



# Sentiment based emotion classification in unstructured textual data using dual stage deep model

S J R K Padminivalli V<sup>1</sup> · M. V. P. Chandra Sekhara Rao<sup>2</sup> · Naga Sai Ram Narne<sup>3</sup>

Received: 14 September 2022 / Revised: 18 May 2023 / Accepted: 13 July 2023 /

Published online: 9 August 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

## Abstract

Classification of sentiments is an essential task in Natural Language Processing (NLP) domain. The powerful sentiment classification helps determine user opinions in product reviews or social networks. However, comprehending hidden opinions, sentiments, and emotions in emails, tweets, reviews, and comments is a challenge and equally crucial for social media monitoring, brand monitoring, customer services, and market research. Also, unstructured data in social media remains a major issue, and a proficient technique to deal with this issue remains a research gap. Therefore, this work presents an automated sentiment polarity and emotion classification in unstructured textual data using dual stage deep learning framework. Initially, pre-processing is performed to remove the noises and promote the quality of input data using stop words removal, Parts-Of-Speech (POS) tagging and duplicates removal. Then, the most discriminative features are extracted in the feature extraction stage, and the optimal set of features is selected to minimize the large feature dimensionality. Finally, the selected features are provided to the Dual-stage Deep Model to classify sentiments and emotions. The proposed classification stage classifies the sentiment and emotions from the given input data. The proposed work used three datasets for simulation analysis, and each dataset's performance is determined. Using Twitter Sentiment Dataset, the proposed model obtains an accuracy of 99.80%, F1-measure of 99.667%, specificity of 99.85% and kappa value of 99.52%, IMDB Movie Reviews attains an accuracy of 99.75%, F1-measure of 99.47%, specificity of 99.75% and kappa value of 98.99% and Yelp Reviews Dataset attains accuracy of 99.83%, F1-measure of 99.6%, specificity of 99.83% and kappa value of 99.32%. The obtained results reveal the effectiveness of a proposed study.

**Keywords** Sentiment analysis · Emotion classification · Pre-processing · Latent semantic analysis · Chaotic artificial hummingbird algorithm · Dual-stage deep model · Convolutional gated attention recurrent unit

## 1 Introduction

Social media is a powerful medium for sharing sentiments and opinions about a particular topic, focusing on content-sharing, communication, interaction, and collaboration. Sentiment indicates an individual's feelings regarding a post or product viewed on social media, and opinion indicates the thoughts expressed by an individual over the post or item [11, 25]. Through numerous social media sites like Facebook, Twitter, and Flickr, people may communicate with family, friends, and different communities around the world through text, photographs, and video [5, 8]. The emergence of the Internet allows users to post sentiment-related content about products [10], people, book, research, hotels [12], events and many others. With the proliferation of the Internet, massive unstructured data can be generated online and creates high risk with the information extraction procedures, whereas decision-making is a great challenge. The presence of unstructured data prohibits the classification methodologies from accurately processing the input textual content, consisting of different slang conveying more emotions [3]. Thus, automatic analysis of sentiments and emotions has become a widely researched domain because of the enormous availability of data on the social web.

In Natural Language Processing (NLP), sentiment analysis (SA) and emotion recognition (ER) are the dual critical research areas [13, 17]. However, both names can be utilized interchangeably but completely differs in some respects. In Sentiment Analysis, the most important concern is polarity which aims to assess whether the data is positive (P), negative (N) or neutral (Ne) [6, 24]. But emotion identification [4, 26] aims to identify the various human emotions such as happy, sad, angry, fear, disgust and surprise. People easily express their opinions, arguments, and feelings on social media focusing on different topics. People's active feedback helps marketers better understand user perspectives to improve their business strategies.

Sentiment, as well as Emotion analysis, offers a wide variety of applications such as monitoring conversations in social media, customer satisfaction and feedback evaluation, recognizing facial emotions [1, 15, 22], identifying physiological behaviours, helping in better decision-making and other Human–Computer Interaction (HCI) applications [18]. The methodologies like machine learning (ML) [2, 27], Lexicon [23], and Deep Learning (DL) [14, 21] can be used to perform sentiment and emotion analysis. Recently, DL has developed as a popular technique which learns features or representations from multiple layers and predicts the outcome. With the rising advances of deep Learning towards many applications, DL is being applied in sentiment and emotion detection. The DL model Convolutional Neural Network (CNN) is adopted only for visual data processing but recently gained improved results with textual inputs. Deep Learning can also be used for data mining and text analysis, enhancing the accuracy of sentiment analysis and emotion detection.

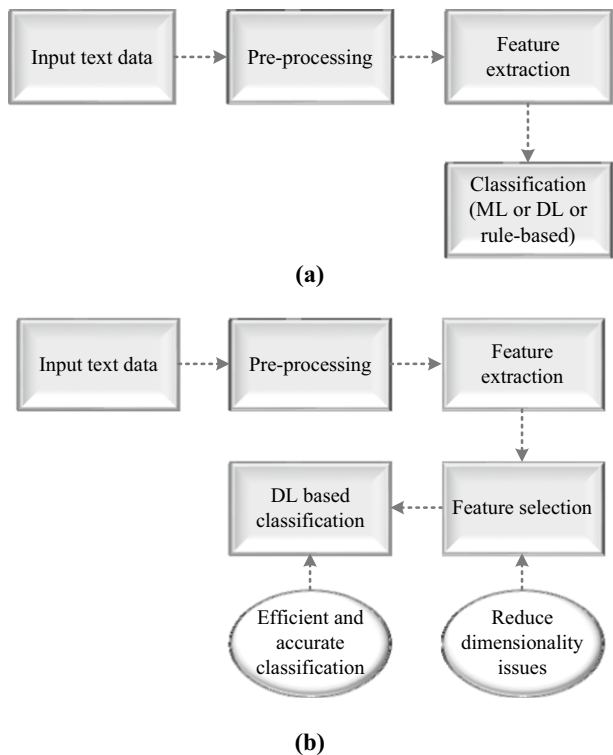
**Motivation, contribution and organization of this research paper** Social networking platforms, namely Twitter, Facebook, LinkedIn, Instagram and others platforms, have become an important source of communication or conveying emotions globally due to the rapid proliferation in the Internet era. People use textual content, audio, pictures, and video data to express their viewpoints or feelings. In web-based networking, text communication is very problematic. Each second, a huge amount of data is generated in unstructured form with the ease of social media platforms. With technical advances, the data can be generated rapidly and processed to understand human psychology using sentiment analysis which identifies the text polarity. The sentiment-based data mainly affords valuable information

for various applications. Classifying sentiments from the provided input data is a challenging task. Despite the availability of multiple techniques for sentiment analysis, researchers find it difficult to accurately identify the emotion behind the statement due to the conveyance of multiple emotions, slang used, ridicule, lexical and syntactical ambiguity. Recently, several existing works used automated deep-learning techniques for emotion classification using text-based data. These deep learning mechanisms provide improved classification results compared to other machine learning techniques and are more scalable. Moreover, deep learning techniques can learn large-level features from the given input data that assist the system in generating higher performance. Thus, the author prefers a robust deep-learning model for sentiment-based emotion classification. The general hierarchy of existing and proposed sentiment classification frameworks is presented in Fig. 1.

The Figure shows that the proposed framework is improved with the feature selection stage to optimize the feature set and enhance the overall classification accuracy. Moreover, the classification is performed with the deep Learning based approach, which is more efficient than traditional techniques. The major contributions of this proposed work are described as,

- This work introduces an automated sentiment and emotion classification framework based on deep Learning to analyze the emotional tone behind a series of words.

**Fig. 1** a Hierarchy of the existing sentiment classification frameworks b Hierarchy of the proposed sentiment classification framework



- The proposed framework extracts the most crucial features from the input textual content using Latent Semantic Analysis (LSA) and Log Term Frequency-based Modified Inverse Class Frequency (LTF-MICF) to enhance the overall classification accuracy.
- To reduce the classifier's dimensionality and computational complexity issues, the proposed framework selects the most optimal set of features using the chaotic artificial hummingbird algorithm (CAHA).
- Further, the selected features are provided to the Dual-stage Deep model called Convolutional Gated Recurrent Unit (CGARU) model to perform multi-class emotion prediction based on sentiment polarity.
- To evaluate the performance on various datasets and compare with the baseline classifier model to describe the efficacy of the proposed network.

The rest of this research paper is distributed as follows. Section 2 presents a literature review of the recent works on sentiment classification using different techniques. Section 3 elaborates on the proposed sentiment-based emotion classification model. Section 4 covers the simulation results and analysis of the proposed study. Finally, Section 5 concludes the paper with future scopes.

## 2 Related work

Some of the works related to this research are:

Karthik and Sethukarasi [16] presented a deep learning framework and Centered Convolutional Restricted Boltzmann Machines (CCRBM) to analyze user behaviour sentiments. The Deep Belief Network (DBN) was utilized to classify the sentiments accurately since this network performs dimensionality reduction and extraction of in-depth sentimental features. The parameters of a DBN architecture were optimized using Hybrid Atom Search Arithmetic Optimization (HASAO) and solved the issues with instability and randomness. Nine datasets were executed and tested to prove the effectiveness of a CCRBM method. The accuracy attained with polarity classes was positive (96%), neutral (90%) and negative (97%). The limitations were lower classification accuracy and required more training time.

Chen et al. [7] introduced deep Learning based sentiment analysis to explore social media data. For sentiment analysis, the data were collected from the largest online forum, PTT board-Military Life. Chinese words were segmented by combining sentiment dictionaries and the Jieba system. The deep learning models such as Long Short Term Memory (LSTM) and Bidirectional LSTM (Bi-LSTM) were used to perform training following the Word2vector (W2V) conversion. The activation function used for modelling was the Tanh function, which effectively enhances sentiment classification. The accuracy outcome obtained was 92.68%, whereas the existing models obtained MILSentic (82.41%) and self-developed senti-dictionary (84.08%). The limitation was that sentiment analysis only focuses on Militarylife PTT board data and not social media military-related data.

Singh et al. [20] developed deep learning LSTM-Recurrent Neural Network (LSTM-RNN) framework for analyzing sentiments of human emotions on COVID-19 Twitter reviews. Text pre-processing was done to eliminate the noisy or irrelevant features, hashtag removal, Internet slang removal, and white space removal. The extraction of features was processed with TF-IDF (Term Frequency-Inverse Document Frequency) method. The introduction of an attention mechanism allows efficient weight values for the features. The LSTM-RNN network classifies the data labels into 4 classes: fear, sad, joy and anger. The

performance was analyzed using accuracy (84.56%), recall (82.12%), precision (82.34%) and F1-score (81.23%). The approach's main drawback was the limited number of training instances used to train the model.

Du et al. [9] presented a new capsule network-based Hybrid Neural Network (HNN) for classifying sentiments. The deep learning capsule-HNN was formulated to obtain implicit semantic information efficiently. The different steps used for classifying the sentiments were Semantic Representation (SR) module, Word Attention (WA) module, Capsule module, Feature Extraction (FE) module and Classification module. The word embedding procedure illustrates each word as a multi-dimensional vector. The semantic features among the words in similar sentences were captured using the Self-attention mechanism. The extraction of features was adopted with CNN, and the reduction of feature dimensions was processed with K-max pooling. Combining a capsule and a Bidirectional Gated Recurrent Unit (BGRU) executes sentiment classification. The approach was evaluated using the Movie Review and NLPCC data. The main limitations of the approach were that the time complexity was high and the model resulted in a lower accuracy rate.

Pathak et al. [19] introduced a topic-based sentiment analysis using deep Learning on social media data. This topic-level sentiment analysis aims to detect topics at the sentence level via online Latent Semantic (LS) indexing. The LSTM framework was introduced to detect the topics with the attention mechanism. Three different datasets with the hashtags #bitcoin, #facebook and #ethereum were gathered from the Twitter site. The topic-level sentiment analysis focus on categorizing the positive and negative classes. The accuracy value estimated was Ethereum (0.846), facebook (0.794) and bitcoin (0.824). The main limitation of the model was that the sentence streaming procedure resulted only in detecting a single topic. Table 1 represents some existing studies and methods utilized for sentiment classification in recent years.

**Problem statement** In recent decades, sharing comments on the social media environment has increased rapidly. The emotions expressed in such comments assist many essential downstream applications. Hence, it is important to establish a powerful emotion classification method to detect emotions from online posts or reviews automatically. Various exiting works try to develop a robust sentiment and emotion classification technique but fail to bring enhanced results. Some of the real-world machine learning techniques face data imbalance issues. This issue directly affects the performance of a classification process. Moreover, extracting and selecting the optimal features from the given text data is essential to obtain higher results in the classification stage. But some of the techniques cannot extract the needed features from the text data and generate many problems while selecting suitable features for accurate classification. This creates higher computational complexity and degrades the system's efficiency. Thus, the proposed study proposes an effective automated sentiment-based emotion classification model using a deep learning method to overcome such issues mentioned in the existing studies.

### 3 Proposed methodology

Sentiment analysis is one of the efficient models for extracting opinion mining with identification and classification from unstructured text data. The text reviews in social media carry a large degree of colloquialism, minimum sentence integrity and non-standard grammatical structures. Hence, it is difficult to accurately categorize the

**Table 1** Existing studies and their limitations

| Authors             | Technology/Methods | Purpose  | Parameters   | Limitations  |
|---------------------|--------------------|--|--|--|
| Karthik et al. [16] | CCRBM, HASAO, DBN  | <ol style="list-style-type: none"> <li>To analyze behavioural sentiments from the provided data</li> <li>To analyze the sentiments with reduced computational time</li> </ol>            | <ol style="list-style-type: none"> <li>Accuracy</li> <li>Speed up capacity</li> </ol>  | The computation process is highly expensive  |
| Chen et al. [7]     | LSTM, Bi-LSTM      | <ol style="list-style-type: none"> <li>Developing a sentimental analysis model for social media</li> <li>To enhance the sentiment classification using a deep learning method</li> </ol> | <ol style="list-style-type: none"> <li>Accuracy</li> <li>F-measure</li> </ol>  | <ol style="list-style-type: none"> <li>Takes more time to process</li> <li>Stuffers from overfitting issues</li> <li>Bi-LSTM consumes additional time for training the input data</li> </ol> |
| Singh et al. [20]   | LSTM-RNN           | <ol style="list-style-type: none"> <li>To develop an automatic model for analyzing sentiments from COVID-19 reviews on the Twitter platform</li> </ol>                                   | <ol style="list-style-type: none"> <li>Accuracy</li> <li>Precision</li> <li>Recall</li> <li>F-score</li> </ol>   | <ol style="list-style-type: none"> <li>Some of the features available in the data are noisy. This may cause performance degradation</li> </ol>   |
| Du et al. [9]       | BGRU               | <ol style="list-style-type: none"> <li>To obtain implicit semantic information and perform effective sentiment classification</li> </ol>   | <ol style="list-style-type: none"> <li>Accuracy</li> <li>Precision</li> <li>Recall</li> <li>F-measure</li> </ol>   | <ol style="list-style-type: none"> <li>Attained reduced performance due to high feature dimensionality</li> <li>Increased computational complexity</li> </ol>                                |
| Pathak et al. [19]  | LSTM               | <ol style="list-style-type: none"> <li>To detect topic-level sentiment using social media data</li> </ol>  | <ol style="list-style-type: none"> <li>Accuracy</li> <li>F1-score</li> <li>Throughput</li> <li>Average time</li> <li>Precision</li> </ol>                      | <ol style="list-style-type: none"> <li>Highly sensitive to varied random initialization of weights</li> <li>Consumes additional memory to train the input data</li> </ol>                    |
| Proposed            | CGARU              | <ol style="list-style-type: none"> <li>To perform automated sentiment polarity and emotion classification from unstructured textual data</li> </ol>                                      | <ol style="list-style-type: none"> <li>Precision</li> <li>Recall</li> <li>F-score</li> <li>Accuracy</li> <li>Specificity</li> <li>Kappa coefficient</li> </ol> | -  |

sentiment polarity and emotions from the text data. Thus, it is inspired to propose an Artificial Intelligence (AI) guided intelligent approach based on Bio-inspired CAHA with CGARU for the sentiment classification process.

The proposed approach of sentiment analysis comprises four main phases: 1) Text Pre-processing (TP), 2) Textual Features Extraction (TFE), 3) Bio-inspired Feature Selection (BFS), and 4) Sentiment Polarity based Emotion Classification (SPEC). Pre-processing steps are carried out on the raw tweets data, such as stop words removal, Parts-Of-Speech (POS) tagging, and duplicate removal. Further, the TFE process is performed on pre-processed data for additional processing using LSA and LTF-MICF models. This work especially employs the CAHA technique for the feature selection process to detect the optimal subset of informative features from the n-number of textual features. Finally, the output of CAHA data is rendered to the Dual-stage Deep Model named CGARU. The first stage (convolutional network) classifies the sentiment as positive, negative, and neutral. Then, based on the sentiments (positive, negative, neutral), the second stage (GARU) identifies the multi-class emotions as Positive (Happiness, Love, Surprise), Negative (Stress, Regret, Disgust) and Neutral. The proposed research framework is illustrated in Fig. 2.

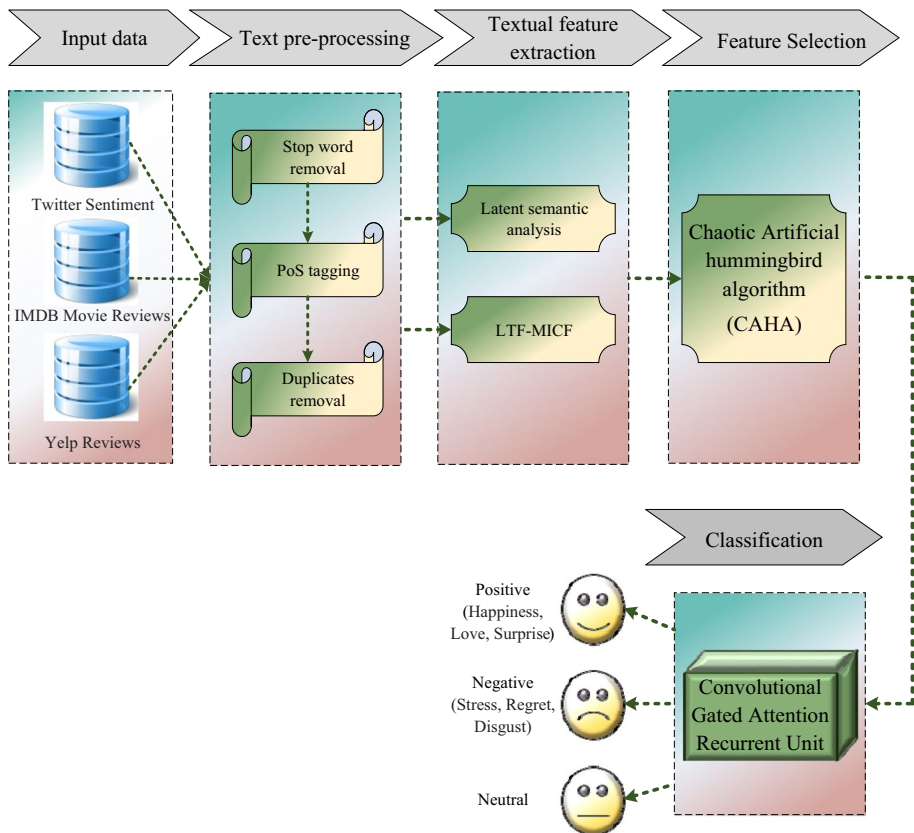


Fig. 2 Workflow of proposed sentiment and emotion classification model

### 3.1 Text pre-processing for quality enhancement

The presented input raw data contain several noises which may degrade the text's quality and reduce system performance. The proposed classification model initially performs the pre-processing stage to overcome this issue. This stage enhances the quality of text data and also aids in classifying the sentiments accurately. The proposed work utilizes the following stages for text pre-processing: stop words removal, POS tagging and duplicate removal.

**Stop words removal** The removal of stop words is a crucial process employed in the pre-processing stage. This process intends to remove the commonly available words across the entire corpus of documents. The articles and pronouns in the provided data are categorized as stop words. The contents which mainly used words are prepositions, conjunctions and determiners. The mentioned words are included in the stop word list, and these words will be eliminated from the text data. An example of stop word removal is shown in Table 2.

Thus, the reduced level information is neglected by eliminating the available stop words from the given text data, and the essential information is focused deeply. This process helps the system to attain higher performance. After removing the stop words, POS tagging is performed in this pre-processing stage.

**POS tagging** POS tagging is the major part of the pre-processing stage and is a famous process in NLP. A text's words are classified with a certain part of speech based on the word's definition and context. The word class to the word's position in the provided sentence is identified in POS tagging. Generally, the POS tagger allocates grammatical information for all words in the sentence. An example of POS tagger are CC, EX, NNP, CD, PRP\$, MD, etc.; POS tagging find out each word as a "noun", "adjective", "adverb" or "verb" and allocates a POS tag to all words.

**Duplicate removal** This process reduces the retrieval of duplicate information. Removing duplicated text is necessary because it creates several issues during classification. The duplicated text will affect the system hence removing the duplicated text makes the proposed model to obtain enhanced results.

After the completion of pre-processing, the quality of text data is improved, easing the classification process. These pre-processed data are moved into the feature extraction stage for further processing.

**Table 2** Removal of stop words: an example

| <i>Text with stop words</i>         | <i>After the removal of stop words</i> |
|-------------------------------------|--|
| I like to play cricket              | "Like", "Play", "Cricket"              |
| This product is so nice             | "Product", "Nice"                      |
| How to download a image from Google | "Download", "Image", "Google"          |



### 3.2 Textual feature extraction

Feature extraction is converting raw text data into numerical features that can be processed while securing the information in the actual data. The proposed feature extraction stage extracts the essential features from the pre-processed data. The features which provide more information for sentiment-based emotion classification should be extracted. This feature extraction stage has a major role in improving classification accuracy. Here, the proposed work chooses LSA and LTF-MICF for texture feature extraction.

#### 3.2.1 Latent Semantic Analysis (LSA)

LSA is also termed as Latent Semantic Index (LSI). LSA utilizes the Bag of Words (BoW) algorithm, which affords a term-document matrix. In this, rows are mentioned as terms and columns are mentioned as documents. LSA learns the latent or hidden topics by enabling a matrix decomposition of the document term matrix through Singular Value Decomposition (SVD). The detailed study of LSA is described as follows,

- Let's consider  $p$  documents and  $m$  words in the given dataset and design a  $p \times m$  matrix  $X$  in which all row mentions a document and all columns mentions a word.
- All entries can be determined by a raw count of the amount of times, the  $i^{\text{th}}$  word presented in the  $j^{\text{th}}$  document.
- The LSA models replace the raw data in the document term matrix with a Term Frequency Inverse Document Frequency (TF-IDF) score. A weight is allocated for each term  $i$  in document  $j$  through TF-IDF. The computation of TF-IDF is provided as,

$$\delta_{j,i} = tf_{j,i} \times \log \frac{M}{df_i} \quad (1)$$

Where,  $\delta_{j,i}$  is represented as TF-IDF score,  $tf_{j,i}$  is mentioned as the number of times a term is encountered in a document,  $M$  specifies the overall documents and the documents that present words is denoted as  $df_i$ . The term contains a higher weight when it appears continuously across the document but rarely across the corpus. The document term matrix  $X$  is very large, noisy and sparse across various dimensions. The dimensionality minimization on a matrix  $X$  is enabled through SVD, in which some latent topics that gather the relation between the documents and words are determined. SVD is a method in linear algebra that changes the matrix  $P$  into the product of three individual matrices  $P = U^*V^*W$ . Here, the singular values of  $P$ 's diagonal matrix are specified as  $V$ . The restricted SVD minimizes dimensionality by choosing only the  $l$  highest singular values and only maintaining the initial  $l$  columns of  $U$  and  $W$ . Now,  $l$  is a chosen hyperparameter and regulated to reflect the amount of topics that need to be identified.

$$X \approx U_l V_l W_l^T \quad (2)$$

In this, the document topic matrix is exhibited by  $\in H^{\wedge}(p \times l)$ , and the term topic matrix is  $W \in H^{\wedge}(m \times l)$ . The columns related to one of the  $l$  topics in both  $U$  and  $W$ . The row mentions document vector in  $U$  and is defined in terms of topics. On the other hand, the rows mentioned term vectors defined as topics in  $W$ . Through the term vectors and document vectors, one can effortlessly evaluate cosine similarity to compute the similarity of varied documents, the relationship of varied words and the relationship among terms and documents. In the

proposed feature extraction process, the TF-IDF vectorizer is used to refine the word counts and penalize the words that arise various times across the document. Depending on the scores in TF-IDF, the largest scored words are extracted to enhance the classification performance.

### 3.2.2 Log Term Frequency-based Modified Inverse Class Frequency (LTF-MICF) model

The feature extraction method, named LTF-MICF, combines two weighting schemes. The term frequency  $TF$  is the computation of how often a term is presented within a review document's contents. However, the continuous occurrence of terms in a document leads to additional weight, and the computation of TF alone becomes insufficient. The supervised methods have increased attention for generating usage of the review class information. Hence, the proposed work combines  $TF$  with MICF. The inverse class frequency  $CF$  is an inverse ratio of the overall classes wherein the term presented on training reviews to the entire classes. In the beginning, LTF centred is computed, which measures  $TF$  of each term in the pre-processed data. Then, perform log normalization on the  $TF$  data output represented as LTF. Next, the MICF is evaluated for all terms and is considered the advanced version  $CF$ . For all terms, the disparate class-specific scores should have varied importance to the entire term score; subsequently, the  $CF$  modification is enabled. Therefore, varied weights for varied class particular scores should be strictly allocated, and the weighted sum across each particular class score is used for the entire term score. The equation of LTF-MICF is represented as,

$$LTF - MICF(t_q) = LTF(t_q) * \sum_{v=1}^m w_{qv} \cdot [{}^i c_h(t_q)] \quad (3)$$

where,  $w_{qv}$  represents a particular weighting factor of the term  $t_q$  for class  $c_v$  and can be mentioned as,

$$w_{qv} = \log \left( 1 + \frac{k_i t'}{\max(1, k_i t')} \cdot \frac{k_i \tilde{t}}{\max(1, k_i, \tilde{t})} \right) \quad (4)$$

The weighting factor  $W_f$  is the technique utilized to consider the weight of a provided input text data. The total amount of data in class  $c_v$  is denoted as  $k_i$ , and it includes the term  $t_q$ ; an amount of  $k_i$  in another class that includes the term  $t_n$  is represented as  $k_i t'$ ,  $k_i \hat{t}$  specifies the amount of  $k_i$  in-class  $c_v$ , which cannot include the term  $t_q$ . The amount of  $k_i$  which does not include the term  $t_q$  in other classes is specified as  $k_i \tilde{t}$ . To neglect the negative weights, fixed '1' is employed in the proposed LTF-MICF approach. When  $k_i t' = 0$  or  $K_i \hat{t} = 0$ , the lowest denominator is fixed to '1' for diminishing the zero-denominator issue in a critical case. An advanced term weighting LTF-MICF ( $t_q$ ) is generated centred on the MICF ( $t_q$ ). The equation for  $LTF(t_q)$  and  ${}^i c_h(t_q)$  can be represented as,

$$LTF(t_q) = \log(1 + TF(t_q, k_i)) \quad (5)$$

where,  $TF(t_q, k_i)$  mentions the raw number of a term  $t_q$  on a provided data. The number of times that  $t_q$  is encountered on the input data is provided as,

$${}^i c_h(t_q) = \log \left( 1 + \frac{v}{c(t_q)} \right) \quad (6)$$

where,  ${}^i c_h(t_q)$  indicates the total number of times  $t_q$  is present on the input data,  $v$  mentions the overall classes in a provided dataset and the amount of classes presenting the term  $t_q$  is specified as  $c(t_q)$ . The dataset features are mentioned as  $H_i = \{H_1, H_2, \dots, H_3, \dots, H_m\}$ . The extracted features from the pre-processed data are considered as  $H_1, H_2, \dots, H_3, \dots, H_m$ . The algorithm of a proposed LSA with LTF-MICF is illustrated in Table 3.

The most discriminative features are extracted in the proposed feature extraction step. Since all the discriminative features are extracted, it may result in dimensionality issues in the classification stage. This may affect the system’s performance by increasing its complexity. Thus, selecting useful features from the extracted features is necessary to improve classification accuracy.

**Table 3** Pseudocode of proposed LTF-MICF-based feature extraction

|   |
|---|
| <p><b>Algorithm: Proposed textual feature extraction</b></p> <p><b>Begin</b><br/>                 Input: Pre-processed data;<br/>                 Perform LSA;<br/>                 Construct <math>p \times m</math> matrix;<br/> <b>for</b> matrix <math>X</math><br/>                     Row <math>\rightarrow</math> Documents;<br/>                     Columns <math>\rightarrow</math> word;<br/>                 Replace raw count of data into the LSA model with the score of TF-IDF;<br/> <b>for</b> each term <math>i</math><br/>                     Allocate weight using TF-IDF;<br/>                 Compute TF-IDF through <math>\delta_{j,i} = tf_{j,i} \times \log \frac{M}{df_i}</math>;<br/>                 Perform dimensionality minimization using SVD;<br/>                 Extract features according to the highest score words;<br/>                 Perform LTF-MICF;<br/> <b>for</b> every term <math>t_q (q = 1, 2, \dots, m)</math> and every class <math>c_v (v = 1, 2, \dots, m)</math> <b>do</b><br/>                     Evaluate MICF <math>(t_q), w_{qv} \cdot [{}^i c_h(t_q)]</math><br/>                     Evaluate <math>w_{qv}</math> and <math>{}^i c_h(t_q)</math> through equations (4) and (6);<br/> <b>end for</b><br/> <b>for</b> every term <math>t_q (q = 1, 2, \dots, m)</math> <b>do</b><br/>                     Evaluate <math>LTF(t_q)</math> utilizing equation (5)<br/>                     Evaluate <math>LTF - MICF(t_q)</math> through equation (3)<br/> <b>end for</b><br/>                 return<br/> <b>end</b></p> |
|---|

### 3.3 Chaotic Artificial hummingbird algorithm (CAHA) based feature selection

The feature selection process is essential to choose the optimal feature set from the extracted features. This stage selects the most needed features that afford higher classification performance. The preceding stage contains more features that can make higher computational complexity. It may affect the efficiency of a proposed classification. Thus, the proposed study attempts to select the optimal feature set using the CAHA approach. The Artificial Hummingbird Algorithm (AHA) approach is a bio-inspired optimization method mainly utilized to resolve optimization issues. This approach simulates certain flight skills and strategies regarding the intelligent foraging of hummingbirds. The foraging strategies of hummingbirds contain three types of flight skills such as axial, diagonal and omnidirectional. Additionally, the hummingbird used the strategies like territorial foraging, guided foraging and migrating foraging. Using the strategies of the AHA approach, the proposed study selects the optimal features according to the fitness function ( $F_f$ ). The fitness function is given as,

$$(F_f) = \beta * \psi + (1 - \beta) * \frac{|N_e|}{N_d} \quad (7)$$

where,  $\beta$  signifies the parameter that degrades the results of a classification stage, the classification error is specified as  $\psi$ , the total number of features obtained in the feature extraction stage is denoted as  $N_e$  and the total features in the given dataset is represented as  $N_d$ . In the beginning, the search agents are randomly initialized as,

$$z_i = L_B + g \cdot (U_B - L_B) \quad i = 1, \dots, m \quad (8)$$

where,  $L_B$  and  $U_B$  represents the lower and upper boundaries for a  $l$ -dimensional issue. The random vector ranges from 0 to 1 is specified as  $g$ , and the position of  $i^{th}$  feature as a resolution of the mentioned issue is denoted as  $z_i$ .

**Initialization** The initialization of features is represented as,

$$I_F = \begin{cases} 0 & \text{if } i \neq j \\ \text{null} & \text{if } i = j \end{cases} \quad i = 1, \dots, m; j = 1, \dots, m \quad (9)$$

where,  $I_F = \text{null}$  mentions a search agent is choosing a feature as its feature source and  $I_F = 0$  represents the  $j^{th}$  feature source visited by the  $i^{th}$  search agent in the present iteration.

**Guided foraging** Each search agent tends to attain a feature source with a high nectar volume. It specifies that an optimal feature requires an increased nectar-refilling rate and a high unvisited time by that search agent. In the AHA approach, a search agent tries to analyze the features with the largest visit level for guided foraging. In this, the search agent selects the features with the largest nectar refilling rate from them as its optimal feature source. After attaining the optimal features, the search agent can explore them to select features. Three flight skills like diagonal, omnidirectional and axial flights, are utilized and modelled in the searching process using a direction switch vector. The above-mentioned vector is employed to limit the several directions in the presented  $l$ -dimension. The pattern of flights can be enlarged to a  $l$  dimensional space, and the formulation of axial flight is represented as,

$$L^{(i)} = \begin{cases} 1 & \text{if } i = \text{randi}([1, l]) \\ 0 & \text{else} \end{cases} \quad i = 1, \dots, l \quad (10)$$

The computation of diagonal flight is represented as,

$$L^{(i)} = \begin{cases} 1 & \text{if } i = K(j), j \in [1, k], K = \text{rand perm}(f), f \in [2, [g_1 \cdot (l - 2)] + 1] \\ 0 & \text{else} \end{cases} \quad i = 1, \dots, l \quad (11)$$

The computation of omnidirectional flight is shown in below equation,

$$L^{(i)} = 1 \quad i = 1, \dots, l \quad (12)$$

where,  $\text{randi}([1, l])$  mentions a random integer ranges from 1 to  $l$ , a random permutation of integer's ranges from 1 to  $f$  is specified as  $\text{rand perm}(f)$ , and the random number ranges between 0 and 1 is denoted as  $g_1$ . The mathematical formulation of a guided foraging strategy of search agent and feature source is provided as,

$$u_i(t + 1) = z_{i, \text{opt}}(t) + \delta \cdot L \cdot (z_i(t) - z_{i, \text{opt}}(t)) \quad (13)$$

$$\delta \sim M(0, 1) \quad (14)$$

where,  $z_i(t)$  mentions the location of  $i^{\text{th}}$  feature source at a time  $t$ , the location of optimal features attained by the search agent is denoted as  $z_{i, \text{opt}}(t)$  and a guided factor is specified as  $\delta$  which is a normal distribution  $M(0, 1)$  with a mean of 0 and standard deviation of 1. The above Eq. (13) represents all present feature sets to update their location in the neighbourhood of the optimal feature source, and the search agent's guided foraging strategy is modelled through varied flight patterns. The equation of position updation of  $i^{\text{th}}$  feature source is mentioned as,

$$z_i(t + 1) = \begin{cases} z_i(t) & h(z_i(t)) \leq h(u_i(t + 1)) \\ u_i(t + 1) & h(z_i(t)) > h(u_i(t + 1)) \end{cases} \quad (15)$$

where,  $h(\cdot)$  mentions the function of fitness value. The above Eq. (15) exhibits that if the nectar refilling rate of a feature source is superior to the present features, the search agent leaves the present feature sets and remains at the candidate resulting from Eq. (13) for exploring. The visit table is an essential component in the AHA approach which preserves the visit information of feature sources. Based on the visit table, any search agent can identify the optimal feature set to visit on each iteration. The visit table details how long the search agent has not visited the feature set since it was visited by the same search agent. A large unvisited time indicates a large visit level. Through Eq. (13), all search agents visit their optimal feature sets. According to the optimal feature set, the search agent enables the guided foraging behaviour in each iteration using Eq. (13). The visit levels of other feature sets for this search agent are maximized by one, and the visit level of its optimal feature set visited is initialized to zero. After guided foraging, the search agent will not change its feature set if there is no suitable nectar-refilling rate. Then, the present source will be alternated by a new one if there is an appropriate nectar-refilling rate, and then this search agent will remain at the new feature sets. The visit level updation is set as the maximum level of another feature set enhanced by one. For each iteration, the visit table is updated in the AHA approach.

**Territorial foraging** After attaining the optimal features, the search agent explores the new feature sets instead of visiting other preceding features. Hence, a search agent can move its location to a neighbouring region within its territory, where a feature may be identified as a candidate solution. It may be superior to the present one, and the mathematical formulation for local exploration of search agents in the territorial foraging behaviour and a candidate feature set is attained as,

$$u_i(t + 1) = z_i(t) + \eta.L.z_i(t) \tag{16}$$

$$\eta \sim M(0, 1) \tag{17}$$

Where,  $\eta$  mentions a territorial factor, and it is the normal distribution  $M(0, 1)$  with a mean of zero and a standard deviation of one. The above Eq. (16) can permit the search agent to effortlessly identify the feature sets in this local neighbourhood based on its own location through certain flight skills.

**Migration foraging** When a position chosen by the search agent includes a small number of features, this search agent moves to a high distant feature set for exploration. The coefficient of migration is defined in the AHA approach. When the pre-decided value of the migration coefficient is enhanced, the search agent with minimal nectar refilling rate will migrate to a new feature set. This new feature set is generated randomly in the exploration space. In this case, the search agent will leave the preceding source and remains at the new feature source for exploration. Then, the visit table is updated by the AHA approach. The equation of migration behaviour is represented as,

$$z_{wor}(t + 1) = L_B + g.(U_B - L_B) \tag{18}$$

where,  $z_{wor}$  is specified as the feature source with the worst rate of nectar refilling. When the guided and territorial foraging steps result in the worst nectar refilling rate, the search agent visits each feature set based on the visit table at the end of each iteration. The migration coefficient, which corresponds to the population size, is given as,

$$N = 2m \tag{19}$$

### 3.3.1 Chaotic maps

Generally, several chaotic maps are available and are used to generate chaotic sets. To enhance the convergence rate of the AHA approach, the proposed study used chaotic maps that can enhance the overall efficiency of the algorithm. The proposed CAHA approach is established by applying chaos in the AHA mechanism. Because of the dynamic strategy of AHA, chaotic maps have been employed in the optimization field, which assists in searching the exploration space more globally and vigorously. The proposed work chooses an iterative map-based chaos to enhance AHA’s convergence rate. It can be given as,

$$z_i = L_B + g.(U_B - L_B) \times y_{q+1} \quad i = 1, \dots, m \tag{20}$$

$$\text{Where, } y_{q+1} = \text{abs}\left(\sin\left[\frac{S}{z_q}\right]\right) \tag{21}$$

where,  $S$  mentions the adjustable parameter. Thus, the proposed CAHA-based feature selection process selects the most optimal sentiment and emotion classification features. This stage diminishes the overall system's computational complexity and helps the classifier to obtain improved results. The Pseudocode of a proposed CAHA approach is illustrated in Table 4.

### 3.4 Dual-stage Deep CGARU model for Sentiment-based Emotion Classification

The chosen features from the proposed CAHA approach are utilized as input in the final classification stage. For classification purposes, the proposed study used a dual-stage deep CGARU model. The modules involved in the CGARU model are the channel-wise attention module, convolutional module, Gated Recurrent Unit (GRU) module and self-attention module. An attention mechanism is utilized initially to find the channels with appropriate information for classification. Then, 3 one-dimensional CNN layers and four GRU layers were utilized to obtain temporal information. Later, a self-attention mechanism encodes the temporal dependency extracted from the preceding layers, like one-dimensional CNN and GRU layers. At last, the classification, fully connected, and softmax layers were employed to classify the sentiments and emotions from the provided text data. Table 5 provides the hyperparameter settings of a proposed CGARU method. Figure 3 depicts the architecture of a proposed CGARU model.

#### 3.4.1 Channel-wise attention module

This module is utilized to attain suitable features by allocating varied weights to all channels in which each channel's significance is analyzed. Initially, average pooling was enabled in each step using the pre-processed data. Then, it obtains the attention matrix with fully connected layers and a non-linear activation function via a channel-wise vector mentioning a mean value over time. It is given as,

$$X_{att} = \text{soft max} \left( \omega_0 \cdot \left( \frac{1}{m} \sum_1^m Z \right)^T + a_0 \right) \quad (22)$$

In Eq. (21), an average pooling of epochs  $Z \in F^{m \times b}$  is employed as an input, and the final result in an attention mechanism is specified as the attention matrix  $X_{att} \in F^{1 \times b}$ . Where,  $m$  is considered as the amount of time steps and  $b$  mentions the amount of channels. During training, the bias and weight matrix ( $a_0 \in F^{1 \times b}$  and  $\omega_0 \in F^{b \times b}$ ) in the fully connected layer is updated. The softmax layer is employed for the activation function, and the equation is,

$$\text{soft max}(s_i) = \frac{\exp(s_i)}{\sum_{j=1}^B \exp(s_j)} \quad (23)$$

where,  $s = [s_1, s_2, \dots, s_b]$  is considered as the fully connected layer. Then, the output feature  $Y_{att} \in F^{m \times b}$  obtained from the channel-wise attention mechanism is attained by increasing the actual data  $Z$  and attention matrix  $X_{att}$ , which is mentioned as,

$$Z_{att} = Z \otimes X_{att} \quad (24)$$

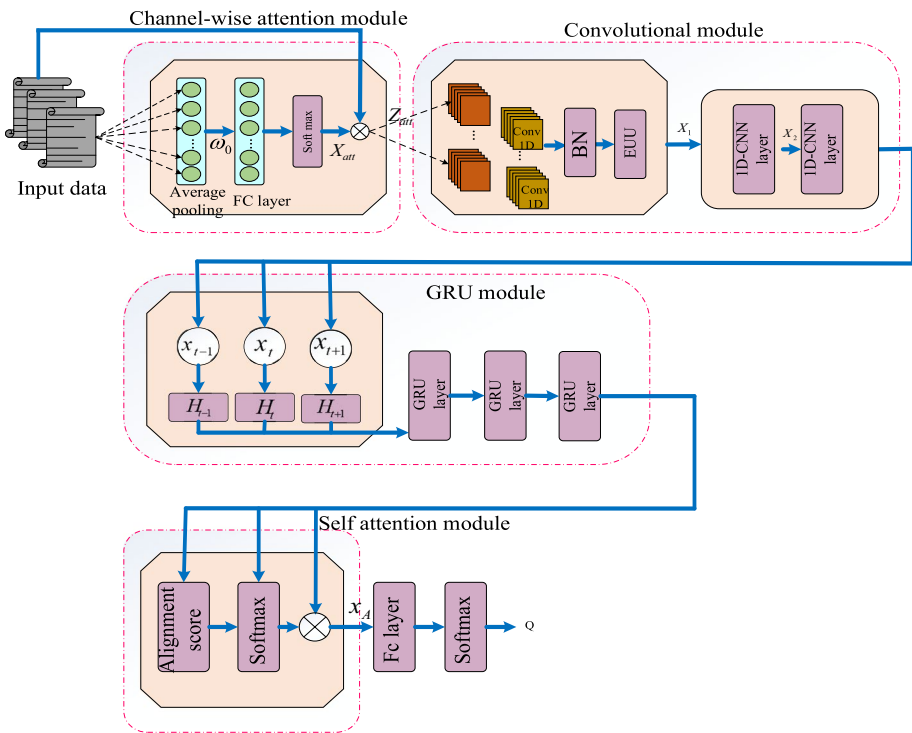
**Table 4** Proposed CAHA algorithm

| <b>CAHA: Algorithm</b>  |
|---|
| <p><i>Begin</i></p> <p>Initialize the parameters;<br/> Initialize the search agent;<br/> Initialize the features;<br/> Evaluate fitness function using equation (7);<br/> Perform guided foraging strategy;</p> <p><b>for</b> <math>i^{th}</math> feature from 1 to <math>m</math><br/> Execute equation (12)<br/> If <math>h(u_i(t+1)) &lt; f = h(z_i(t))</math><br/> <math>z_i(t+1) = u_i(t+1)</math></p> <p><b>for</b> <math>j^{th}</math> feature set from 1 to <math>m</math> (<math>j \neq opt, i</math>)<br/> Visit_table (<math>i, j</math>) = Visit_table (<math>i, j</math>) + 1<br/> <b>end</b><br/> Visit_table (<math>i, opt</math>) = 0</p> <p><b>for</b> <math>j^{th}</math> feature set from 1 to <math>m</math><br/> Visit_table (<math>j, i</math>) = <math>\max_{c \in m \text{ and } c \neq j} (\text{Visit\_table}(j, c)) + 1</math></p> <p><b>end</b><br/> <b>else</b><br/> <b>for</b> <math>j^{th}</math> feature set from 1 to <math>m</math> (<math>j \neq opt, i</math>)<br/> Visit_table (<math>i, j</math>) = Visit_table (<math>i, j</math>) + 1<br/> <b>end</b><br/> Visit_table (<math>i, opt</math>) = 0<br/> <b>end</b><br/> <b>end</b></p> <p><i>Perform territorial foraging strategy;</i></p> <p><b>for</b> <math>i^{th}</math> searching agent from 1 to <math>m</math><br/> <b>if</b> <math>h(z_i(t+1)) &lt; h(z_i(t))</math><br/> <math>z_i(t+1) = u_i(t+1)</math></p> <p><b>for</b> <math>j^{th}</math> feature set from 1 to <math>m</math> (<math>j \neq i</math>)<br/> Visit_table (<math>i, j</math>) = Visit_table (<math>i, j</math>) + 1<br/> <b>end</b></p> <p><b>for</b> <math>j^{th}</math> feature set from 1 to <math>m</math><br/> Visit_table (<math>j, i</math>) = <math>\max_{c \in m \text{ and } c \neq j} (\text{Visit\_table}(j, c)) + 1</math></p> <p><b>end</b><br/> <b>else</b><br/> <i>Perform migration foraging behaviour;</i><br/> <b>If</b> <math>\text{mod}(t, 2m) == 0</math><br/> Execute equation (17)</p> <p><b>for</b> <math>j^{th}</math> feature set from 1 to <math>m</math> (<math>j \neq wor</math>)<br/> Visit_table (<math>wor, j</math>) = Visit_table (<math>wor, j</math>) + 1<br/> <b>end</b></p> <p><b>for</b> <math>j^{th}</math> feature set from 1 to <math>m</math><br/> Visit_table (<math>j, wor</math>) = <math>\max_{c \in m \text{ and } c \neq j} (\text{Visit\_table}(j, c)) + 1</math></p> <p><b>Do</b> iterative map-based chaotic using equation (20);<br/> <b>end</b><br/> <b>end</b><br/> <i>stop</i></p> |



**Table 5** Hyperparameters of the CGARU model

|                                     |          |
|-------------------------------------|----------|
| Learning algorithm                  | Adam     |
| Initial learning rate               | 1e-3     |
| Mini batch size                     | 32       |
| Max epochs                          | 100      |
| Input size                          | 8000,336 |
| Number of neurons in a hidden layer | 8000     |
| Hidden layer                        | 3        |
| The number of an output layer       | 1        |



**Fig. 3** Structure of the proposed dual-stage CGARU model

The represented symbol  $\otimes$  signifies element-wise multiplication, and it operates utilizing the product of related elements of dual matrices when both matrices hold similar dimensions.

### 3.4.2 Convolutional neural network module

Two-dimensional CNNs are utilized in computer vision due to the capability of adaptive Learning of spatial features from the text data. The proposed study prefers three 1D CNNs to extract temporal features from recalibrated input data whose informative channels were

identified through a channel-wise attention mechanism. The initial convolutional layers enabled the process of convolution at the time step  $m$ . When the convolutional operations are enabled with a kernel size of  $p$ , the amount of output filters of  $g$  and stride length of  $l$  by utilizing the similar padding to manage the input size, the output  $X_{\text{conv}} \in F^{m+2k-p/l \times g}$  can be computed as,

$$X_{\text{conv}} = \cup_{i=1}^g ELU(BN(\omega_i * Z_{\text{att}} + a_i)) \quad (25)$$

where,  $\omega_i \in F^{p \times b}$  and  $a_i \in F^{(m+2k-p)/l \times 1}$  mentions the convolutional kernel of filter  $i$  and bias correspondingly. The represented symbol  $*$  defines convolution operation with the batch normalization function and temporal dimension, which is employed for stability training. The equation of batch normalization is given as,

$$B_N(y) = y_{B_N} = \psi \frac{y - \mu_A}{\sqrt{\sigma_A^2 + \lambda}} + \eta \quad (26)$$

where,  $\mu_A$  and  $\sigma_A$  mentions the mean and standard deviation of every batch correspondingly, the hyperparameters are denoted as  $\psi$  and  $\eta$ . The term  $ELU$  specifies an exponential linear unit and was utilized as an activation function. With the assistance of concatenation, the output  $X_{\text{conv}}$  is attained and is signified by the symbol  $\cup$  of the outcome from each filter. The resting two convolutional layers effectively enabled a similar operation according to the above Eqs. (24) and (25), with the same parameters as the initial convolutional layers.

### 3.4.3 Gated recurrent unit (GRU) module

Using a gating concept, GRU is an advanced version of the conventional RNN method for resolving long-term dependency issues. This module is employed to further acquire temporal features from the resultant samples of a convolutional module. The GRU employs a reset gate and an update to obtain one hidden state, which is faster and highly simpler than the traditional LSTM method that utilizes an input, output and forget gate to attain a cell and hidden state. The update gate  $Ug_t \in F^{1 \times mh}$  combines the input and forget gate's function in the LSTM method, analyses the update ratio of previous and current information as mentioned in the below equation,

$$Ug_t = \text{sigmoid}(h_{t-1}\omega_{uh} + x_t\omega_{ux} + a_u) \quad (27)$$

where,  $h_{t-1}\omega_{uh}$  represents the hidden state and  $x_t \in F^{1 \times g}$  signifies the present input vector. The weight matrices like  $\omega_{uy} \in F^{g \times mh}$  and  $\omega_{uh} \in F^{mh \times mh}$  and the bias  $a_u \in F^{1 \times mh}$  represent the update gate which is set with the amount of hidden units  $mh$ . In the initial GRU layer, the input vector  $x_t$  denotes the present output of a convolutional layer  $X_{\text{conv}}$ . Hence,  $g$  represents the amount of output filters presented in CNN. Such values were consistent across each GRU layer. The reset gate decides whether to discard or retain the preceding information, and it can be attained as,

$$f_t = \text{sigmoid}(h_{t-1}\omega_{fh} + x_t\omega_{fx} + a_f) \quad (28)$$

where,  $\omega_{fx} \in F^{g \times mh}$  and  $\omega_{fh} \in F^{mh \times mh}$  are considered as the weight matrices and  $a_f \in F^{1 \times mh}$  is represented as bias. The sigmoid function was utilized for the activation function in reset and update gates. Next, by enabling element-wise multiplication, the candidate hidden state  $H_t$  can be attained among the preceding hidden states and reset gates. This is given in below equation,

$$H_t = \tanh(H_{t-1}\omega_{HH} \otimes f_t + x_t\omega_{xh}) \quad (29)$$

where,  $\omega_{HH} \in F^{mh \times mh}$  and  $\omega_{xh} \in F^{g \times mh}$  mentions the weight matrix and  $\tanh$  signifies the hyperbolic tangent function. Hence, the preceding hidden state is totally concealed if  $f_t = 0$ . Then, the new hidden state  $H_t$  can be computed along with the update gate  $Ug_t$  in which element-wise multiplication of the preceding hidden state  $H_{t-1}$  and present candidate state  $H_t$  is enabled. The computation of new hidden states is provided as,

$$H_t = (1 - Ug_t) \otimes H_{t-1} + Ug_t \otimes H_t \quad (30)$$

In this, the four GRU layers extract the temporal features of CNN layers. All GRU layers include similar input and output dimensions; as well as every four layers distribute similar parameter sets.

### 3.4.4 Self-attention module

The extracted temporal features from both CNN and GRU are time-varying. Therefore, the self-attention module differentiates the importance of all features and provides appropriate prediction. This self-attention concept was initially established in machine translation to enhance long-range dependencies. The proposed study uses multiplicative attention, termed as Luong attention, to compute the alignment function  $A_t$  and is represented as,

$$A_t = \frac{\exp(H_t^T H_l)}{\sum_l \exp(H_t^T H_l)} \quad (31)$$

where,  $H_t$  and  $H_l$  mentions query and key correspondingly. The proposed classification stage sets the query and key as the output state of a previous GRU layer. Next, the new feature set  $X_A$  mentions the feature set  $H_t$  context, which can be mentioned utilizing the score of attentive alignment  $A_t$ . It is given as,

$$X_A = \sum_t A_t H_t \quad (32)$$

In the end, a fully connected layer and a softmax function enabled the classification of sentiments and emotions. The computation of a classifier's probability is represented as,

$$Q = \text{soft max}(\omega X_A + \varepsilon) \quad (33)$$

Where,  $\omega$  represents the weight and  $\varepsilon$  mentions bias correspondingly. The model is trained by diminishing the cross entropy loss among the estimated probability and ground truth label. The equation of binary cross entropy loss function is determined as,

$$Loss = - \sum_t (X_t \log(Q_t) + (1 - X_t) \log(1 - Q_t)) \quad (34)$$

where,  $X_t$  and  $Q_t$  defines the target label and estimated probability at  $t^{\text{th}}$  time correspondingly. The presented loss function is reduced by using the Adam optimizer, making the classification stage more robust. The attention layer provides varied attention weights for each data due to the different samples. Hence the attention weight for each input text data can be employed to minimize the amount of channels. Thus, the proposed CGARU classifier effectively classifies the sentiments and emotions from the given input. In CGARU, the convolutional module initially classifies positive negative and neutral sentiments. Also, the

emotions behind the sentiments are categorized in the proposed GARU module. The working process of a proposed framework is presented in Algorithm 3.

## 4 Simulation results and analysis

This section describes the simulation results and discussions of the proposed dual-stage deep learning model. The simulation is performed in the Python platform to illustrate the robustness of the proposed techniques. The datasets are included, followed by the training approach, and hyperparameters are employed. The performance of a proposed study is

|  |
|--|
| <p><b>Input:</b> Raw text input from datasets<br/> <b>Output:</b> Classified emotions<br/> <i>//Pre-processing</i><br/> Step 1: The raw input text is evaluated and the stop words are removed<br/> Step 2: Allocate PoS tag to each word in the given input<br/> Step 3: After tagging, the duplicate text from the input are removed<br/> <i>//feature extraction</i><br/> Step 4: Using the input documents and words, a <math>p \times m</math> matrix is created where the rows indicate documents and columns indicate words<br/> Step 5: The entries in the matrix are checked to identify the amount of times, the <math>i^{th}</math> word is presented in <math>j^{th}</math> document<br/> Step 6: The TF-IDF score is computed for each term in the document and a weight value is provided<br/> Step 7: Using the SVD algorithm, dimensionality reduction is performed<br/> Step 8: Based on the TF-IDF scores, the largest scored words are extracted<br/> Step 9: Perform LTF-MICF<br/> Step 10: Evaluate LTF-MICF using equation (3)<br/> Step 11: Obtain the crucial high-dimensional features as output<br/> <i>//feature selection</i><br/> Step 12: Define the fitness function for the CAHA using equation (7)<br/> Step 13: Initialize the extracted features as the initial population of the algorithm using equation (9)<br/> Step 14: Perform guided foraging strategy as in equation (13)<br/> Step 15: Update the position of the population using equation (15)<br/> Step 16: Perform territorial foraging strategy using equation (16)<br/> Step 17: Perform migration foraging strategy using equation (18)<br/> Step 18: Follow iterative map based chaotic using equation (20)<br/> Step 19: Output the selected optimal features<br/> <i>//Classification</i><br/> Step 20: The optimal set of features reaches the input layer and channel-wise attention is performed to obtain the suitable features for classification<br/> Step 21: The resultant features are provided to the convolutional module for better Learning of spatial features from the text data<br/> Step 22: Then the features are provided to the GRU module to preserve the temporal features from the resultant samples of convolutional module<br/> Step 23: The time-varying temporal features are then forwarded to the self-attention module for feature discrimination<br/> Step 24: The random weight parameters of the network model are then optimized using the Adam optimizer<br/> Step 25: The sentiments and emotions are classified from the input using the convolutional and GARU modules</p> |
|--|

**Algorithm 3** Working principle of the proposed framework

measured by evaluating some of the metrics, and the results are compared with other deep learning techniques to determine the efficacy of a proposed model. Table 6 represents the system configuration that supports the proposed simulation.

#### 4.1 Dataset description

The proposed study utilized three different datasets to classify the sentiment-based emotion from the text. Thus, the study analyses the proposed classification model with the following datasets such as,

- Twitter Sentiment Dataset <https://www.kaggle.com/datasets/saurabhshahane/twitter-sentiment-dataset>
- IMDBMovieReviews <https://www.kaggle.com/code/lakshmi25npathi/sentiment-analysis-of-imdb-movie-reviews/notebook>
- YelpReviewsDataset <https://www.kaggle.com/code/omkarsabnis/sentiment-analysis-on-the-yelp-reviews-dataset>

The Twitter sentimental dataset is attained from the Kaggle repository, mainly used for sentiment classifications. This dataset involves an amount of 162981 Twitter data. The IMDB Movie Reviews dataset is also collected from the Kaggle repository, which holds 50001 data. The third dataset, YelpReviewsDataset, contains 10001 data which holds customer reviews with positive and negative comments.

#### 4.2 Performance metrics

The performance of a proposed dual-stage CGARU model is evaluated by measuring the performance metrics like accuracy, precision, recall, specificity, F-measure, specificity and kappa coefficient. The attained results from the proposed technique are compared with other conventional deep learning techniques like MLP (Multi-Layer Perceptron), LSTM (Long-Short Term Memory), Bi-LSTM (Bi-directional Long-Short Term Memory), Convolutional Neural Network (CNN) and DNN (Deep neural network). The comparison analysis exhibits the efficiency of a proposed model.

**Accuracy** Accuracy is an essential measure to determine the correctly classified data for the entire data in the given dataset. When the classifier provides a classification of sentiment, it is compared with the actual sentiment of a word and then the true positives and false positives are categorized. This metric has a major role in analyzing the performance of a proposed dual-stage CGARU-based classification. The term accuracy is evaluated as,

**Table 6** System configuration

|               |   |
|---------------|---|
| Processor     | Intel (R) Core™ i5-3570 CPU @ 3.40 GHz 3.40         |
| Installed RAM | 8.00 GB (7.89 GB usable)                            |
| Device ID     | 330431f3-8552-4664-BCAD-E0108D59137B                |
| Product ID    | 00330-80000-00000-AA440                             |
| System type   | 64-bit operating system, ×64-based processor        |
| Pen and touch | No pen or touch input is available for this display |

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (35)$$

**Precision** The precision metric also exhibits the proposed classifier's efficacy and determines the rate of perfectly classified sentiments from the total amount of true positives. It is formulated as the sum of accurately classified data in a specific class divided by the whole categorized data. The equation of precision measure is given as,

$$Precision = \frac{TP}{TP + FP} \quad (36)$$

**Recall** The recall metric is the ratio of an entire amount of true positives to the entire amount of both true and false negatives. This recall metric is also considered a sensitivity measure and can identify the error rate of classifier. A recall is mathematically represented as,

$$Recall = \frac{TP}{TP + FN} \quad (37)$$

**F1-score** It is the harmonic mean of precision and recall. In the f-score, appropriate precision and recall are computed by one. If the value of recall or precision is zero, then the f-score value is also assumed as zero. The computation of the F1-score is represented as,

$$F - measure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (38)$$

**Specificity** Specificity determines the proportion of real negatives that are classified as negative by the proposed model. The correctly classified sentiments and emotions are analyzed in the specificity measure. It is evaluated as,

$$Specificity = \frac{TN}{TN + FP} \quad (39)$$

**Kappa coefficient** Kappa defines the classifier's performance rate, i.e. it measures how much better the classifier is accurately classifying the output. This kappa measure provides the inter and intra-rater reliability of classified samples.

$$K = \frac{P_{observed} - P_{chance}}{1 - P_{chance}} \quad (40)$$

where,  $TP$  mentions true positives,  $TN$  is considered as true negatives,  $FN$  signifies false negatives,  $FP$  signifies false positives and  $P$  is the probability of an event.

### 4.3 Performance evaluation of the proposed dual-stage deep CGARU model

This section presents the performance evaluation results of a proposed sentiment and emotion classification model. The efficiency of a proposed study is proved by enabling the comparative analysis using different deep learning techniques like MLP, LSTM, Bi-LSTM, CNN and DNN. The performance of a proposed model is determined in three datasets, and the attained accuracy and loss of three datasets are shown in Fig. 4.

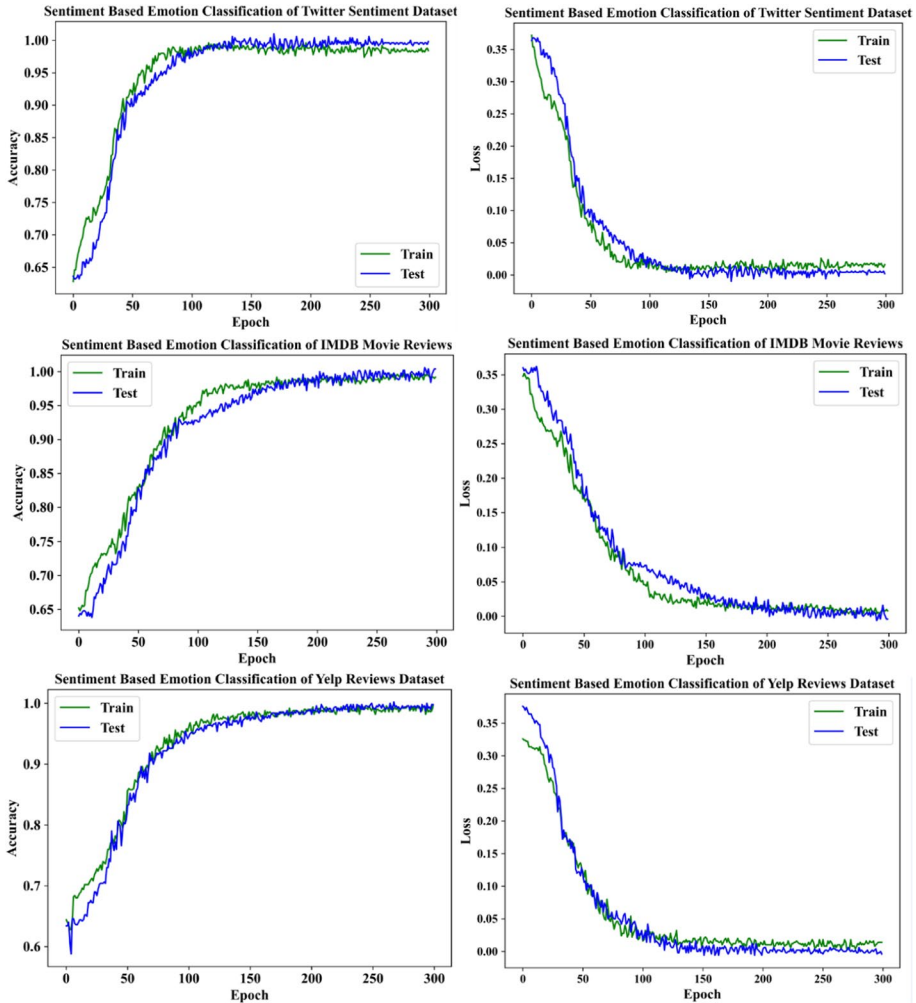


Fig. 4 Accuracy and loss comparison with different datasets

The obtained accuracy and loss in the proposed classification stage are analyzed by varying the size of epochs. The accuracy and loss are determined using three datasets during training and testing. In the Twitter sentiment dataset, the accuracy is minimal at the epoch of 0 to 40 compared to the training stage. When the epoch size is varied to 150, the accuracy in testing is increased. Similarly, the loss is reduced at the epoch of 150 using the Twitter sentiment dataset. The accuracy remains the same for training and testing at the epoch of 150 to 200 in the IMDB Movie Reviews dataset. After the epoch of 200, the accuracy of a proposed classifier is improved in the testing phase. On the other hand, the loss remains the same at the epoch of 200 to 300. In the Yelp Reviews dataset, almost the accuracy attained in the training and testing phase is similar. However, the loss is minimized in the testing phase at the 150<sup>th</sup> epoch. This states that the proposed method has attained better performance for classifying the sentiments and emotions from the provided data. Figure 5 represents the confusion matrix for three datasets.

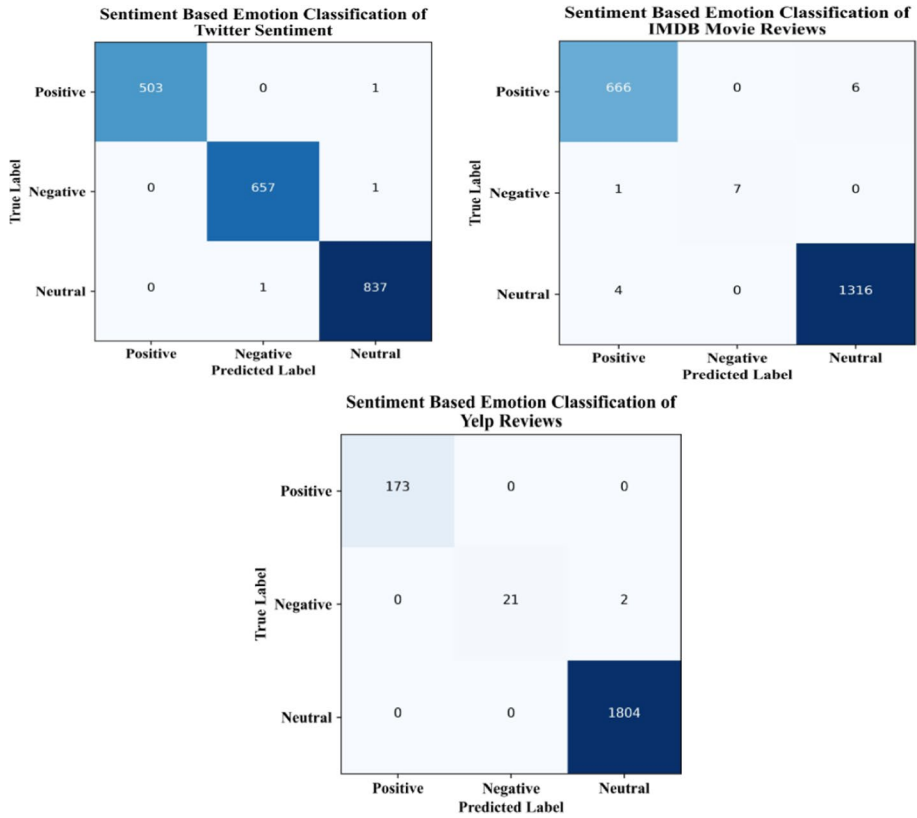


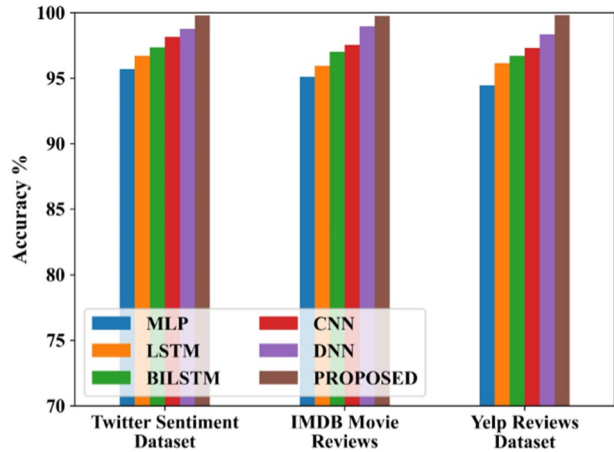
Fig. 5 Confusion matrix for different dataset

The confusion matrix shows the robustness of a proposed classifier. Using the Twitter sentiment dataset, only a few data are misclassified as other classes. In the positive class, 503 data are correctly classified as positive, and only one class is misclassified. Similarly, only one data is misclassified for the negative and neutral classes. Using the IMDB Movie Reviews dataset, 666 data are accurately classified as positive, and 6 are misclassified. For the negative class, only one data is misclassified from the total of 8 data. For the neutral class, 1316 are correctly classified, and the remaining 4 data are wrongly predicted. Also, using the Yelp Reviews dataset, 173 positive sentiment data are accurately classified as positive and only a few data are misclassified from negative and neutral classes. Thus, it states that the classifier has higher efficiency in categorizing sentiments from each dataset. The accuracy comparison of a proposed and other deep learning techniques is shown in Fig. 6.

The above graphical representation provides the accuracy of both proposed and existing techniques. Using three datasets, the accuracy measure is determined for each technique. The Figure states that the accuracy of a proposed model is better than the existing techniques. The existing techniques cannot accurately classify sentiment-based emotion because of several issues. The existing MLP provides the lowest accuracy as compared with other techniques. During the classification process, the extending parameters in the MLP approach make the classification accuracy a minimum. Similarly, the LSTM and



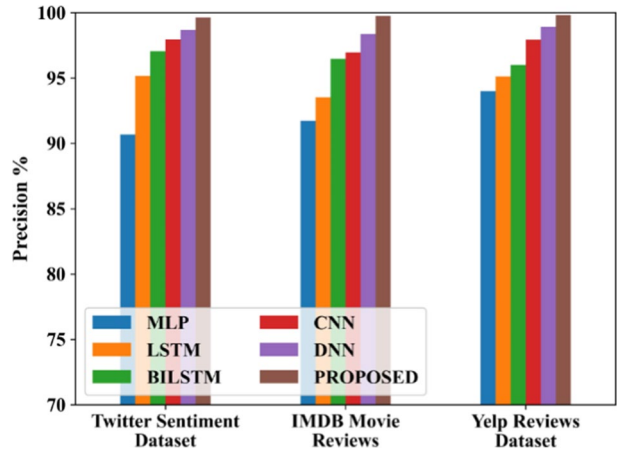
**Fig. 6** Graphical comparison of the accuracy of different datasets for the proposed and the existing method



Bi-LSTM consume more time to train the input data, and it easily falls into an overfitting issue. The conventional CNN method automatically extracts the needed features. But, it generates an increased class imbalance problem during emotion categorization; hence, the accuracy of CNN is diminished. In the proposed study, the effective features selected through the optimization technique help the classifier to attain enhanced classification accuracy. Also, the proposed model maintains the neural network weights to resolve the vanishing gradient issues and helps to attain higher accuracy. Thus, the proposed technique accurately classifies the sentiments and emotions in the provided datasets. Using the Twitter sentiment dataset, the accuracy of CGARU is 99.80%, MLP is 95.70%, LSTM is 96.70%, Bi-LSTM is 97.35%, CNN is 98.15%, and DNN is 98.75. By performing the simulations with the IMDB Movie Review Dataset, the obtained accuracy of CGARU is 99.75%, MLP is 95.10%, LSTM is 95.95%, Bi-LSTM is 97%, CNN is 97.55%, and DNN is 98.95. In the Yelp Reviews dataset, the attained accuracy of CGARU is 98.83%, MLP is 94.45%, LSTM is 96.15%, Bi-LSTM is 96.70%, CNN is 97.30%, and DNN is 98.35%. The result analysis proves that the proposed model attains higher classification accuracy for each dataset than other techniques. The comparative analysis in terms of precision performance is illustrated in Fig. 7.

The comparison analysis in terms of the precision measure states that the proposed CGARU-based classification model provides higher precision results than the other techniques. Because of various issues, the existing deep learning techniques have failed to provide enhanced precision performance. Computational complexity is the major issue in the existing techniques in which the overall system's performance degrades. But, the proposed study used the most effective sentiment-based emotion classification techniques. Moreover, the proposed model minimizes computational complexity and enhances classification performance. Using the Twitter sentiment dataset, the attained precision value of the proposed CGARU is 99.64%, MLP is 90.67%, LSTM is 95.17%, Bi-LSTM is 97.04%, CNN is 97.95%, and DNN is 98.69%. By analyzing the classification performance using the IMDB Movie Review Dataset, the obtained precision of CGARU is 99.75%, MLP is 91.73%, LSTM is 93.53%, Bi-LSTM is 96.46%, CNN is 96.95%, and DNN is 98.37%. Finally, the Yelp Reviews dataset attained the precision value of CGARU is 98.83%, MLP is 94.45%, LSTM is 96.15%, Bi-LSTM is 96.70%, CNN is 97.30%, DNN is 98.35%. The attained values show that the proposed model's

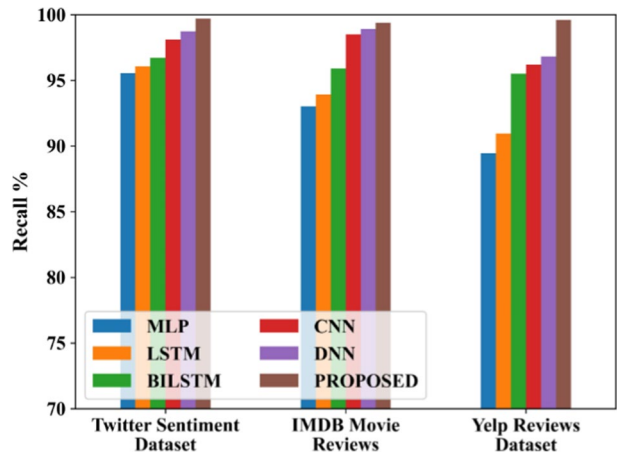
**Fig. 7** Graphical comparison of the precision of different datasets for the proposed and the existing method



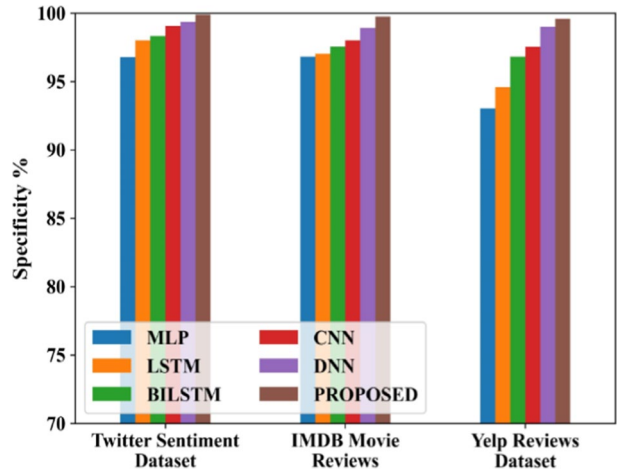
precision is superior to the other methods. The performance comparison of proposed and existing techniques in terms of recall is shown in Fig. 8.

The above graphical representation exhibits the effectiveness of a proposed model. For each dataset, the recall performance of a proposed is higher than the other compared techniques. This analysis proves the effectiveness of the proposed classifier in determining the perfect positive predictions from all the positive predictions made. The Twitter sentiment dataset achieved the recall value of a proposed CGARU is 99.70%, MLP is 95.54%, LSTM is 96.06%, Bi-LSTM is 96.71%, CNN is 98.10%, DNN is 98.73%. By analyzing the classification performance using the IMDB Movie Review Dataset, the obtained recall of CGARU is 99.38, MLP is 93.01%, LSTM is 93.92%, Bi-LSTM is 95.90%, CNN is 98.50%, and DNN is 98.90%. Whereas the Yelp Reviews dataset attained the recall value of CGARU is 99.60%, MLP is 89.44%, LSTM is 90.95%, Bi-LSTM is 95.50%, CNN is 96.19%, and DNN is 96.82%. The simulation results state that the proposed classification model is highly suitable for sentiment and emotion classification. The performance comparison in terms of specificity is depicted in Fig. 9.

**Fig. 8** Graphical comparison of the recall of different datasets for the proposed and the existing method



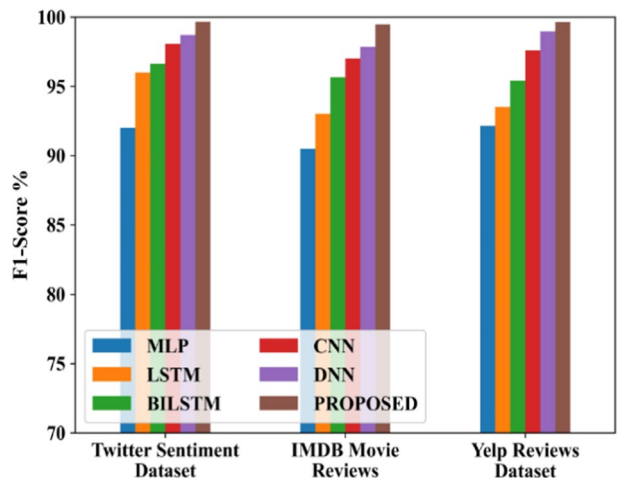
**Fig. 9** Graphical comparison of the specificity of different datasets for the proposed and the existing method



The specificity performance of a proposed dual-stage deep CGARU model is compared with other deep learning techniques. For each dataset, the specificity is determined for all the techniques. Compared with other techniques, the MLP attained reduced specificity performance in sentiment and emotion classification. Using the Twitter Sentiment dataset, the specificity of a proposed CGARU is 99.85%, MLP is 95.54%, LSTM is 96.71%, Bi-LSTM is 97.36%, CNN is 98.10%, DNN is 98.73%. Using the IMDB Movie Review Dataset, the obtained precision of CGARU is 99.75, MLP is 94.60%, LSTM is 96.92%, Bi-LSTM is 98.07%, CNN is 98.50%, and DNN is 99%. Finally, the Yelp Reviews dataset attained the precision value of CGARU is 99.83%, MLP is 96.05%, LSTM is 96.72%, Bi-LSTM is 97.07%, CNN is 97.91%, DNN is 98.44%. The mentioned analysis proves the efficacy of a proposed model. The performance comparison analysis in terms of the F1-score is shown in Fig. 10.

The above graphical representation represents the F1-score comparison of proposed and conventional deep learning techniques. The graph mentions that the proposed

**Fig. 10** Graphical comparison of the F1-score of different datasets for the proposed and the existing method

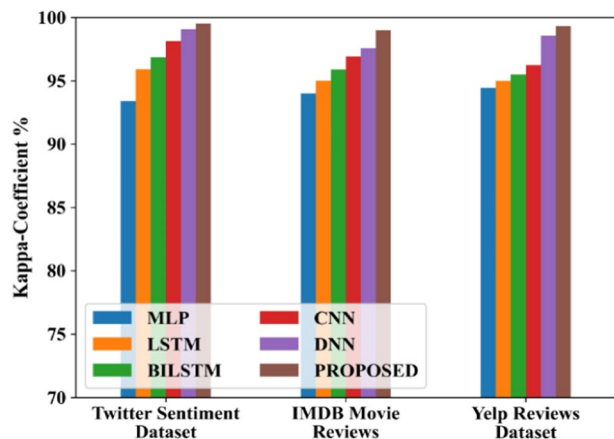


dual-stage deep CGARU model obtains higher F1-score results for all datasets. Using the Twitter sentiment dataset, the F1-score of a proposed CGARU is 99.67%, MLP is 92%, LSTM is 96%, Bi-LSTM is 96.63%, CNN is 98.08%, and DNN is 98.71%. By determining the classification performance using the IMDB Movie Review Dataset, the obtained F1-score of CGARU is 99.47, MLP is 90.49%, LSTM is 93.01%, Bi-LSTM is 95.66%, CNN is 97.01%, and DNN is 97.85%. On the other hand, the Yelp Reviews dataset attained the F1-score value of CGARU is 99.66%, MLP is 92.14%, LSTM is 93.52%, Bi-LSTM is 95.41%, CNN is 97.59%, and DNN is 98.97%. The result shows that the proposed model is highly better regarding F1-score performance than others. The kappa performance comparison of proposed and existing techniques is shown in Fig. 11.

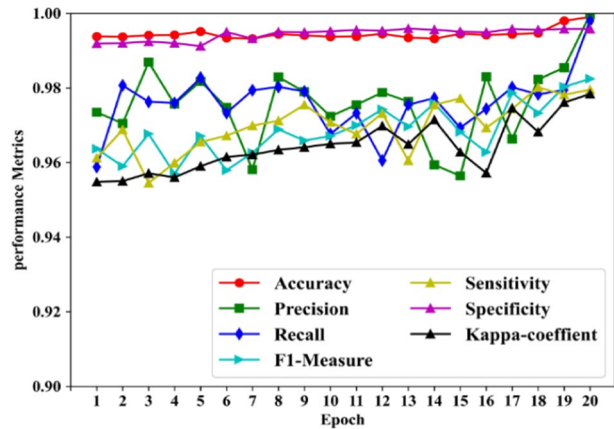
The analysis of a kappa performance of proposed and existing techniques shows that the proposed sentiment-based emotion classification is better than the other techniques. Analyzing the classifier performance with different datasets enhances the kappa performance in each dataset in the proposed method. This shows that the proposed model is appropriate for sentiment and emotion classification. By performing the simulation using the Twitter sentiment dataset, the kappa coefficient of a proposed CGARU is 99.52%, MLP is 93.40%, LSTM is 95.92%, Bi-LSTM is 96.84%, CNN is 98.13%, and DNN is 99.07%. By determining the classification performance using the IMDB Movie Review Dataset, the obtained precision of CGARU is 98.99%, MLP is 94.01%, LSTM is 95.02%, Bi-LSTM is 95.90%, CNN is 96.91%, and DNN is 97.57%. On the other hand, the Yelp Reviews dataset attained the precision value of CGARU is 99.32%, MLP is 94.44%, LSTM is 95%, Bi-LSTM is 95.50%, CNN is 96.25%, DNN is 98.57%. The comparative analysis of a kappa coefficient shows that the proposed framework resulted in optimal performance compared to the other classifiers. The proposed approach on all three datasets resulted in a better kappa score than the other algorithms. This is because of improvements in the model that resulted in accurate classification. The proposed study analyses the classifier's performance by varying the epoch size, depicted in Fig. 12.

By varying the epoch size from 1 to 20, the performance is analyzed for the proposed classification model. The performance of the proposed model is influenced by the number of epochs followed. The proposed study also analyses the performance of each classified emotion. Table 7 shows the performance attained for each emotion in the proposed study.

**Fig. 11** Graphical comparison of the kappa-coefficient of different datasets for the proposed and the existing method



**Fig. 12** Overall performance analysis by varying epochs



The performance, like accuracy, precision, recall, specificity, F1-score and kappa coefficient, is analyzed for varied classes of emotions in three datasets. Initially, the developed dual-stage deep CGARU model classifies the sentiments from the input data. The GARU module also classifies emotions based on positive, negative and neutral sentiments. The performance is varied for each class of emotions according to its nature. The results reveal that the proposed model is more efficient for classifying emotions from the input text data.

#### 4.4 Discussion

The overall simulation results and outcomes prove that the proposed model can accurately predict the emotional tone behind the input text. One of the significant advantages of the proposed method is that it can deal with unstructured textual data, which is a real-time problem in most of the existing sentiment analysis frameworks. The unstructured data presents complexities such as difficulty in understanding the slang used, original emotion behind the words, number of emotions exhibited and processing of words in sequence. Due to these complexities, the unstructured data issue is ill-focused in most existing sentiment analysis mechanisms. However, the proposed framework efficiently classified the unstructured data and provided accurate emotional labels. The analysis conducted in the last section also proved the performance efficacy of a proposed method compared to other algorithms. The existing state-of-the-art works have been evaluated, and the results are presented in Table 8.

The comparison shows that the proposed approach attained state-of-the-art results compared to the other techniques. It is also seen that the proposed method outperformed the existing techniques in terms of all the considered metrics. Most of the approaches considered for comparison considered the LSTM model to preserve the temporal features. However, the proposed model included the GRU model in place of LSTM to preserve the temporal features, further enhancing classification quality. Moreover, the training of a proposed model has been enhanced by using a convolutional module that extracted and learned only the crucial features suitable for classification. Among the compared methods, the method presented by Chen et al. [7] resulted in a better accuracy rate of 92.68% which is incomparable to the other considered techniques. Also, the accuracy attained by the method introduced by Pathak et al. [19] is 88.9% which is also better in terms of sentiment classification. The F1-score attained by the method of Chen et al. [7] is also high, with

**Table 7** Attained performance for varied classes of emotions utilizing three datasets

| Dataset                   | Class    | Accuracy (%) | Sensitivity (%) | Specificity (%) | Precision (%) | F1-Score (%) | Kappa (%) | Recall (%) |
|---------------------------|----------|--------------|-----------------|-----------------|---------------|--------------|-----------|------------|
| Twitter Sentiment Dataset | Happy    | 98.95        | 97.59           | 99.56           | 99.02         | 98.30        | 97.54     | 97.59      |
|                           | Love     | 98.55        | 98.55           | 98.54           | 97.96         | 98.25        | 97.01     | 98.55      |
|                           | Neutral  | 99.8         | 97.26           | 99.89           | 97.26         | 97.26        | 97.15     | 97.26      |
|                           | Disgust  | 99.25        | 96.65           | 99.55           | 96.19         | 96.42        | 96.0      | 96.65      |
|                           | Regret   | 99.75        | 97.14           | 99.89           | 98.07         | 97.60        | 97.47     | 97.14      |
|                           | Stress   | 99.86        | 97.25           | 99.87           | 98.75         | 96.58        | 96.54     | 97.12      |
| IMDB Movie Reviews        | Surprise | 99.6         | 98.56           | 99.67           | 95.80         | 97.16        | 96.94     | 98.56      |
|                           | Happy    | 98.85        | 98.27           | 99.34           | 99.23         | 98.75        | 97.68     | 98.27      |
|                           | Love     | 99.15        | 99.53           | 98.96           | 97.85         | 98.68        | 98.06     | 99.52      |
|                           | Neutral  | 99.55        | 97.05           | 99.73           | 96.35         | 96.70        | 96.46     | 97.05      |
|                           | Disgust  | 99.7         | 96.33           | 99.89           | 98.13         | 97.22        | 97.06     | 96.33      |
|                           | Regret   | 99.6         | 96.74           | 99.78           | 96.74         | 96.74        | 96.53     | 96.74      |
| Yelp Reviews Dataset      | Stress   | 99.8         | 96.78           | 99.65           | 96.78         | 95.87        | 97.54     | 97.26      |
|                           | Surprise | 99.95        | 98.27           | 99.87           | 99.65         | 99.09        | 99.07     | 98.21      |
|                           | Happy    | 99.6         | 97.48           | 99.78           | 97.48         | 97.48        | 97.26     | 97.48      |
|                           | Love     | 98.75        | 99.08           | 97.92           | 99.15         | 99.12        | 96.95     | 99.08      |
|                           | Neutral  | 99.9         | 99.87           | 99.89           | 95.83         | 97.87        | 97.82     | 96.32      |
|                           | Disgust  | 99.3         | 95.51           | 99.62           | 95.51         | 95.51        | 95.13     | 95.51      |
|                           | Regret   | 99.65        | 97.61           | 99.83           | 98.20         | 97.91        | 97.71     | 97.61      |
|                           | Stress   | 99.87        | 96.58           | 98.74           | 96.32         | 98.71        | 96.32     | 97.86      |
|                           | Surprise | 99.9         | 96.29           | 99.94           | 96.29         | 96.29        | 96.24     | 96.29      |

**Table 8** Performance comparison of the proposed and existing state-of-the-art works

| Reference          | Method         | Precision (%) | Recall (%)   | F1-score (%) | Accuracy (%) |
|--------------------|----------------|---------------|--------------|--------------|--------------|
| Chen et al. [7]    | LSTM, Bi-LSTM  | -             | -            | 88.41        | 92.68        |
| Singh et al. [20]  | LSTM-RNN       | 82.34         | 82.12        | 81.23        | 84.56        |
| Du et al. [9]      | BGRU + Capsule | 87.06         | 87.12        | 87.09        | 82.55        |
| Pathak et al. [19] | LSTM           | -             | 87.9         | 87.9         | 88.9         |
| Proposed           | CGARU          | <b>99.75</b>  | <b>99.38</b> | <b>99.47</b> | <b>99.75</b> |

an average F1-score of 88.41%, and this is followed by the method introduced by Pathak et al. [19] with an average F1-score of 87.9%. Apart from these, the recall value scored by Pathak et al. [19] is also high, proving that it offers effective performance. However, the proposed approach resulted in higher precision, recall, F1-score and accuracy values. Thus, it can be proved from the analysis that a proposed approach is more optimal and accurate in performing sentiment classification compared to the existing state-of-the-art techniques.

Though the proposed model is optimal and stable in efficiently classifying sentiments from textual data, some challenges can still be addressed to achieve higher performance. The pre-trained models are highly utilized in most existing domains to achieve higher performance results. This can also be followed in this field to attain better results where the pre-training process can be improved on any considered granularity (i.e. character, word or sub-word). Another possible research direction is deep multi-task Learning, a growing area in NLP-related tasks. Multiple tasks can be learned simultaneously to avoid overfitting issues in the network. Since the existing DL models are highly non-interpretable, explainable DL models can also be followed. The recent focus on common sense knowledge can also be utilized along with the classifier to attain better and satisfactory results. Another research direction arises from the availability of poorly labelled data resources. To deal with such problem, low-resource methods can also be adopted to provide better results in classification. Another research scope is that deeper network models with changes in the architectural design can also help the developers to attain better future results.

## 5 Conclusion

This paper proposes an effective sentiment polarity-based emotion classification with the help of dual stage deep learning model. The proposed model includes four phases: pre-processing, textual feature extraction, feature selection and classification. Initially, the noises in the input data are eliminated in the pre-processing stage through effective mechanisms. After pre-processing, the features are extracted using LSA and LTF-MICF model in the feature extraction stage. The features extracted in this stage contain higher dimensionality, which may generate increased computational complexity. Thus, the proposed work selects the optimal feature set in the feature selection phase using the CAHA approach. Next, the selected features are fed into the deep CGARU model for sentiment and emotion classification. The initial convolutional stage classifies the positive, negative, and neutral sentiments, and then the CGARU stage classifies the multi-class emotions from the given text data. The proposed experimental setup used three datasets, and the performance are analyzed for these datasets. The results show that the proposed model obtained increased classification accuracy for Twitter Sentiment Dataset 99.80%, IMDB Movie Reviews 99.75%, and Yelp

Reviews Dataset 99.83%. Different features will be extracted in future studies using other approaches to improve classification performance. Moreover, the proposed work will be extended to determine emotions through big data.

**Authors contributions** All authors have equal contributions in this work.

**Data availability** Data sharing not applicable to this article.

## Declarations

**Ethical approval** This article does not contain any studies with human participants or animals performed by any of the authors.

**Consent to participate** All the authors involved have agreed to participate in this submitted article.

**Consent to publish** All the authors involved in this manuscript give full consent for publication of this submitted article.

**Conflict of interest** Authors declare that they have no conflict of interest.

## References

1. Acheampong FA, Wenyu C, Nunoo-Mensah H (2020) Text-based emotion detection: Advances, challenges, and opportunities. *Eng Rep* 2(7):e12189
2. Ahire V, Borse S (2022) Emotion detection from social media using machine learning techniques: a survey. In: *Applied Information Processing Systems* Springer, Singapore 83–92
3. Ahmad S, Mehfuz S, Mebarek-Oudina F, Beg J (2022) RSM analysis based cloud access security broker: a systematic literature review. *Clust Comput* 25(5):3733–3763
4. Almeida J, Vilaça L, Teixeira IN, Viana P (2021) Emotion Identification in Movies through Facial Expression Recognition. *Appl Sci* 11(15):6827
5. Balahur A (2013) Sentiment analysis in social media texts. In *Proceedings of the 4th workshop on computational approaches to subjectivity, sentiment and social media analysis* 120–128
6. Birjali M, Kasri M, Beni-Hssane A (2021) A comprehensive survey on sentiment analysis: Approaches, challenges and trends. *Knowl-Based Syst* 226:107134
7. Chen LC, Lee CM, Chen MY (2020) Exploration of social media for sentiment analysis using deep Learning. *Soft Comput* 24(11):8187–8197
8. D'Souza RS, Hooten WM, Murad MH (2021) A proposed approach for conducting studies that use data from social media platforms. In *Mayo Clinic Proc* 96(8):2218–2229
9. Du Y, Zhao X, He M, Guo W (2019) A novel capsule based hybrid neural network for sentiment classification. *IEEE Access* 7:39321–39328
10. Ghosh D (2020) A Sentiment-Based Hotel Review Summarization. In *Emerging Technology in Modelling and Graphics*. Springer, Singapore 39–44
11. Gorodnichenko Y, Pham T, Talavera O (2021) Social media, sentiment and public opinions: Evidence from# Brexit and# USElection. *Eur Econ Rev* 136:103772
12. Guo C, Du Z, Kou X (2018) Products ranking through aspect-based sentiment analysis of online heterogeneous reviews. *J Syst Sci Syst Eng* 27(5):542–558
13. Huddar MG, Sannakki SS, Rajpurohit VS (2021) Attention-based multi-modal sentiment analysis and emotion detection in conversation using RNN
14. Jabreel M, Moreno A (2019) A deep learning-based approach for multi-label emotion classification in tweets. *Appl Sci* 9(6):1123
15. Jain PK, Pamula R, Srivastava G (2021) A systematic literature review on machine learning applications for consumer sentiment analysis using online reviews. *Comput Sci Rev* 41:100413
16. Karthik E, Sethukarasi T (2022) A Centered Convolutional Restricted Boltzmann Machine Optimized by Hybrid Atom Search Arithmetic Optimization Algorithm for Sentimental Analysis. *Neural Processing Letters* 1–29



17. Mao X, Chang S, Shi J, Li F, Shi R (2019) Sentiment-aware word embedding for emotion classification. *Appl Sci* 9(7):1334
18. Nyo MT, Mebarek-Oudina F, Hlaing SS and NA Khan (2022) Otsu's thresholding technique for MRI image brain tumor segmentation. *Multimed Tools Appl* 1–13
19. Pathak AR, Pandey M, Rautaray S (2021) Topic-level sentiment analysis of social media data using deep Learning. *Appl Soft Comput* 108:107440
20. Singh C, Imam T, Wibowo S, Grandhi S (2022) A Deep Learning Approach for Sentiment Analysis of COVID-19 Reviews. *Appl Sci* 12(8):3709
21. Stojanovski D, Strezoski G, Madjarov G, Dimitrovski I, Chorbev I (2018) Deep neural network architecture for sentiment analysis and emotion identification of Twitter messages. *Multimed Tools Appl* 77(24):32213–32242
22. Sudhir P, Suresh VD (2021) Comparative study of various approaches, applications and classifiers for sentiment analysis. *Glob Trans Proc* 2(2):205–211
23. Taj S, Shaikh BB, Meghji AF (2019) Sentiment analysis of news articles: a lexicon based approach. In 2019 2nd international conference on computing, mathematics and engineering technologies (iCoMET) IEEE 1–5
24. Yadav A, Vishwakarma DK (2020) Sentiment analysis using deep learning architectures: a review. *Artif Intell Rev* 53(6):4335–4385
25. Yue L, Chen W, Li X, Zuo W, Yin M (2019) A survey of sentiment analysis in social media. *Knowl Inf Syst* 60(2):617–663
26. Zad S, Heidari M, James Jr H, Uzuner O (2021) Emotion detection of textual data: An interdisciplinary survey. In: 2021 IEEE World AI IoT Congress (AIIoT) 0255–0261
27. Zhang J, Yin Z, Chen P, Nichele S (2020) Emotion recognition using multi-modal data and machine learning techniques: A tutorial and review. *Inf Fusion* 59:103–126

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

## Authors and Affiliations

S J R K Padminivalli V<sup>1</sup> · M. V. P. Chandra Sekhara Rao<sup>2</sup> · Naga Sai Ram Narne<sup>3</sup>

✉ S J R K Padminivalli V  
srivallivasantham@gmail.com

<sup>1</sup> Department of Computer Science and Engineering, Dr. Y.S.R. ANU College of Engineering & Technology, Acharya Nagarjuna University, Guntur, Andhra Pradesh 522510, India

<sup>2</sup> Department of C.S.B.S, R.V.R. & J.C. College of Engineering, Chowdavaram, Andhra Pradesh 522019, India

<sup>3</sup> Northwest Missouri State University, 800 University Drive, Maryville, USA