtosis, spam word-based features, term frequency-inverse document frequency (TF-IDF) features and emoticon-based features, are extracted. In addition, angular similarity and the decision tree model are employed for grouping the reviewed data into specific sets. Next, the deep neuro-fuzzy network (DNFN) classifier is used to classify the sentiment grade. The proposed adaptive rider feedback artificial tree optimization (A-RFATO) approach is utilized for the training of DNFN. The A-RFATO technique is a combination of the feedback artificial tree (FAT) approach and the rider optimization algorithm (ROA) with an adaptive concept. The effectiveness of the proposed A-RFATO-based DNFN model is evaluated based on such metrics as sensitivity, accuracy, specificity, and precision. The sentiment grade classification method developed achieves better sensitivity, accuracy, specificity, and precision rates when compared with existing approaches based on Large Movie Review Dataset, Datafiniti Product Database, and Amazon reviews.

Keywords — deep learning network, feedback artificial tree, natural language processing (NLP), rider optimization algorithm, sentiment grade classification.

1. Introduction

Sentiment analysis [1], [2] is a growing field of research focusing on analyzing product and service reviews published online. Sentiment analysis is considered to be a "baggage issue", which deals with polarity identification [2], [3] aspect extraction [4], [5], and natural language processing (NLP) tasks [2], [6]. Aspect-level sentiment classification is the basic approach in sentiment analysis, and it serves to deduce sentiment polarity. For instance, whether a given sentence is classified as neutral, positive or negative in sentiment depends on finding a specific element therein [2]. Sentiment analysis is effective for evaluating users' opinions with regards to services or products described by text data [7], [8]. Therefore, its primary task is to identify and categorize the user's polarity based on content [9]–[11]. Sentiment analysis is also the main research area in NLP, relied upon to analyze the opinions and emotions of users from the texts they provide [12].

Sentiment analysis is usually harnessed by e-commerce websites and online social networks [13] for predicting their users' needs based on polarity or sentiment categories [4]. The analyses are performed with the intention to identify specific options and recognize their polarity. Sentiment is categorized into three sets (neutral, negative, and positive) with five different groups existing within each of those sets (strong opposition, opposition, support, strong support, and neutrality) [14].

Subjective information from the text concerned is recognized automatically using emotions, opinions, attitudes, etc. [15] and plays a vital part in NLP and data mining [16]. Three levels (target, sentence, and document) are taken into consideration when analyzing sentiment. The sentence and document levels are intended to categorize sentences and documents, and they consider the entire document or sentence as the basic information-carrying component. Polarity sentiment is also analyzed based on textual content retrieved from a sentence or document [17]. The features extracted from texts are subjected to classification in the corpus-based approach. In addition, the intensification and negation process is conducted by combining a lexicon-based technique for identifying scores from input text based on keywords and phrases dictionary [18]. Strength and polarity are specified, in sentiment dictionaries, using corpus words and phrases [12]. The entire process is relied upon for boosting sales and creating product rankings [19], [20].

Recently, neural networks have been used in several areas, such as discrete-time signals, computer vision, and NLP for sentiment classification [2], as a widely utilized approach for an attention model for aspect-level sentiment classification to recognize the sentiment polarity of targets in context [21], [22]. The deep learning technique [23], [24] performs better than the machine learning approach when it comes to sentiment classification, due to the fact that GPU accelerators are capable of relying on large databases [8], [25]. Adaptive RNN is used in the target-based classification approach, where the sentiment of the wording is transferred to the target based on syntactic and context-related patterns [14], [26]. Tree structures of Long Short-Term Memory (LSTM) networks are also used in the sentiment classification process [16], [27], as are the Gated Recurrent Unit (GRU), LSTM, and gated RNN with LSTM [4].

The major goal of this work is to design and develop an A-RFATO-based DNFN for sentiment classification. The paper provides a meaningful contribution to two main topics:

– **Adaptive RFATO-based DNFN** for efficient sentiment grade classification with a DNFN trained by A-RFATO,
– **Adaptive RFATO** developed by integrating FAT and ROA approaches with the adaptive concept.

The rest of the paper is organized as follows. Section 2 reviews the literature and Section 3 elaborates on the proposed A-RFATO-based DNFN model. Section 4 presents the results and the discussion, while the conclusion is given in Section 5.

2. Literature Review

The CNN approach was developed by Kim and Jeong in [28] to classify sentiment. The input image was taken from a database and then its significant features were extracted for further processing. The flattening process was executed for converting a two-dimensional feature map to a one-dimensional structure. This approach easily confines higher level and local patterns for better precision. However, this method has not identified any improved patterns for sentiment classification. Fu *et al*. [29] introduced a semi-supervised aspect level sentiment classification approach using a variational auto encoder (AL-SSVAE) and employed in during the classification process. The topical word embedding (TWE) method was utilized for identifying aspect-specific word embedding, while joint sentiment topic (JST) was employed for obtaining sentiment from given words. The proposed method efficiently extracts sentiment features and provides global semantics for better performance. Nevertheless, this method did not include a particular neural network for identifying aspect-level sentiments of various texts.

A multi-layer attention-based CNN was modeled by Zhang *et al*. [14]. In this paper, every conventional layer output is considered to form a context representation for capturing additional context features. This method extracts high-level features, like fractional context, and low-level features, such as semantic relationships, phrases, and words, using these in the classification process. This technique utilizes sentiment-related vocabulary for the classification process, but offers poor results in complicated modeling.

Chen *et al*. [2] proposed a fusion model with multi-source data to classify sentiment with unified data structure. The model collects knowledge from different types of resources. After that, aspect-level sentiment is categorized using bidirectional encoder representations from transformers (BERT). The use of BERT means that the classification is performed in a precise manner, in the form of a good aspect-specific sentence representation. Unfortunately, this method fails to cover various resources, such as sentiment knowledge bases and document-level corpora in aspect-level classification.

A graph convolutional network (GCN) was introduced by Zhao *et al*. in [30]. Initially, aspect-level representations were captured by employing a bi-directional attention model using position encoding. Then, a multi-aspect sentiment classification structure was developed to efficiently confine sentiment dependencies among several features. This model effectively classifies the sentiment based on various aspects, but fails to overcome the problem of over-fitting. Salur and Aydin introduced, in [8], a hybrid deep learning technique to classify sentiment. At first, features were extracted using character-level embedding based on CNN and BiLSTM. Next, the extracted features were integrated and broadcast to the SoftMax layer for the classification process. The approach produced high level of classification accuracy. However, attention-based approaches were not deployed for increasing performance. Zhang *et al*. [31] introduced a bi-channel capsule network (SC-BiCapsNet) for sentiment classification. The Cot-Att scheme was used in text semantic representation to optimize text expression and coding patterns. Finally, a capsule network was developed for improving the weight value of significant features in the text classification phase. This method efficiently increases accuracy of the classification process, but failed to incorporate other auxiliary networks in order to decrease computational cost. A multi-task learning technique was devised by Jin *et al*. in [32], with multi-scale CNN and LSTM helping classify sentiment. This approach effectively enhances the quality of the encoder and emotion classification. Time complexity was reduced significantly, but it failed in enhancing sentiment classification outcomes for multitask learning.

3. Proposed A-RFATO-based DNFN

The proposed A-RFATO technique is a combination of the adaptive ROA concept and the FAT approach. The diagram of the classification model is presented in Fig. 1.

A dataset P containing reviews can be expressed by:

$$P = \left\{X_{a,b}\right\} 1 \leqslant a \leqslant A\,,\; 1 \leqslant b \leqslant B\,, \tag{1}$$

where $X_{a,b}$ represents data from the a-th review and with a b-th attribute, A and B indicate all data points and attributes.

The input review data is treated as input in the pre-processing stage and is subjected to stemming and stop word removal processes.

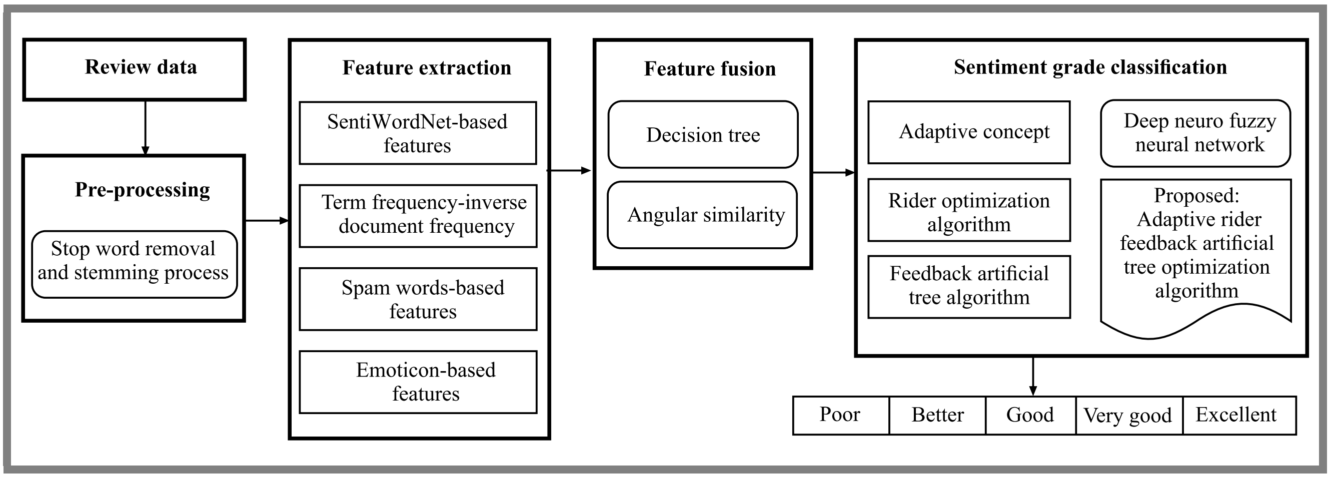

Fig. 1. Diagram of the A-RFATO-based deep neuro-fuzzy network for sentiment grade classification.

3.1. Stop Word Removal

It is an important process in the pre-processing stage, as it allows to avoid processing huge amounts of data in a speedy manner. The current goal is to perform the text examination phase while keeping the negation and intensity of the words in mind in order to effectively complete the sentiment analysis. The Python natural language toolkit (NLTK) package has been used to remove the stop words. Negation words are deleted from the list of stop words and are applied to the input review text. The stop word removal method decreased the amount of information noise by eliminating non-informational behavior from the sentences.

3.2. Stemming

The retrieved words are trimmed to their base form by applying the stemming process, as the processing of large data sets increases complexity of the sentiment classification process. The stemming process allows to avoid complications by eliminating unnecessary words from the input text file. The stemming process generates a compressed document by removing prefixes and suffixes. The output is formulated as:

$$X = \left\{C_d\,,\ 1 \leqslant d \leqslant n\right\}\ , \tag{2}$$

where C_d is the pre-processed data and n is the total number of reviews.

After data pre-processing, the extraction of features is performed. The features taken into consideration are explained below.

3.3. SentiWordNet Features

The establishment of a SentiWordNet structure is a two-step procedure comprising exploration of WordNet and determination of a subset of WordNet. Initially, WordNet term relationships, such as hyponyms, synonyms, and antonyms are explored for enlarging the seed word's core. Next, a subset of WordNet terms, i.e. negative and positive labels, is identified [33]. Positive and negative scores of SentiWordNet features are given by:

$$\left\{\mathrm{M}_x\,,\mathrm{N}_x\right\} = \mathrm{SW}(C_d)\ , \tag{3}$$

where a positive score is given by M_x and a negative score is denoted by N_x, while $\mathrm{SW}(C_d)$ indicates the SentiWordNet features with C_d data. The features identified from the negative and positive score are:

$$m_1^d = \mathrm{M}_x\ ,\ m_2^d = \mathrm{N}_x\ , \tag{4}$$

where m_1^d and m_2^d are features acquired from positive and negative scores of SentiWordNet. These features are applied to train the classifier algorithm. After that, the mean, variance, and kurtosis of positive and negative scores of SentiWordNet are extracted.

The mean offers an estimation of the average amount of data in a document and is expressed by:

$$m_3^d = \frac{1}{e}\sum_{i=1}^{e} \mathrm{E}_i\ , \tag{5}$$

where $i = 1, \ldots, e$, $e = 1$ or 2, which means positive and negative scores, respectively. The total amount of positive and negative data enclosed in a document is specified as E_i while m_3^d indicates the mean value.

Variance is based on the mean value that is specified as m_4^d, and is formulated as follows:

$$m_4^d = \frac{1}{e}\sum_{i=1}^{e}\left(\mathrm{E}_i - m_3^d\right)^2 . \tag{6}$$

Kurtosis is a measure of tailedness of a distribution, i.e. how much density there is in the tails, and is denoted by m_5^d.

3.4. TF-IDF Features

TF-IDF features are extracted using pre-processed data, based on the vector space model (VSM), while the weight of words is estimated by IDF, whereas the occurrence of words is evaluated by DF. The TF feature is expressed by:

$$m_6^d = \frac{1}{e} \sum_{\substack{z=1 \\ y \in z}}^{|e|} TF_d^z , \tag{7}$$

where m_6 represents the TF features acquired from the review document. The word count and frequency of a z-th word in a d-th review is denoted as $|e|$ and TF_d. The IDF feature is calculated by:

$$m_7^d = \sum_{\substack{z=1 \\ y \in z}}^{|e|} IDF_d , \tag{8}$$

where m_7 is the IDF feature and it is extracted from the review document.

3.5. Feature Based on Spam Word

Identification of spam is an important step, as it identifies precise feedback using customers' product reviews. Spam reviews (not valid) also help estimate the best sentiment scores using various reviews. The spam-based feature is estimated by analyzing the content present in the reviews using such review properties as large word count, or detection of a higher number of pronouns. In contrast, reliable reviews involve more coordinating conjunctions, adjectives, nouns, determiners, and prepositions [34]. The features acquired from spam identification are represented by:

$$m_8^d = \frac{\mathrm{N}_x}{|e|} . \tag{9}$$

The spam word and entire word counts are identified as N_x and $|e|$, respectively. For a d-th review:

$$m_9^d = \frac{e_{N_z}}{|e|} , \tag{10}$$

where e_{N_z} is the sum of the frequency of spam words.

3.6. Emoticon-based Feature

Emoticons are visual depictions of face expressions made up of such characters as letters, punctuation marks, and numbers. Each emoticon intensity is measured based on the text preceding that particular emoticon. The difference in the incidence of negative and positive emoticons is estimated as m_{10}^d .

3.7. Feature Fusion by Decision Tree and Angular Similarity

Once the feature extraction process is completed, emoticon-based features, SentiWordNet features, mean, variance, kurtosis, spam word-based features, and TD-IDF features are combined to form feature vectors in order to decrease complexity of sentiment estimation. Feature vectors are formulated as:

$$\mathbf{m} = \left\{ m_1^d,\, m_2^d,\, m_3^d,\, m_4^d,\, m_5^d,\, m_6^d,\, m_7^d,\, m_8^d,\, m_9^d,\, m_{10}^d \right\} , \tag{11}$$

where m_1^d, m_2^d, ... indicate the feature set. The final fused feature vector is expressed as:

$$\mathbf{m}_i^{fused} = \sum_{y-1}^{f} \frac{\mu}{y} m_y , \tag{12}$$

$$y = 1 + \frac{N}{M} , \tag{13}$$

where $i = 1, \ldots, z$, the selected features are indicated as z, N is total feature number and M is the number of features which are to be selected.

3.8. Decision Tree

This subsection describes the procedure of identifying the μ parameter using the decision tree. First, the review content is trained to group the data under division. The μ parameter is identified as the entropy function of the review data if it belongs to a specific group. The diagram illustrating the process of identifying μ via the decision tree and the ground truth is presented in Fig. 2 and is identified by:

$$\mu = AS(n_d, \beta_d) , \tag{14}$$

where n_d is the review data and β_d is the average of the review data belonging to a specific class.

	m_1	m_2	...	m_y	Ground truth
n_1					μ_1
n_2					μ_2

Fig. 2. Determination of the μ parameter.

Then, angular similarity is computed based on the angular distance measure estimated by cosine similarity:

$$AD = \frac{\cos^{-1}(CS)}{\pi} , \tag{15}$$

$$AS = 1 - AD , \tag{16}$$

where AD is the angular distance, CS represents cosine similarity, and AS specifies angular similarity.

The decision tree is further executed for the purpose of the rule induction and classification process. The decision tree is basically a flowchart that has a tree-like shape, in which the test result is specified by an attribute test and a branch. The class label indicated by the leaf node and internal node is used to determine attribute test and branch. The source set is divided into subsets based on the attribute value test in the decision tree and every subset endures recursive way splitting. The recursive portioning ends when the subset node has a target with a similar variable. The decision is characterized by better accuracy and efficiently manages multi dimensional data. The target occurrence is categorized in the decision tree through arranging the tree from the root node followed by attribute testing and shift to the leaf node. This procedure is repeated for the sub trees of new nodes.

3.9. Classification of Sentiment Grade

The input for categorization of the sentiment grade using DNFN has the form of feature fused output. The DNFN is trained using the A-RFATO technique which incorporates ROA, FAT, and adaptive concepts to reduce computational complexity.

3.10. DNFN Structure

The DNFN [35] is a hybrid of fuzzy logic and a deep neural network. In this network, the first processing step is performed based on a deep neural network and then the second processing step is executed by means of fuzzy logic in order to estimate the objective of the system. This system involves three types of layers: input layer, a number of hidden layers for verification, and the output layer. Three rules are applied, including de-fuzzification layer, normalization layer, and rule layer. The output layer is the de-fuzzification layer. Consequents and premises constitute very important parameters of deep neural networks. Premises form the pedestal of the membership function is realized in the modified fuzzification in the input layer, also it deals with the incidence levels. Besides, the consequent parameter depends on the de-fuzzification procedure. A neuro-fuzzy block contains the fuzzy interface system (FIS) used for rule base evaluation. The DNFN structure used for the classification of sentiment is shown in Fig. 3.

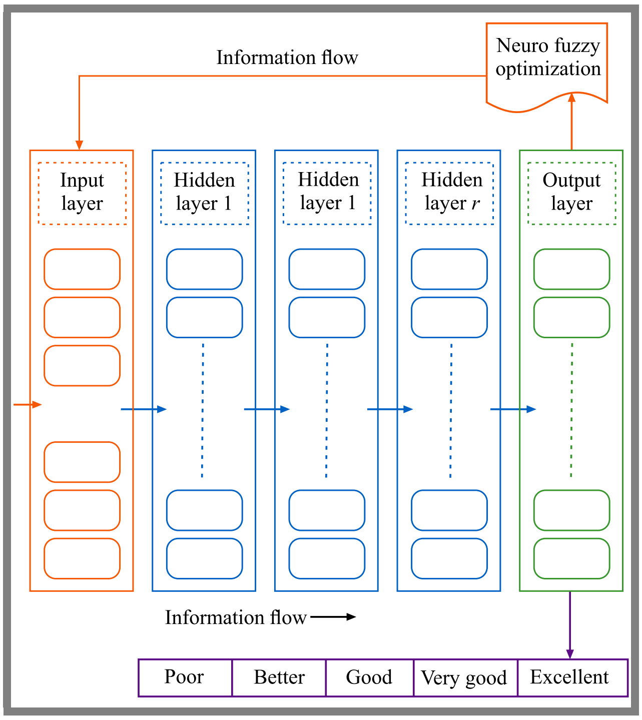

Fig. 3. DNFN structure diagram.

The degree of each input is distributed within the 0–1 range. Output follows every entity of the first layer. Let us consider a case in which the number of premises is two, with one consequent as:

$$J_{1,h} = kC_h(r) \ \text{ or } \ J_{1,h} = kC_{h-2}(t) \,, \ \ \forall h = 1,\, 2,\, 3 \,, \tag{17}$$

where r represents input to each h-th entity, kC and kC_{h-2} specify antecedent membership function and degree of the membership function is indicated by $J_{1,h}$.

The bell-shaped function is used to model the membership function which is distributed with a minimum value of 0 and a maximum value of 1. It can be formulated by:

$$kC_h(r) = \frac{1}{1 + \left|\frac{h = A_h}{B_h}\right|^{2T_h}} \,, \tag{18}$$

where T_h, A_h, and B_h describe the membership function of the premise parameter, enhanced with the help of training.

The rule-based layer refers to layer 2, as it is used to express a group of rules. Each entity in the layer is multiplied by linguistic variables to satisfy the membership degree. The rules' firing strength is symbolized by the multiplication of variable membership values as:

$$J_{2,h} = \rho_h = kC_h(r) kC_{h-2}(t), \ \ \forall h = 1,\, 2 \,. \tag{19}$$

Layer 3 is the normalization level in which each entity calculates the ratio between the firing strength of the h-th rule and the total firing strength of all rules, and ρ_h refers to a generic network parameter. The result for each rule is normalized based on the firing strength of that rule and equals:

$$J_{3,h} = \overline{\rho}_h = \frac{\rho_h}{\rho_1 + \rho_2} \ \ \forall h = 1,\, 2 \,. \tag{20}$$

Layer 4 describes the de-fuzzification process, wherein the estimation of each rule consequent is performed for the outputs specified, and equals:

$$J_{4,h} = \overline{\rho}_h E_h = \overline{\rho}_h (\mathrm{H}_h r + \mathrm{I}_h r + \mathrm{K}_h), \ \ \forall h = 1,\, 2 \,, \tag{21}$$

where H, I, and K indicate consequent parameter sets. The process of final outcome estimation is:

$$J_{5,h} = \sum_h \overline{\rho}_h E_h = \frac{\sum_h \rho_h E_h}{\sum_h \rho_h} \,. \tag{22}$$

The parameters are then distributed using random values and are tuned by means of training to obtain optimized results. The proposed A-RFATO is utilized in such training and the extent of hidden layers is used for efficient system training on a large amount of data.

3.11. DNFN Training

Below is presented an 8-step procedure used for training.

Step 1 – initialization of riders. The riders are initialized randomly by:

$$G_\sigma = \left\{ G_\sigma(p, q) \right\}; \ \ 1 \leqslant p \leqslant U; \ \ 1 \leqslant q \leqslant V \,, \tag{23}$$

where σ is a moment in time, U is the total number of riders, p, q is the p-th rider location $G_\sigma(p, q)$, and V denotes the total coordinates of the dimension. After group initialization, rider parameters are initialized as well: gear I_σ, steering S_σ, brake R_σ, and accelerator A_τ.

Step 2 – computation of fitness. Estimation of the fitness function is performed by identifying the space between the final destination and the location of the rider. The solution with a minimum error value is considered to be the optimal one. The following equation is used to compute the fitness value:

$$MSE = \frac{1}{v}\left[\sum_{j=1}^{v} Z_{target} - w^o\right] , \quad (24)$$

where Z_{target} is the target output, v denotes the number of training samples, and w^o indicates the estimated output from DNFN.

Step 3 – leading rider determination. After computation of the fitness function, the leading rider is calculated. The rider with lowest fitness value is selected to be the leading rider and its location varies at each moment.

Step 4 – updating rider location. The location of the leading rider is estimated by updating the rider location at a given time and the location of the bypass rider in the group. Then, the follower is updated.

The bypass rider avoids the regular route, without bearing in mind the pathway of the leading rider. Location of the bypass rider is updated by:

$$G_{\sigma_1}(p,q) = \lambda\Big[G_\sigma(\chi,q)\cdot\eta(q) + G_\sigma(\gamma,q)\cdot\big(1-\eta(q)\big)\Big] , \quad (25)$$

where χ and γ are the random numbers from 1 to R. An adaptive concept is realized by using η and terms λ for better performance of sentiment classification, expressed as:

$$\eta = \frac{fitness_i}{\max(fitness)} , \quad (26)$$

$$\lambda = \frac{\tau}{T} , \quad (27)$$

The follower rapidly achieves its destination by updating the location of the leading rider. The area of the follower is modified using the values chosen and the location of the follower is influenced by the coordinate selector as:

$$G_{\sigma+1}(p,q) = G^K(K,q) + \big\lfloor\cos(KF^\sigma_{p,q})\cdot G^K(K,q)\big\rfloor + g^\sigma_p , \quad (28)$$

where G^K and K indicate the leading rider's location and index, the distance of the p-th rider is denoted as g^σ_p, q is the coordinate selector and $KF^\sigma_{p,q}$ denotes the steering angle in the q-th coordinate of p-th rider.

The factors affecting the location update of the overtaker include the following: relative success rate, coordinate selector, and direction indicator. The overtaker provides faster convergence with a huge global neighborhood, thus the self-evaluation operator of FAT is used to update the value. The final equation of the overtaker updated with FAT is expressed as:

$$G_{pq}(\sigma+1) = \frac{[1-\mathrm{rand}(0,1)]}{\mathrm{rand}(0,1)}\left[\frac{G_{best,q}\mathrm{rand}(0,1)}{1-\mathrm{rand}(0,1)} + D^Z_p(\sigma)G^K_{Kq}\right], \quad (29)$$

where $G_{pq}(\sigma)$ is the p-th rider location in the q-th coordinate and $D^Z_p(\sigma)$ denotes the direction indicator of the p-th rider at instant σ. The location of the best branch is indicated by $G_{best,q}$, rand(0, 1) denotes a random number between 0 and 1.

Step 5 – updating attacker location. The attacker tries to reach the leader's location and uses the same update procedure as the follower. The attacker location is updated using the following formula:

$$G_{\sigma+1}(p,q) = G^K(K,q) + \big\lfloor\cos(KF^\sigma_{p,q})\cdot G^K(K,q)\big\rfloor + g^\sigma_p , \quad (30)$$

where $G_{\sigma+1}(p,q)$ is the leading rider location, $KF^\sigma_{p,q}$ denotes the steering angle in the q-th coordinate of the p-th rider and the distance of the p-th rider is expressed as g^σ_p.

Step 6 – calculation of fitness function. Once the location of each rider has been estimated, the rider with the best fitness value is selected as the race champion.

Step 7 – updating rider parameters. The activity counter is computed to update the rider parameters in order to identify the optimal solution.

Step 8 – termination. The process described above is iterated until the time is reached.

The pseudo code of the adaptive RFATO technique is specified in Algorithm 1.

Algorithm 1. Pseudocode of the proposed A-RFATO method.

1: Input: rider's random position G_σ.
2: Initialize the population
3: Initialize rider parameter: gear I_σ, steering S_σ, brake R_σ, and accelerator A_τ
4: Fitness function computation
5: While $\sigma < \sigma_{off}$
6: For $p = 1$ to U
7: Position update of bypass rider using Eq. (25)
8: Include adaptive concept by $\eta = \frac{fitness_i}{\max(fitness)}$ and $\lambda = \frac{\tau}{T}$
9: Modify the area of follower based on Eq. (28)
10: Modify the location of overtaker based on Eq. (29)
11: Position update of attacker based on Eq. (30)
12: Ranking of riders based on fitness function
13: Rider with maximum fitness function is taken as leading rider
14: Modify I_σ, S_σ, R_σ, and A_σ
15: Return G^K
16: $\sigma = \sigma + 1$
17: End for
18: End while
19: Output: Leading rider G^K

4. Results and Discussion

To evaluate performance of proposed algorithm, the following metrics are used:

Accuracy is determined in order to describe the correctness of the sentiment classification process and is expressed as:

$$ACC = \frac{T_{positive} + T_{negative}}{T_{positive} + T_{negative} + F_{positive} + F_{negative}} . \quad (31)$$

Specificity is estimated to identify actual negative sentiments and is denoted in:

$$Spe = \frac{T_{negative}}{T_{negative} + F_{positive}} . \tag{32}$$

Sensitivity is the measure used for finding definite positives and is estimated by:

$$Sen = \frac{T_{positive}}{T_{positive} + F_{negative}} . \tag{33}$$

Precision is defined as the ratio between accurately classified outcomes and all classified positive values:

$$Pre = \frac{T_{positive}}{T_{positive} + F_{positive}} . \tag{34}$$

where true positives, false positives, true negatives and false negatives are specified as $T_{positive}$, $F_{positive}$, $T_{negative}$ and $F_{negative}$, respectively.

The Large Movie Review Dataset [36], the Datafiniti Product Database [37], and Amazon Reviews [38] are utilized for experimentation purposes and for testing the sentiment classification model developed.

The Large Movie Review Dataset includes 50,000 highly polarized movie reviews for used in testing and training processes. The number of positive and negative review files used in testing is 12,500, while 12,500 negative opinions and 12,500 positive review files are employed for the training process.

The Datafiniti Product Database contains 34,000 reviews of such Amazon products as Kindle, Fire TV stick and so on. It consists of the review text, product information and a rating assigned to each product.

Amazon Reviews contain reviews provided by several million Amazon customers. Star ratings, commonly used for sentiment analysis, as available as well. The database contains real business data on a reasonable scale.

The proposed A-RFATO-based DNFN is compared with other existing techniques, such as CNN [28], RNSA [11], AL-SSVAE [29], GRNN-SR [4], and RFATO-based deep RNN. In the comparison, different metrics are used, such as specificity, sensitivity, accuracy, and precision.

4.1. Analysis Based on the Large Movie Review Dataset

A comparative analysis of the A-RFATO-based DNFN, performed with the use of the Large Movie Review Dataset by changing training data percentages, is depicted in Fig. 4. Figure 4a shows the accuracy analysis performed by altering training data percentages. The accuracy score obtained by CNN is 0.647, by GRNN-SR is 0.777, by RNSA is 0.778, by AL-SSVAE is 0.778 and by RFATO-based deep RNN is 0.795, whereas the value achieved by the proposed A-RFATO-based DNFN is 0.813 with 70% of the training data considered. Additionally, improvement in the performance of Adaptive RFATO-based DNFN is 17.92% vs. CNN, 4.39% vs. GRNN-SR, 4.30% vs. RNSA, 4.26% vs. AL-SSVAE, and 2.23% vs. RFATO-based deep RNN techniques.

The sensitivity analysis performed by changing the training data percentage is illustrated in Fig. 4b. Sensitivity for CNN is 0.709, for GRNN-SR is 0.823, for RNSA is 0.824, for AL-SSVAE is 0.824, and for RFATO-based deep RNN is 0.845. The developed method achieves the score of 0.864. Improvement in the performance of the proposed technique versus existing methods, such as CNN, is 17.92%. It equals 4.75% vs. GRNN-SR, 4.68% vs. RNSA, 4.65% vs. AL-SSVAE and 2.19% vs. the RFATO-based deep RNN approach.

The specificity analysis is shown in Fig. 4c. In 70% of the training data, the proposed model achieves specificity of 0.746, while the score of CNN equals 0.55, GRNN-SR 0.7, RNSA 0.70, AL-SSVAE 0.701, and RFATO-based deep RNN 0.723. In addition, the improvement in performance of the proposed method is 26.35% compare with CNN, 6.27% with GRNN-SR, 6.14% with GRNN-SR, 6.10% AL-SSVAE, and 3.17% with RFATO-based deep RNN. Figure 4d presents an analysis of proposed method based on precision-related metrics, with the training data volumes altered. In 70% of training data, the A-RFATO-based DNFN achieves a precision score of 0.854, while the value of existing techniques equal 0.701 for CNN, 0.814 for GRNN-SR, 0.814 for RNSA, 0.815 for AL-SSVAE and 0.836 RFATO-based deep RNN.

A comparative analysis of the A-RFATO-based DNFN, using the Large Movie Review Dataset and with a varying feature size is shown in Fig. 5. The analysis is based on the comparison of accuracy with the changing feature size (Fig. 5a. The accuracy value achieved by CNN is 0.777, by GRNN-SR is 0.855, by RNSA is 0.856, by AL-SSVAE is 0.856, by RFATO-based deep RNN is 0.879, and the rate obtained by the proposed A-RFATO-based DNFN equaled 0.898, with the feature size 3. Furthermore, the developed sentiment classification model achieves a higher performance improvement rate of 13.41%, compared with the results of 4.74%, 4.61%, 4.59% and 2.12% attained by the existing sentiment classification approaches, respectively.

A comparative analysis of sensitivity with varying feature sizes is shown in Fig. 5b. The proposed method scores 0.935, while CNN achieves the score of 0.832, GRNN-SR of 0.888, RNSA of 0.888, AL-SSVAE of 0.889, and RFATO-based deep RNN methods of 0.910, for the feature size of approx. 3. The improvement in performance of the developed sentiment classification approach versus CNN is 11.99%, vs. GRNN-SR is 5.08%, vs. RNSA is 5.03%, vs. AL-SSVAE is 4.95%, and vs. RFATO-based deep RNN is 2.70%. Figure 5c presents the results of a specificity analysis based on a varying feature size. With a feature size of 3, the proposed method obtains the specificity rate of 0.845, thus outperforming other classification models: CNN achieves the score of 0.7, GRNN-SR of 0.798, RNSA of 0.799, AL-SSVAE of 0.800, and RFATO-based deep RNN of 0.822. The proposed technique achieved a higher performance improvement of 17.23% compared with CNN, 5.54% with GRNN-SR, 5.44% with RNSA, 5.30% with AL-SSVAE, and 2.74%, with RFATO-based deep RNN. Figure 5d illustrates the results of a precision analysis based on a varying feature size. With the feature size of 3, the pro-

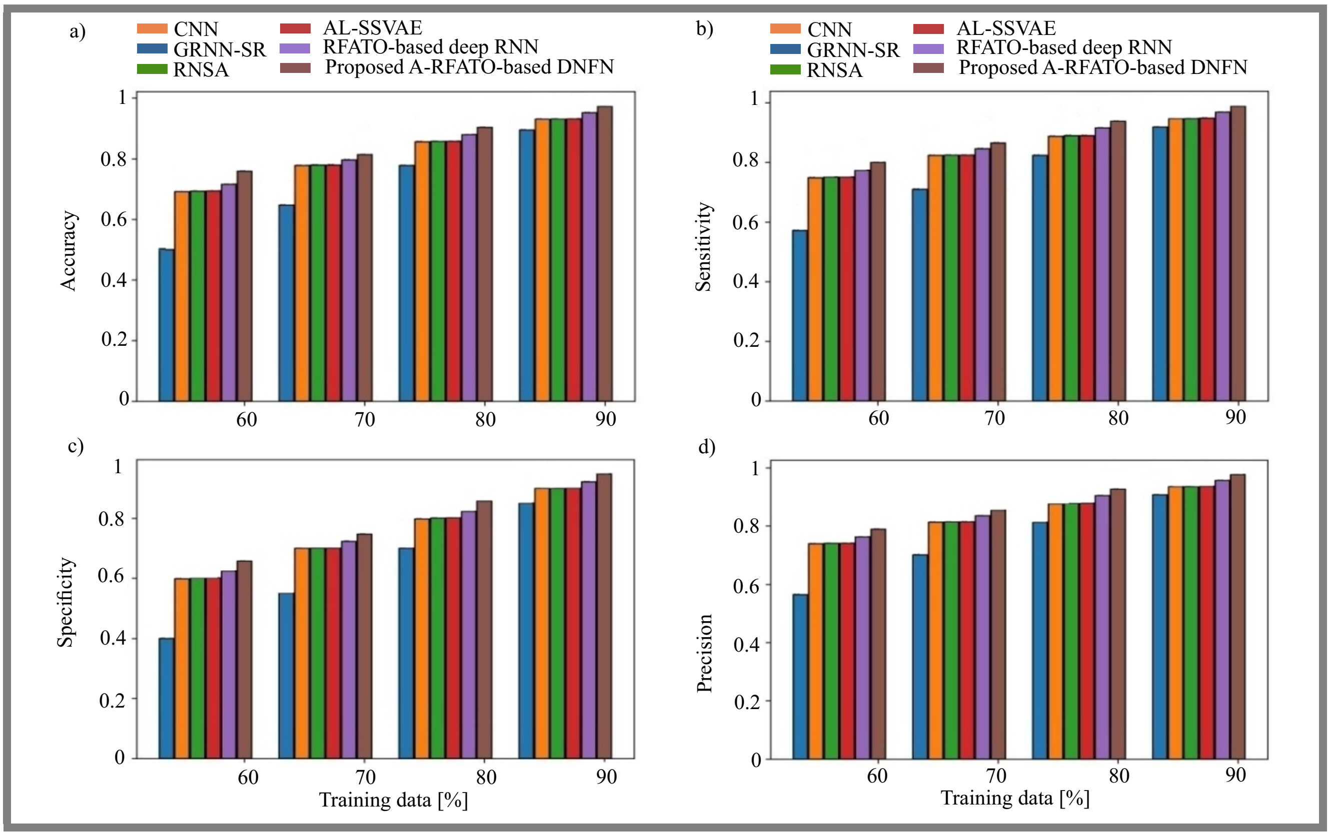

Fig. 4. Analysis of A-RFATO-based DNFN with the Large Movie Review Dataset depending on: a) accuracy, b) sensitivity, c) specificity, and d) precision with varying training data percentages.

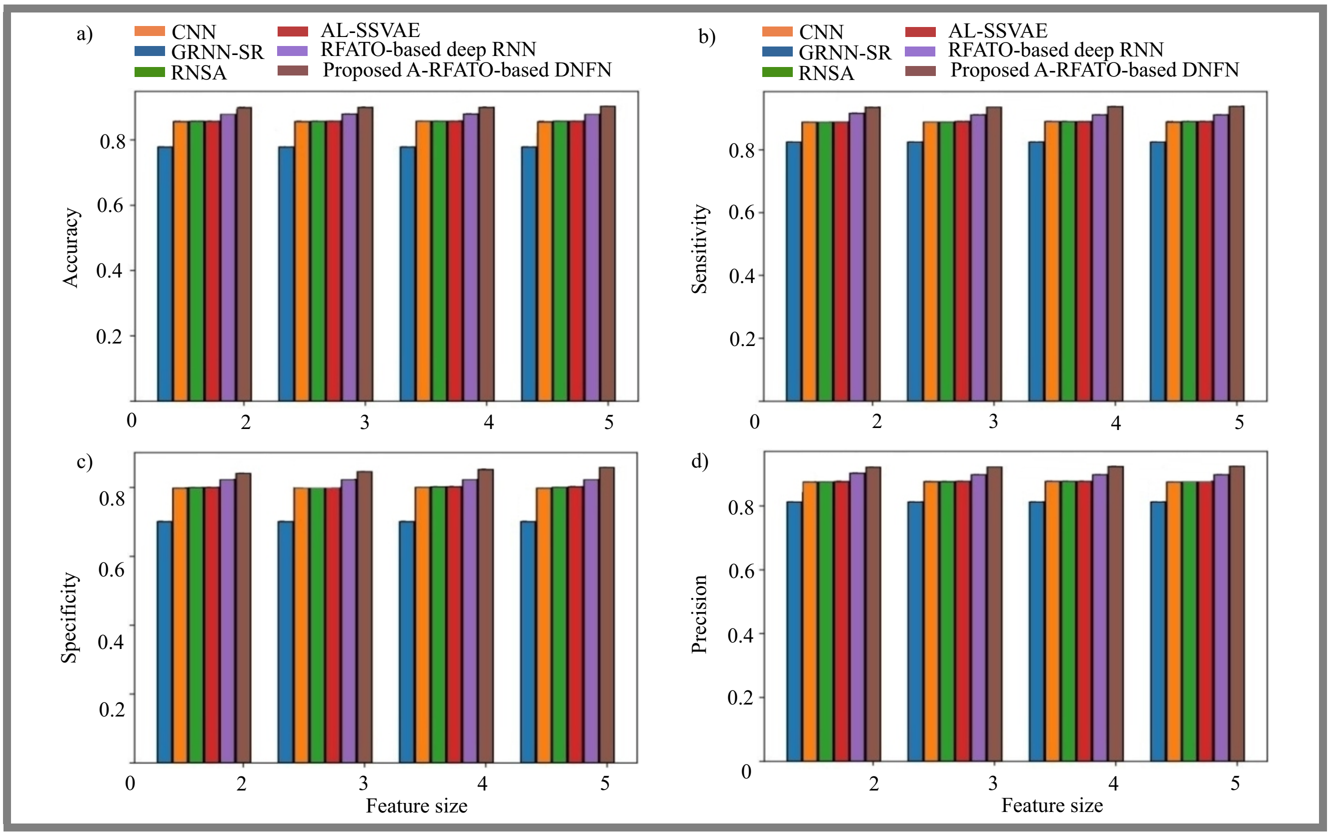

Fig. 5. Analysis of the proposed A-RFATO-based DNFN using the Large Movie Review Dataset with respect to: a) accuracy, b) sensitivity, c) specificity, and d) precision based on a varying feature size.

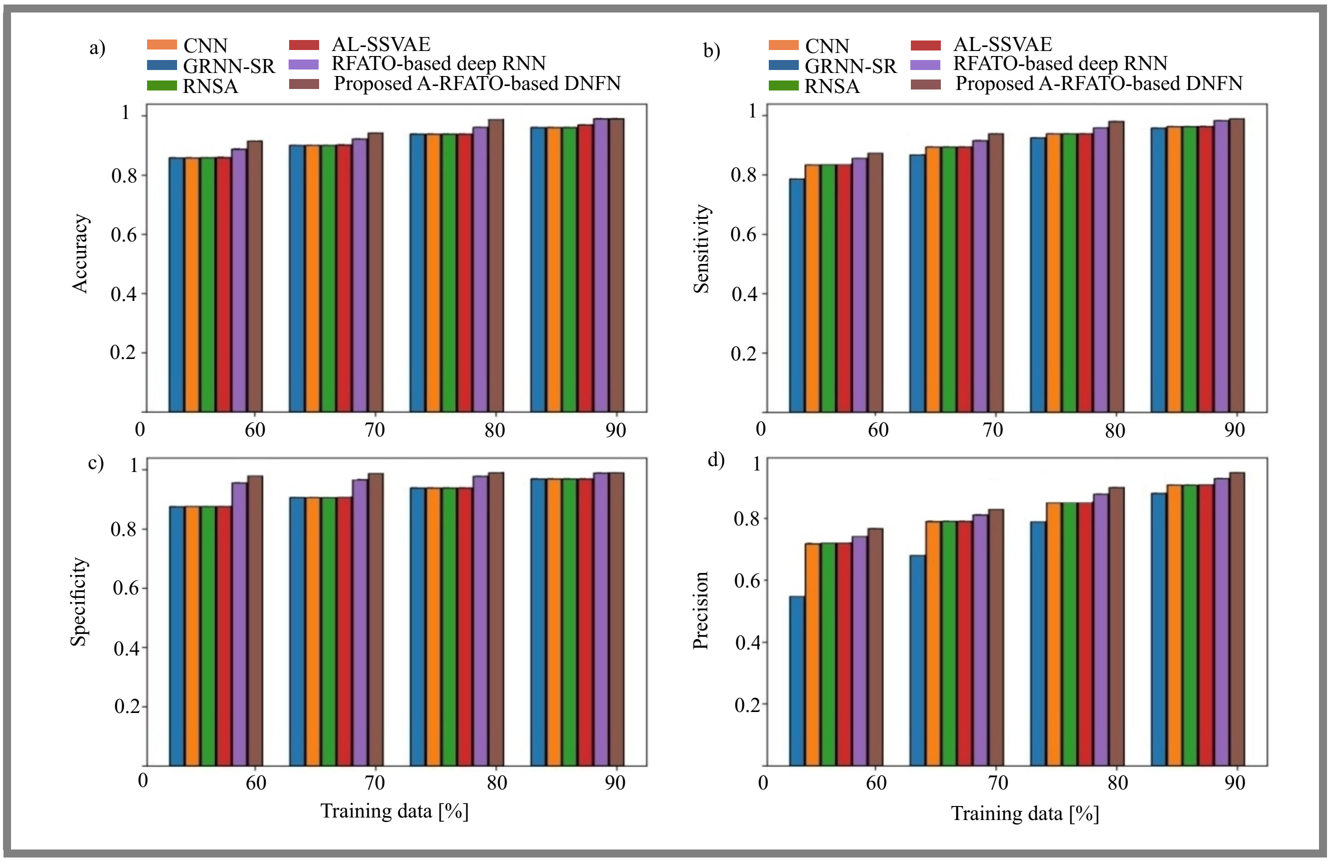

Fig. 6. Analysis of the proposed method using the Datafiniti Product Database, considering: a) accuracy, b) sensitivity, c) specificity, and d) precision with altering training data percentages.

posed method achieves the precision rating of 0.923 and also overperforms other sentiment classification models: CNN has the rating of 0.812, GRNN-SR of 0.877, RNSA of 0.877, AL-SSVAE of 0.877, and RFATO-based deep RNN of 0.898.

4.2. Analysis Based on the Datafiniti Product Database

A comparative analysis of the proposed A-RFATO-based DNFN using the Datafiniti Product Database with altering training data percentages is represented in Fig. 6. Figure 6a shows the accuracy analysis based on using different training data percentages. The existing techniques achieve the following accuracy scores: 0.937 for CNN, 0.937 for GRNN-SR, 0.938 for RNSA, 0.938 for AL-SSVAE, and 0.961 for RFATO-based deep RNN, whereas the proposed A-RFATO-based DNFN scores a ratio of 0.988 with 80% of training data. Moreover, A-RFATO-based DNFN shows an improvement in performance of 5.58% vs. CNN, 5.10% vs. GRNN-SR, 5.09% vs. RNSA, 5.06% vs. AL-SSVAE, and 2.68%, vs. RFATO-based deep RNN.

Results of the sensitivity analysis with changing training data percentages are presented in Fig. 6b. The sensitivity score of CNN is 0.924, of GRNN-SR is 0.937, of RNSA is 0.937, of AL-SSVAE is 0.938, and of RFATO-based deep RNN is 0.958. The proposed method achieves the ratio of 0.979. A specificity analysis of the proposed method with altering training data percentages is presented in Fig. 6c. With 80% of training data, the proposed model achieves the score of 0.99, where the existing techniques, such as CNN, RNSA and GRNN-SR, achieve the score of 0.937 each, AL-SSVAE reaches the value of 0.938, and RFATO-based deep RNN of 0.977. The improvement in performance of the proposed method versus CNN is 5.26%, vs. GRNN-SR is 5.26%, vs. RNSA is 5.25%, vs. AL-SSVAE is 5.23%, and vs. RFATO-based deep RNN is 1.23%. Figure 6d shows the precision measure for altering training data percentages. With 80% of training data, the developed model achieves the precision rating of 0.899, while CNN achieves the score of 0.790, RNSA of 0.851, GRNN-SR of 0.853, AL-SSVAE of 0.853, and RFATO-based deep RNN of 0.878.

Figure 7 shows a comparative analysis of the A-RFATO-based DNFN approach using the Datafiniti Product Database with a varying feature size. The comparative accuracy analysis with an altering feature size is shown in Fig. 7a. The values achieved by existing approaches are: CNN 0.937, GRNN-SR 0.937, RNSA 0.937, AL-SSVAE 0.938, RFATO-based deep RNN 0.961, with the A-RFATO-based deep neuro fuzzy network scoring 0.985 with the feature size of 4. The proposed sentiment classification model achieves a higher improvement in performance (4.89%, 4.88%, 4.87%, 4.82%, and 2.42%) compared with the present sentiment classification approaches, respectively.

Figure 7b shows a comparative sensitivity analysis based on a varying feature size. The proposed approach achieves

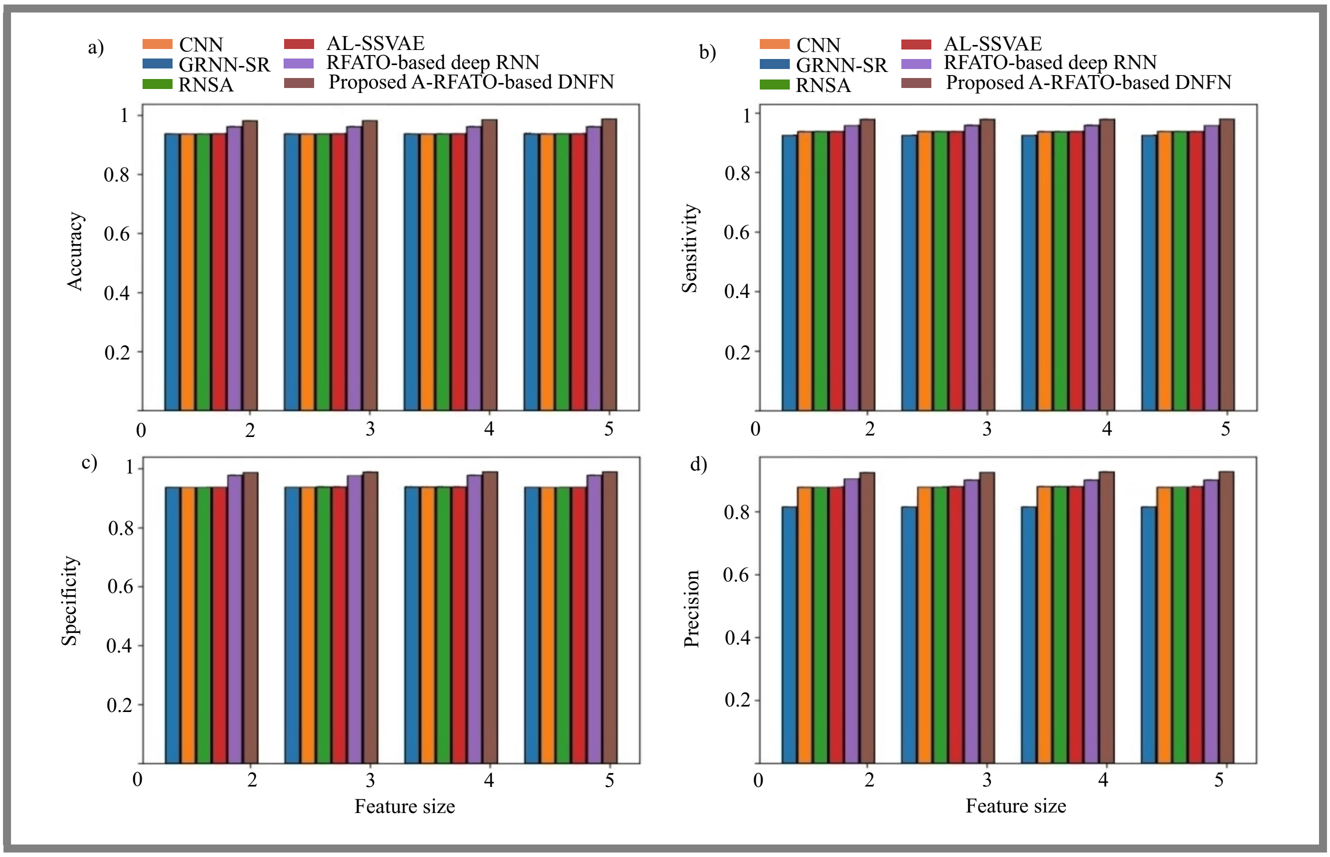

Fig. 7. Analysis of the proposed method using the Datafiniti Product Database, considering: a) accuracy, b) sensitivity, c) specificity, and d) precision with a varying feature size.

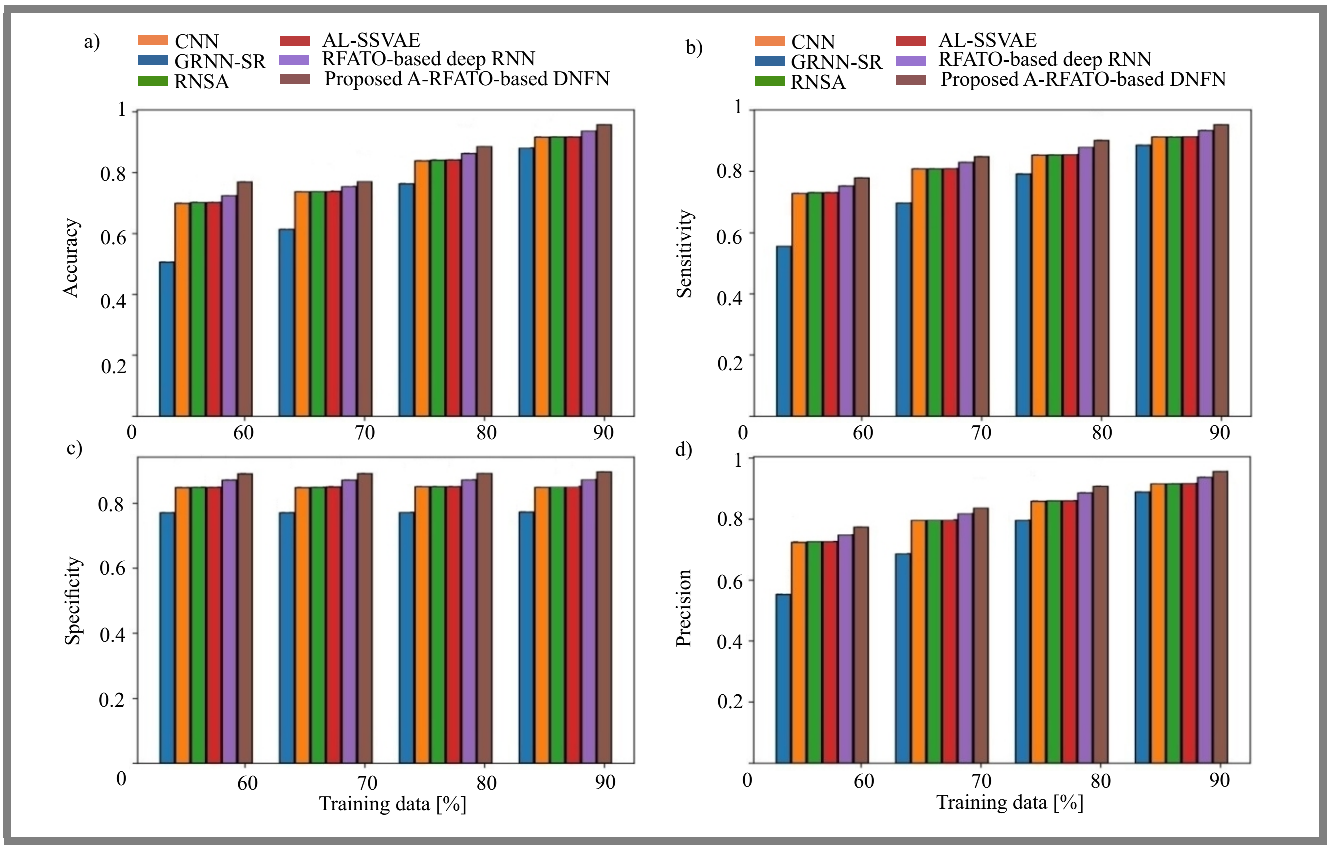

Fig. 8. Analysis of the proposed method based on Amazon Reviews, considering: a) accuracy, b) sensitivity, c) specificity, and d) precision, with varying training data percentages.

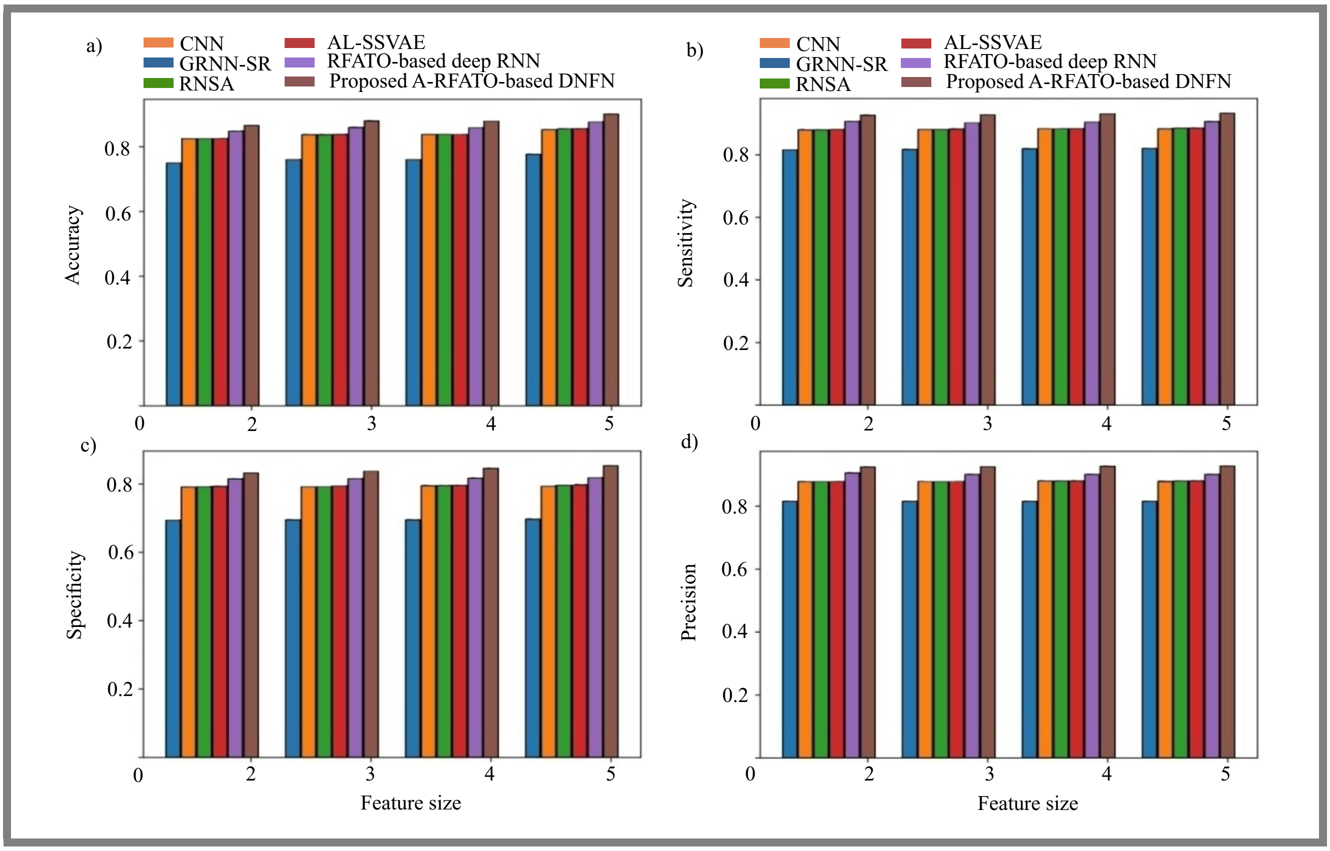

Fig. 9. Analysis of the proposed method using Amazon Reviews, considering: a) accuracy, b) sensitivity, c) specificity, and d) precision with varying feature sizes.

the value of 0.978, while CNN of 0.924, RNSA, GRNN-SR and AL-SSVAE of 0.937, and RFATO-based deep RNN of 0.958 (feature size 4). The improvement in performance of the proposed sentiment classification approach versus CNN is 5.51%, vs. GRNN-SR is 4.22%, vs. RNSA is 4.21%, vs. AL-SSVAE is 4.20%, and vs. RFATO-based deep RNN is 2.10%.

Figure 7c shows the results of a specificity analysis based on a varying feature size. For the feature size of 4, the proposed method reaches the specificity score of 0.99, CNN of 0.937, RNSA of 0.937, GRNN-SR of 0.937, AL-SSVAE of 0.938, and RFATO-based deep RNN of 0.977. The proposed technique achieves a performance improvement of 5.30% versus CNN, 5.29% vs. GRNN-SR, 5.28% vs. RNSA, 5.23% vs. AL-SSVAE, and 1.23% vs. RFATO-based deep RNN. Figure 7d shows the results of a precision analysis based on the feature size. For the feature size of 4, the proposed method reaches the precision score of 0.928, while CNN achieves the rating of 0.815, RNSA of 0.878, GRNN-SR of 0.880, AL-SSVAE of 0.881, and RFATO-based deep RNN of 0.901.

4.3. Analysis Based on the Amazon Reviews Dataset

Analysis of A-RFATO-based DNFN using the Amazon Reviews with an altering training data percentage is shown in Fig. 8. Figure 8a shows the results of the accuracy analysis with different training data percentages. The existing techniques achieve the following accuracy ratings: 0.763 for CNN, 0.839 for GRNN-SR, 0.841 for RNSA, 0.842 for AL-SSVAE, and 0.863 for RFATO-based deep RNN. The proposed A-RFATO-based DNFN achieved a score of 0.886 with 80% of training data. Figure 8b shows the results of the sensitivity analysis with changing training data percentages. Sensitivity of the existing sentiment classification methods are as follows: CNN 0.696, GRNN-SR 0.808, RNSA 0.809, AL-SSVAE 0.809, and RFATO-based deep RNN 0.830. The developed method obtained the score of 0.848 for 70% of training data.

An analysis of specificity-related performance of the proposed method, with varying training data percentages, is presented in Fig. 8c. With 80% of training data used, the proposed model achieves specificity of 0.893, while the existing techniques achieved the following results: CNN 0.772, RNSA 0.851, GRNN-SR 0.851, AL-SSVAE 0.851, and RFATO-based deep RNN 0.872. Analysis of the precision of the proposed method is presented in Fig. 8d. Using 80% of training data, the proposed model scores 0.907, while CNN achieves the rating of 0.796, RNSA of 0.858, GRNN-SR of 0.860, AL-SSVAE of 0.860, and RFATO-based deep RNN of 0.886.

Figure 9 illustrates results of the analysis of A-RFATO-based DNFN using Amazon Reviews, with varying feature sizes. The level of accuracy vs. the altering feature size is presented in Fig. 9a. The accuracy scores achieved by exist-

Tab. 1. Comparative study for the Large Movie Review Dataset.

Analysis	Metrics	GRNN-SR	CNN	RNSA	RFATO-based deep RNN	AL-SSVAE	**A-RFATO-based deep neuro fuzzy network**
Based on training data	Accuracy	0.930	0.894	0.931	0.951	0.931	**0.972**
	Sensitivity	0.947	0.918	0.947	0.968	0.947	**0.988**
	Specificity	0.899	0.85	0.900	0.921	0.900	**0.947**
	Precision	0.936	0.908	0.936	0.956	0.936	**0.977**
Based on feature size	Accuracy	0.855	0.777	0.857	0.879	0.857	**0.903**
	Sensitivity	0.887	0.823	0.889	0.910	0.889	**0.937**
	Specificity	0.800	0.797	0.801	0.857	0.822	**0.797**
	Precision	0.875	0.812	0.877	0.898	0.877	**0.925**

Tab. 2. Comparative summary for the Datafiniti Product Database.

Analysis	Metrics	GRNN-SR	CNN	RNSA	RFATO-based deep RNN	AL-SSVAE	**A-RFATO-based deep neuro fuzzy network**
Based on training data	Accuracy	0.960	0.960	0.960	0.989	0.970	**0.990**
	Sensitivity	0.962	0.957	0.962	0.982	0.962	**0.988**
	Specificity	0.968	0.968	0.968	0.988	0.969	**0.990**
	Precision	0.908	0.881	0.909	0.928	0.909	**0.948**
Based on feature size	Accuracy	0.937	0.937	0.938	0.961	0.938	**0.988**
	Sensitivity	0.937	0.924	0.937	0.958	0.938	**0.979**
	Specificity	0.937	0.937	0.937	0.977	0.938	**0.990**
	Precision	0.878	0.815	0.880	0.901	0.881	**0.928**

ing approaches are: CNN 0.760, GRNN-SR 0.838, RNSA 0.838, AL-SSVAE 0.838, RFATO-based deep RNN 0.859, and A-RFATO-based deep neuro fuzzy network 0.879, with the feature size of 4. Figure 9b shows a similar analysis focusing on sensitivity. Sensitivity of the proposed approach is 0.931, while that of CNN equals 0.819, RNSA 0.884, GRNN-SR 0.884, AL-SSVAE 0.937, and RFATO-based deep RNN 0.905 (feature size 4).

Figure 9c shows the specificity analysis. The developed method obtained a specificity score of 0.845, while CNN of 0.695, RNSA of 0.795, GRNN-SR of 0.796, AL-SSVAE of 0.796, and RFATO-based deep RNN of 0.817. Part “d” of Fig. 9 shows a precision analysis. The developed method scored 0.926, CNN 0.814, RNSA 0.879, GRNN-SR 0.880, AL-SSVAE 0.880, and RFATO-based deep RNN 0.901.

4.4. Comparative Summary

A summary of the performance of different techniques relied upon in the classification of sentiment using the Large Movie Review Dataset is presented in Tab. 1. The proposed method obtains better performance with 90% of training data and the feature size of 5, and achieves the maximum values of sensitivity, accuracy and specificity.

Comparison of the performance of different techniques used for sentiment grade classification based on the Datafiniti Product Database is presented in Tab. 2. The accuracy score achieved by the developed sentiment classification approach equals 0.99 (feature size 5), and its sensitivity ratio is 0.988. Likewise, the developed A-RFATO-based deep neuro fuzzy network obtains a specificity ratio of 0.99, which is better than that of its competitors. The proposed method obtains the best accuracy and sensitivity levels of 0.988 and 0.979, respectively, with the feature size of 5, when the percentage of training data considered is 90%. Hence, the table clearly indicates that the proposed A-RFATO-based DNFN approach achieves better accuracy, specificity, sensitivity, and precision scores than the existing methods.

The comparison of the techniques used in sentiment grade classification based on the Amazon Reviews dataset is shown in Tab. 3. The maximum accuracy, sensitivity, specificity, and precision scores attained by the A-RFATO-based deep neuro fuzzy network are 0.958, 0.953, 0.897, and 0.956, respectively.

5. Conclusion

This paper describes a technique for sentiment grade classification, known as A-RFATO-based DNFN. The classification process involves four stages: pre-processing stage, extraction of features, and feature fusion along with the classification of sentiment. The proposed A-RFATO scheme is designed by integrating the FAT technique and ROA with the adaptive concept. As a result, the developed sentiment grade classification technique allows to achieve the maximum values of

Tab. 3. Comparative data for the Amazon Reviews dataset.

Analysis	Metrics	GRNN-SR	CNN	RNSA	RFATO-based deep RNN	AL-SSVAE	**A-RFATO-based deep neuro fuzzy network**
Based on training data	Accuracy	0.917	0.881	0.917	0.938	0.917	**0.958**
	Sensitivity	0.913	0.886	0.913	0.933	0.914	**0.953**
	Specificity	0.849	0.772	0.852	0.873	0.852	**0.897**
	Precision	0.916	0.889	0.916	0.936	0.916	**0.956**
Based on feature size	Accuracy	0.853	0.775	0.855	0.877	0.855	**0.901**
	Sensitivity	0.883	0.819	0.885	0.906	0.885	**0.933**
	Specificity	0.793	0.697	0.797	0.819	0.797	**0.854**
	Precision	0.878	0.814	0.879	0.901	0.880	**0.927**

accuracy (0.99), sensitivity (0.988) and specificity (0.99). Future work related to this field of research will focus on optimizing the technique in order to enhance the classification process and tune it for data involving sarcasm.

References

[1] B. Liu, *Sentiment Analysis and Opinion Mining*. Springer Cham, serie *Synthesis Lectures on Human Language Technologies*, 2012 (ISBN: 9783031010170, https://doi.org/10.1007/978-3-031-02145-9).

[2] F. Chen, Z. Yuan, and Y. Huang, “Multi-source data fusion for aspect-level sentiment classification”, Knowledge-Based Systems, vol. 187, pp. 104831, 2020 (https://doi.org/10.1016/j.knosys.2019.07.002).

[3] Y. Wang, M. Huang, Z. Zhu, and L. Zhao, “Attention-based LSTM for aspect-level sentiment classification”, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 606–615, 2016 (https://doi.org/10.18653/v1/D16-1058).

[4] C. Chen, R. Zhuo, and J. Ren, “Gated recurrent neural network with sentimental relations for sentiment classification”, *Information Sciences*, vol. 502, pp. 268–278, 2019 (https://doi.org/10.1016/j.ins.2019.06.050).

[5] X. Li X and W. Lam, “Deep multi-task learning for aspect term extraction with memory interaction”, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2886–2892, 2017 (https://doi.org/10.18653/v1/D17-1310).

[6] E. Cambria, S. Poria, A. Gelbukh, and M. Thelwall, “Sentiment analysis is a big suitcase”, *IEEE Intelligent Systems*, vol. 2, no 6, pp. 74–80, 2017 (https://doi.org/10.1109/MIS.2017.4531228).

[7] L. Yue, W. Chen, X. Li, W. Zuo, and M. Yin, “A survey of sentiment analysis in social media”, *Knowledge and Information Systems*, vol. 60, no. 2, pp. 617–663, 2019 (https://doi.org/10.1007/s10115-018-1236-4).

[8] M.U. Salur and I. Aydin, “A novel hybrid deep learning model for sentiment classification”, *IEEE Access*, vol. 8, pp. 58080–93, 2020 (https://doi.org/10.1109/ACCESS.2020.2982538).

[9] Z. Wang and Z. Lin, “Optimal feature selection for learning-based algorithms for sentiment classification”, *Cognitive Computation*, vol. 12, no. 1, pp. 238–248, 2020 (https://doi.org/10.1007/s12559-019-09669-5).

[10] C. Zhang, Q. Li, and D. Song, “Aspect-based sentiment classification with aspect-specific graph convolutional networks”, *Proceedings of the 2019 Conf. on Empirical Methods in Natural Language Processing and the 9th International Joint Conf. on Natural Language Processing* (*EMNLP-IJCNLP*), pp. 4568–4578, 2019 (https://doi.org/10.18653/v1/D19-1464)

[11] A. Abdi, S.M. Shamsuddin, S. Hasan, and J. 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 (https://doi.org/10.1016/j.ipm.2019.02.018).

[12] Y. Zhang, Z. Zhang, D. Miao, and J. Wang, “Three-way enhanced convolutional neural networks for sentence-level sentiment classification”, *Information Sciences*, vol. 477, pp. 55–64, 2019 (https://doi.org/10.1016/j.ins.2018.10.030).

[13] J. Serrano-Guerrero, J.A. Olivas, F.P. Romero, and E. Herrera-Viedma, “Sentiment analysis: A review and comparative analysis of web services”, *Information Sciences*, vol. 311, pp. 18–38, 2015 (https://doi.org/10.1016/j.ins.2015.03.040)

[14] S. Zhang, X. Xu, Y. Pang, and J. Han, “Multi-layer attention based CNN for target-dependent sentiment classification”, *Neural Processing Letters*, vol. 51, no. 3, pp. 2089–2103, 2020 (https://doi.org/10.1007/s11063-019-10017-9).

[15] C. Zhao, S. Wang, and D. Li, “Multi-source domain adaptation with joint learning for cross-domain sentiment classification”, *Knowledge-Based Systems*, vol. 191, pp. 105254, 2020 (https://doi.org/10.1016/j.knosys.2019.105254).

[16] H. Chen, M. Sun, C. Tu, Y. Lin, and Z. Liu, “Neural sentiment classification with user and product attention”, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1650–1659, 2016 (https://doi.org/10.18653/v1/D16-1171).

[17] M. Gangappa, K.C. Mai C, and P. Sammulal, “Enhanced crow search optimization algorithm and hybrid NN-CNN classifiers for classification of land cover images”, *Multimedia Research*, vol. 2 , no. 3, pp. 12–22, 2019 (https://doi.org/10.46253/j.mr.v2i3.a2).

[18] S. Vasamsetti and C. Santhirani, “Hybrid particle swarm optimization-deep neural network model for speaker recognition”, *Multimedia Research*, vol. 3, no. 1, pp. 1–10, 2020 (https://doi.org/10.46253/j.mr.v3i1.a1).

[19] G. Parente, T. Gargano, G. Ruggeri, M. Maffi, S. D’Antonio, E. Sacchet, and M. Lima, “Anastomotic stricture definition after esophageal atresia repair: role of endoscopic stricture index”, *Journal of Surgical Research*, vol. 257, pp. 572–578, 2021 (https://doi.org/10.1016/j.jss.2020.08.035).

[20] G. Parente, T. Gargano, S. Pavia, C. Cordola, M. Vastano, Baccelli, G. Gallotta, L. Bruni, A. Corvaglia, and M. Lima, “Pyelonephritis in pediatric uropathic patients: Differences from community-acquired ones and therapeutic protocol considerations. A 10-year single-center retrospective study”, *Children*, vol. 8, no. 6, 2021 (DOI: 10.3390/children8060436).

[21] R.K. Bakshi, N. Kaur, R. Kaur, and G. Kaur, “Opinion mining and sentiment analysis”, *Proceedings of 2016 3rd International Conference on Computing for Sustainable Global Development* (*INDIACom*), pp. 452–455, 2016.

[22] D. Ma, S. Li, X. Zhang, and H. Wang, “Interactive attention networks for aspect-level sentiment classification”, *Proceedings of the 26th International Joint Conference on Artificial Intelligence* (*IJCAI-17*), pp. 4068–4074, 2017 (https://www.ijcai.org/proceedings/2017/0568.pdf).

[23] R. John Martin, “EEG Feature Engineering Methods-A Comprehensive Review”, *Multimedia Research*, vol. 5, no. 2, 2022 (URL: https://publisher.resbee.org/mr/archive/v5i2/a5.html).

[24] S.A.J. Al Raisi, “A review on congestion management methodologies and its applications”, *Journal of Computational Mechanics, Power System and Control*, vol. 3, no. 3, 2020 (https://doi.org/10.46253/jcmps.v3i3.a3).

[25] M. Adil, R. Madani, S. Tavakkol, and A. Davoudi, “A First-Order Numerical Algorithm without Matrix Operations”, arXiv preprint arXiv:2203.05027, 2022 (https://arxiv.org/pdf/2203.05027).

[26] L. Dong, F. Wei, C. Tan, D. Tang, M. Zhou, and K. Xu, “Adaptive recursive neural network for target-dependent twitter sentiment classification”, *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, vol. 2, pp. 49–54, 2014 (https://doi.org/10.3115/v1/P14-2009).

[27] K.S. Tai, R. Socher, C.D. Manning, “Improved semantic representations from tree-structured long short-term memory networks”, *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conf. on Natural Language Processing*, vol. 1, pp. 1556–1566, 2015 (https://doi.org/10.3115/v1/P15-1150).

[28] H. Kim and Y.S. Jeong, “Sentiment classification using convolutional neural networks”, *Applied Sciences*, vol. 9, no. 11, pp. 2347, 2019 (https://doi.org/10.3390/app9112347).

[29] X. Fu, Y. Wei, F. Xu, T. Wang, Y. Lu, J. Li, and J.Z. Huang, “Semi-supervised aspect-level sentiment classification model based on variational autoencoder”, *Knowledge-Based Systems*, vol. 171, pp. 81–92, 2019 (https://doi.org/10.1016/j.knosys.2019.02.008)

[30] P. Zhao, L. Hou, and O. Wu, “Modeling sentiment dependencies with graph convolutional networks for aspect-level sentiment classification”, *Knowledge-Based Systems*, vol. 193, pp. 105443, 2020 (https://doi.org/10.1016/j.knosys.2019.105443).

[31] K. Zhang, M. Jiao, X. Chen, Z. Wang, B. Liu, and L. Liu, “SC-BiCapsNet: a sentiment classification model based on bi-channel capsule network”, *IEEE Access*, vol. 7, pp. 171801–171713, 2019 (https://doi.org/10.1109/ACCESS.2019.2953502).

[32] N. Jin, J. Wu, X. Ma, K. Yan, Y. Mo, “Multi-task learning model based on multi-scale CNN and LSTM for sentiment classification”, *IEEE Access*, vol. 8, pp. 77060–77072, 2020 (https://doi.org/10.1109/ACCESS.2020.2989428).

[33] B. Ohana and B. Tierney, “Sentiment classification of reviews using SentiWordNet”, *9th IT&T Conference*, 2009 (https://doi.org/10.21427/D77S56).

[34] N.M.K. Saeed, N.A. Helal, N.L. Badr, and T.F. Gharib, “The impact of spam reviews on feature-based sentiment analysis”, *Proceedings of 13th International Conference on Computer Engineering and Systems* (*ICCES*), 2018 (https://doi.org/10.1109/ICCES.2018.8639343).

[35] S. Javaid, M. Abdullah, N. Javaid, T. Sultana, J. Ahmed, and N.A. Sattar, “Towards buildings energy management: using seasonal schedules under time of use pricing tariff via deep neuro-fuzzy optimizer”, *Proceedings of 15th International Wireless Communications & Mobile Computing Conference* (*IWCMC*), 2019, pp. 1594–1599 (https://doi.org/10.1109/IWCMC.2019.8766673).

[36] –, Large movie review dataset taken from (http://ai.stanford.edu/~amaas/data/sentiment), accessed on Dec. 2020.

[37] –, Consumer Reviews of Amazon Products dataset taken from (https://www.kaggle.com/datafiniti/consumer-reviews-of-amazon-products), accessed on Dec. 2020.

[38] –, Amazon Reviews for Sentiment Analysis dataset taken from, 2022 (https://www.kaggle.com/datasets/bittlingmayer/amazonreviews), accessed on Nov. 2022

Sireesha Jasti
Research Scholar
https://orcid.org/0000-0002-4755-5033
E-mail: sireeshajasti14@gmail.com
Department of Computer Science and Engineering, GITAM School of Technology, GITAM (Deemed to be University), Visakhapatnam, Andhra Pradesh, India

G.V.S. Raj Kumar, Prof.
E-mail: gganapav@gitam.edu
Department of Computer Science and Engineering, GITAM School of Technology, GITAM (Deemed to be University), Visakhapatnam, Andhra Pradesh, India