Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Enhancing Sentiment Analysis on Social Media using Lemurs Optimization Algorithm with Deep Ensemble Model

K. Manikandan[1], V. Ganesh[2]

[1]Research Scholar, Department of Computer and Information Science, Annamalai University, Annamalai Nagar, Tamilnadu, India, skmaniswamimalai@gmail.com
[2]Department of Computer Science, Government Arts College(A), Kumbakonam, Tamil Nadu, India.

Sentiment analysis (SA), also addressed as opinion mining, is the computational process of recognizing the emotional tone behind a text. It utilizes natural language processing (NLP), machine learning (ML), and statistical approaches for classifying the sentiment expressed in a sentence, document, or phrase as negative, positive, and neutral. This analysis is important for understandinggeneral opinion, customer feedback, brand perception, and trends in various domains namely market research, social media, and customer service. Deep learning (DL) has transformedSA by allowingapproachesfor automatically learning intricate patterns and representations from raw text data, eliminating the necessity for manual feature engineering. Awidespread training on huge corpora of considered data, these DLapproaches discern sentiment with unprecedented accuracy, even among the workings of sarcasm, irony, and colloquialisms common in social media discourse. The scalabilityand adaptability of DLstructureshave propelled SA to novel heights, allowing businesses, researchers, and policymakers with valuable opinions into public opinion and sentiment dynamics. This study concentrates on the design and development of Sentiment Analysis on Social Media using Lemurs Optimization Algorithm with Deep Ensemble (SASM-LOADE) technique. The goal of the SASM-LOADE technique is to recognize different kinds of sentiments that exist in the social media data. In the SASM-LOADE technique, the primary phase of preprocessing involves different sub-processes. In addition, the feature extraction using TF-IDF model takes place which generates word embedding. Next, the detection of sentiments can be performed by the use of ensemble model comprising three classifiers such as gated recurrent unit (GRU), recurrent neural network (RNN), and bidirectional long short-term memory (BiLSTM) model. For hyperparameter tuning of the DL models, the LOA is applied in the SASM-LOADE technique. In order to demonstrate the significance of the SASM-LOADE technique, a widespread simulation analysis is performed. The experimental values illustrate that the SASM-LOADE technique reaches promising results over existing models.

Keywords: Sentiment Analysis; Lemurs Optimization Algorithm; Social Media; Deep Learning; Word Embedding.

1. Introduction

The high reputation of quickly developing online social networks and electric media based cultures has stimulated the researchers to follow their work on sentiment analysis (SA) [1]. Generally, societies moderately evaluate their users or public opinions regarding their services or products from the text of social media platforms [2]. The online service providers are curved on measuring information on online media, blogs, comments, tweets, and product reviews. This evaluation was developed mainly for their decision-making or enhancement of their superiority of products or services [3]. The SA uses include many fields such as election campaigning, social event planning, healthcare observing, awareness services, and user products. SA is the analysis of feelings and forecasts. SA is the text mining method, which utilizes natural language processing (NLP) for dual identification of text [4]. SA is executed in 4 stages built on the range of text such as sentence-level, aspect-level, word-level, and document-level SA. In the document-level SA, complete view of the text regarding solitary entity is gathered into negative or positive [5]. In sentence-level SA, the opinion stated in a sentence is categorized as both negative and positive. In aspect-level SA, feelings regarding entities were clustered based on exact object elements. At word-level SA, thoughts about entities are clustered based on exact terms [6].

Owing to the huge data growth and the volume of data being switched and formed each second, the need to know, extract, and analyze this data has enlarged remarkably [7]. Then the regular machine learning (ML) models and Neural Networks were not sufficient to be gained on this large data, deep learning (DL) was crucial in big data period. DL is a sub-type of ML and a change of neural network. That is, regular neural network is a solitary network with output and input layers along with hidden layers (HL) among, where computation is completed [8]. A Deep Neural Network (DNN) mainly contains many neural networks where the output of single network is an input to the subsequent network. This idea has overwhelmed the restriction of the integer of HLs in Neural Networks and made functioning with big data more viable [9]. DL networks absorb the features on its specific, that is, it has turned into apparent as a strong ML model that acquires numerous layers of data features and prompts outcomes of prediction. DL has been currently utilized in numerous applications in the area of signal and data processing, particularly with the development of big data [10]. Furthermore, DL networks were employed in SA and opinion mining.

This study concentrates on the design and development of SA on Social Media using Lemurs Optimization Algorithm with Deep Ensemble (SASM-LOADE) technique. The goal of the SASM-LOADE technique is to recognize different kinds of sentiments that exist in the social media data. In the SASM-LOADE technique, the primary phase of preprocessing involves different sub-processes. In addition, the feature extraction using TF-IDF model takes place which generates word embedding. Next, the detection of sentiments can be performed by the use of ensemble model comprising three classifiers such as gated recurrent unit (GRU), recurrent neural network (RNN), and bidirectional long short-term memory (BiLSTM) model. For hyperparameter tuning of the DL models, the LOA is applied in the SASM-LOADE technique. The experimental values illustrate that the SASM-LOADE technique reaches promising results over existing models.

2. Literature Works

Brinda et al. [11] presented a complete study of SA in social media communication over the combination of DL models with NLP technique. This method will permit the accurate classification of appropriate content that is dynamically altering and the real-time range of key phrases based on obtainable data. The research constructs a message SA method and an image message multi-modal SA method, discovering unimodal and multi-modal SA techniques in social networks. Chiranjeevi and Rajaram [12] present a Light DL (LightDL)-based suggested method that utilizes Twitter-based analyses. Initially, the data is composed of Twitter and prepared by succeeding data procedures. Next, this data pre-processing is served into the LightDL method that absorbs the significant features such as unigrams, hashtags, multi-grams, and much more from every part of data. Lastly, the data has been categorized into negative, positive, and neutral as per the learned feature. In [13], a new DL structure is presented for Roman Urdu and English language SA depends on dual layers such as LSTM method for long-term dependence protection and a single-layer CNN method for local feature extractor. In order to get the last identification, the feature map learned by LSTM and CNN was served to numerous ML classification algorithms. Numerous word embedding methods support this idea.

Paulraj et al. [14] offer a proficient SA model in Twitter data. The Twitter dataset is pre-processed and contains tokenization, stemming, number and stop word removal, etc. The pre-processed words are then distributed into the HDFS (Hadoop Distributed File System) to decrease the repeated words and remove them utilizing the MapReduce model. The emoticons and the non-emoticons are removed as features. The resulting features are hierarchical with their proposed meaning. Next, the classification is executed using the DLMNN (DL Modified Neural Network). Kora and Mohammed [15] projected a meta-ensemble DL technique in order to enhance the performance of SA. This model trains and merges baseline DL techniques utilizing 3 stages of meta-learners. Then, numerous experimentations are led on 6 benchmark databases of SA in dissimilar languages to assess the performance of projected meta-ensemble DL model.

Singh et al. [16] unite an improved dynamic weight layer with Bi-LSTM to precisely obtain contextual. It takes graph convolutional network (GCN) to encode syntactic data from the syntactic dependency tree. Also, a model for interactive attention is used in order to determine the complex relations among aspect and context terms, which outcomes in the renovation of those word representations. Adam and Setiawan [17] analyze the sentiments of the feelings utilizing DL models and variants in situations. To conduct SA, tweets are gathered by edging the information. While, tweets are considered neutral, negative, and positive, which is signified as 0, -1, and 1, respectively. The model employed to categorize tweet emotion is CNN and Gated Recurrent Units (GRUs).

3. The Proposed Model

In this study, we concentrate on the design and development of SASM-LOADE technique. The goal of the SASM-LOADE technique is to recognize different kinds of sentiments that exist in the social media data.To accomplish that, the SASM-LOADE technique has preprocessing, feature extractor, ensemble learning, and LOA based parameter tuning. Fig. 1 exemplifies the entire process of SASM-LOADE approach.

3.1. Preprocessing

Primarily, the SASM-LOADE technique takes place preprocessing involves different sub-processes.The quality of results relies on the input dataset [18]. Generally, pre-processing step has the subsequent steps:

- Textcleaning. Initially, the text is cleaned. Based on the task, cleaning involves eliminating various tags, non-alphabets, spaces, punctuation, URLs, and other markup components;

- Segmentationandtokenization. The text is divided into separate words and sentences and is relevant in most cases. In general, each word is converted to lowercase after tokenization;

- Lemmatizationandstemming. The text has many grammatical forms of a similar word and words with similar roots. Lemmatization is used to reduce the word form into a lemma-its dictionary (normal) form. Stemming is a heuristic method that cutoff excess" from the root, which leads to loss of derivational suffixes. Lemmatization is a subtle process that makes use of morphological analysis and vocabulary to diminish a word into canonical form (lemma).

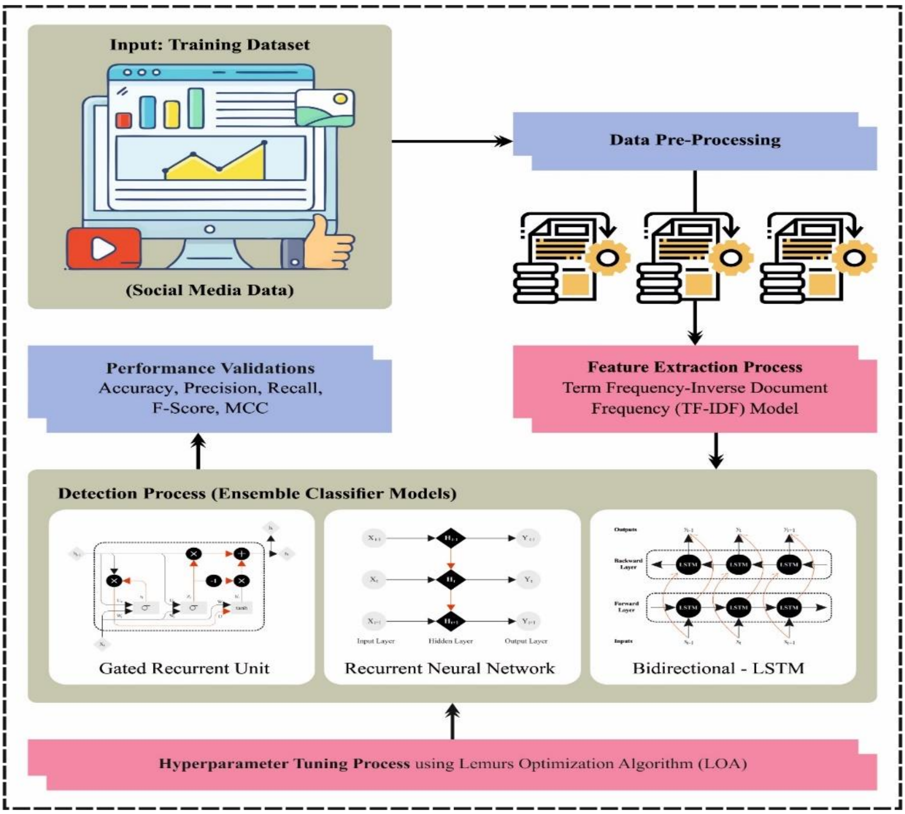

Fig. 1. Overall process of SASM-LOADE technique

- Descriptionofcontext-independentfeatures that describe the token that is not reliant on the adjacent components;

- Filteringsignificanceandusingafiltertostopwords. A stop word is common word that doesn't add further data to the text. After ML is applied to texts, this word may add various noises, hence there is a need to remove them;

- Dependencyparsing. This is a form of tree configuration, but the token is allocated to single parent, and the kind of connection is introduced;

- Renovatingtextcontenttoavectorrepresentation which highpoints words utilized in same contexts.

3.2. Feature Extraction

In this work, the feature extraction using TF-IDF model takes place which generates word embedding.TF is the ratio of appearing to the overall amount of words in the document. Therefore, the significance of word t in a single document $\mathrm{d_i}$ is assessed:

$$\mathrm{TF}(\mathrm{t}, \mathrm{d_i}) = \frac{\mathrm{n_t}}{\sum_{\mathrm{k}} \mathrm{n_k}}, \tag{1}$$

In Eq. (1), $\mathrm{n_t}$ refers to the number of $\mathrm{t^{th}}$ word occurrences from the document and the denominator of the fraction represents the overall amount of words in the document.

However, frequency scoring hasa problem: words with high frequency have high scores.

IDF is the inverse of the frequency that specific word takes place in the document of the collection

$$\mathrm{IDF}(\mathrm{t}, \mathrm{d_i}, \mathrm{D}) = \log \frac{|\mathrm{D}|}{|\{\mathrm{d_i} \in \mathrm{D} | \mathrm{t} \in \mathrm{d_i}\}|}. \tag{2}$$

In Eq. (2), $|\mathrm{D}|$ refers to the document count in the corpus, $\{\mathrm{d_i} \in \mathrm{D} | \mathrm{t} \in \mathrm{d_i}\}$ denotes the document count in the collection D.

There is IDF value for all the unique words within the document collection. IDF metric decreases the weight of commonly corpus-used words.

$\mathrm{TF} - \mathrm{IDF}$ is a statistical measure to estimate the word significance in the document viz., part of corpus or collection:

$$\mathrm{TF} - \mathrm{IDF}(\mathrm{t}, \mathrm{d_i}, \mathrm{D}) = \mathrm{TF}(\mathrm{t}, \mathrm{d_i}) \times \mathrm{IDF}(\mathrm{t}, \mathrm{d_i}, \mathrm{D}). \tag{3}$$

TF-IDF scoring upsurges according to the word occurrence in the document.

The drawback of the frequency technique based on these metrics is that it doesn't consider the context of single words. Furthermore, it doesn't discriminate the semantic similarity of words. Each vector is similarly far from one other in the feature space. Fig. 2 defines the structure of TF-IDS.

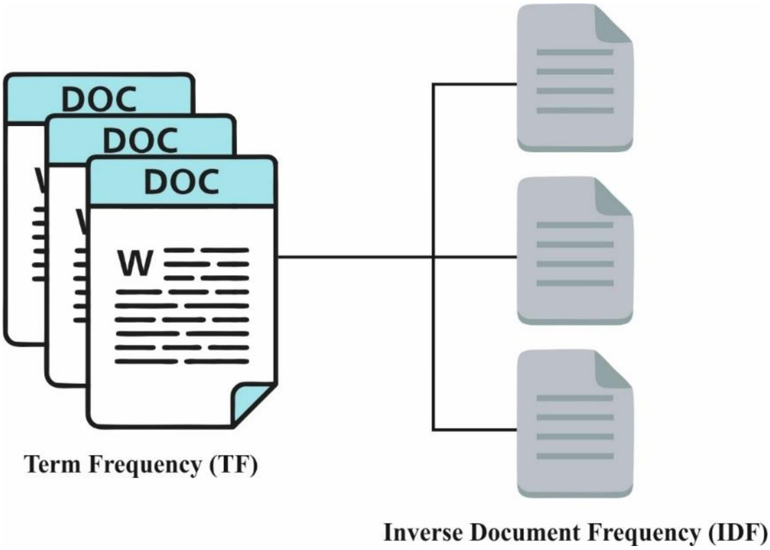

Fig. 2. Structure of TF-IDF

3.3. Ensemble Learning

Next, the detection of sentiments can be performed by employing of ensemble model encompassing3 classifiers such as GRU, RNN, and BiLSTM model.

3.3.1. GRU Classifier

A GRU is variant of LSTM that resolves the gradient vanishing or exploding problems while handling the dependency connection of long-term, and it keeps the connection of short- and long-distance dependency for temporal information [19]. While offering a simple network architecture, GRU retains these advantages. The GRU has only reset and updated gates compared to LSTM networks with three-gate structure.

The forward computation of GRU is given in the following:

Update gate:

$$u_t = \sigma(W_u x_t + V_u h_{t-1} + b_u) \tag{4}$$

Reset gate:

$$r_t = \sigma(W_r x_t + V_r h_{t-1} + b_r) \tag{5}$$

Memory unit:

$$\tilde{h}_t = \tanh(W_h x_t + V_h(r_t^{\star} h_{t-1}) + b_r) \tag{6}$$

Output:

$$h_t = u_t h_{t-1} + (1 - u_t)\tilde{h}_t \tag{7}$$

Here $W_u, U_\Gamma, W_h, V_u, V_\Gamma$, and V_h are the weight matrices, b_i, b_f, b_c, and b_o are the bias vectors, and tanh is the tangent activation function.

3.3.2. RNN Classifier

An RNN is a kind of FFNN with a specific memory function and a ring structure. Its input consists of obtained data in the preceding time and existing input samples such that data can be cycled from the network at any time. Where x, O, and H are the input, output, and hidden layers (HL), and, V, and w are the weights of corresponding layer. The resultant of HL is given below:

$$\mathrm{h_t = \sigma(W_{xh}x_t + W_{hh}h_{t-1} + b_h)} \tag{8}$$

The output is:

$$\mathrm{o_{t+1} = \sigma\left(W_{hy}h_t + b_y\right)} \tag{9}$$

$$\mathrm{y_t = softmax(o_t)} \tag{10}$$

Where $\mathrm{y_t}$ is the output in time $\mathrm{t}, \mathrm{W_{hh}}, \mathrm{W_{xh}}$, and $\mathrm{W_{hy}}$ are the weight matrices, $\mathrm{b_h}$ and $\mathrm{b_y}$ are the bias term, $\mathrm{h_t}$ is the HL at t, and $\mathrm{x_t}$ is the input dataset.

3.3.3. BiLSTM Classifier

An LSTM is a special type of RNN that can learn the dependency connection of long-distance, illustrate the time series data, and resolve the problems of exploding or vanishing gradients. Hochreiter and Schmidhuber in 1997 first proposed LSTM model. Later, several researchers improved and optimized it, which led to its wider usage amongst various NLP aspects and rapid development.

Each unit of LSTM has forget, input, and output gates and memory cells. The memory unit is circularly connected with one another. Three non-linear gate cells are utilized for adjusting the input and output data flows of memory unit.

The forward calculation of LSTM is given below.

Input gate:

$$\mathrm{i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)} \tag{11}$$

Forget gate:

$$\mathrm{f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)} \tag{12}$$

Memory unit:

$$\mathrm{c_t = f_t \cdot c_{t-1} + i_t \cdot tanh(W_c x_t + U_c h_{t-1} + b_c)} \tag{13}$$

Output gate:

$$\mathrm{o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)} \tag{14}$$

Result output:

$$\mathrm{h_t = o_t \cdot tanh(c_t)} \tag{15}$$

Where f, i, and o are the activation vector of the forget, input, and output gates, correspondingly; $\mathrm{x_t}$ is the input vector at t time; hrefersto the resultant vector of LSTM unit, c is the vector memory unit, σ and tanh are the activation functions and

$W_i, U_i, W_f, U_f, W_c, U_c, W_o$, and U_o are the weight matrix; b_i, b_f, b_c, and b_o are the bias vectors.

A BiLSTM is an enhanced version of LSTM. One-directional LSTM exploits historical data to deduce succeeding data, requires data processing, and prevents it from integrating context information or accessing future context, affecting the performance of prediction system. A BiLSTM exploits two LSTM model for training together and starts their corresponding sequence from opposite ends while connected to a similar output layer. Therefore, it integrates the historical and present data of all the points. A BiLSTM consists of forward and backward computations.

The forward computation of h hidden vector is given below:

$$h_t = \mathrm{LSTM}(x_t, h_{t-1}) \tag{16}$$

The backward computation of h hidden vector is given below:

$$h_t = \mathrm{LSTM}(x_t, h_{t-1}) \tag{17}$$

The output is

$$y_t = g\left(W_{hy}h_t + W_{hy}h_t + b_y\right) \tag{18}$$

In Eq. (18), x_t is the input data, y_t is the output at t time, W_{hy} and W_{hy} are the weight matrix, and b_y is the bias term.

3.4. Hyperparameter Tuning Process

For hyperparameter tuning of the DL models, the LOA is applied in the SASM-LOADE technique.This section presents mathematical formula and the steps for LOA [20]. LO is a population-based algorithm hence, the lemur set is formulated in the matrix procedure. The input matrix of population for the LO technique is defined in Eq. (19).

$$X = \begin{bmatrix} l_1^1 & l_1^2 & \cdot & l_1^d \\ l_2^1 & l_2^2 & \cdot & l_2^d \\ \vdots & \vdots & \vdots & \vdots \\ l_n^1 & l_n^2 & \cdot & l_n^d \end{bmatrix} \tag{19}$$

In Eq. (1), X is the matrix of set in n×d size, n denotes the solution candidate, and d is the decision variable. The LO process runs into several steps for resolving the optimization problems such as Feature selection (FS).

Step 1: Determine the Lemurs parameter: N Population, Max_{iter} maximal iteration count, d dimensionality of search range. In addition, UB and LB are the upper, and lower bounds.

Step 2: Create X decision variables in i^{th} solution using the following expression:

$$X_i^j = \left(LB + \left(UB_j - LB_j\right)\right) \times r \tag{20}$$

In Eq. (20), r denotes the uniformly distributed random value $\in [0, 1]$.

Step 3: At all the iterations, evaluate the Free Risk Rate (FRR) which is coefficient of LO

$$FRR = HRR - t \times ((HRR - LRR)/Max_{iter} \tag{21}$$

Where t indicates the existing iteration count. Max_{iter} is the iterationsize. Eq. (21) uses two predefined and constant values as Low-Risk Rate (LRR) and High-Risk Rate (HRR).

Step 4: Calculate the fitness values (FV) for x_i^j, as follows:

$$Fit(x_i^j) = \alpha \times (1 - Acc) + \beta \times (s/S) \tag{22}$$

Where $Fit(x_i^j)$ is the FV, smaller s denotes the overall features selected, S indicates the maximal feature selected, and Acc shows the accuracy of all the subsets which can be extracted by KNN classifier for evaluating the elected subset in all the iterations.

Step 5: To enhance the FVs of lemurs, we classify them into 2 procedures. First, recognizing the best near lemurs (bnl) is identified that implies electing the solution with the low FV. bnl provides the optimum feature based on the FS objective for the existing iteration. Next, the global best lemur (gbl)waselected from the entire populations.

Step 6: Fixed the r_1 value which is a randomly created integer $\in [0,1]$, and compare it with FRR. Next, the position was upgraded for all the lemurs away from the risk-based based on the following expression:

$$X_i^j = \begin{cases} x(i,j) + |(x(i,j) - x(bnl,j)| \times (r_3 - 0.5) \times 2; \; r_1 < FRR \\ x(i,j) + |(x(i,j) - x(gbl,j)| \times (r_3 - 0.5) \times 2; \; r_1 > FRR \end{cases} \tag{23}$$

In Eq. (23), r_1 indicates the random value $\in [0,1]$. (i,j) is the present i^{th} lemurs of N^{th} population, which is the candidate solution at the j^{th} parameter. bnl and gbl are the best near and the global best lemurs at the iteration.

Algorithm 1 presents the LOA that randomly begins with the swarm of lemurs. Then, it attempts to move nearby the lemurs with minimalFV via dance hup, as the best FV. The optimization process arbitrarily begins creating a collection of lemurs. The FRR values start neighboringLRR, representing that the lemur starts to shift and move towards the optimum one through the 'dance hup'. The LO executes these dance hup actions to reduce the FRR value nearbyHRR. Next, the leap-up act is used to represent the lemur toward the global optimum solution. This procedure is repeated till the ending condition is met.

Algorithm 1: Pseudocode of LOA
Determine the parameters UB, LB, Max_{iter}, Dimension d, LRR, HRR.
Initialize the size of N Lemurs population.
Create random population. $\rhd$ Eq. 20
$r = 1$
While t< Max_{iter} do
Compute coefficient FRR $\rhd$ Eq. 21
Compute Fit for all the candidate Lemurs
Sort lemurs candidates

Upgrade global optimum candidate lemur gbl

Upgradebnl

for i = 1 to N do

Set $r_1 \leftarrow \mathrm{rand}[0,1]$

if $r_1 < \mathrm{FRR}$ then

Upgrade all the decision variables through dance $\rhd$ Eq. 23 hup

else

Update all the decision variables through leap up $\rhd$ Eq. 23

end if

end for

end While

The fitness optimal is a main aspect of controlling the LOA solution. The parameter process contains the encoded method for measuring the solution of candidate outcomes. During this case, the LOAassumes that accuracy is a primary condition to plan the fitness function (FF) that is expressed as:

$$\mathrm{Fitness} = \max(\mathrm{P}) \quad (24)$$

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \quad (25)$$

Whereas, FP and TPsignify the false and true positive rates.

4. Performance Validation

The performance assessment of the SASM-LOADE approach was examined utilizing dual datasets: Sentiment140 [21] and Airlines [22] dataset. The Sentiment140 dataset contains 2000 samples and Airlines dataset includes 2000 samples as signified in Table 1.

Table 1 Details of 2 datasets

Sentiment140 Dataset	
Class	No. of Instances
Negative	1000
Positive	1000
Total Instances	2000
Airlines Dataset	
Class	No. of Instances

Negative	1000
Positive	1000
Total Instances	2000

Fig. 3 establishes the classifier outcomes of the SASM-LOADE model under Sentiment140 dataset. Figs. 3a-3b describes the confusion matrices delivered by the SASM-LOADE technique at 70:30 of TRAS/TESS. The outcome inferred that the SASM-LOADE system has known and considered all 2 class labels exactly.Also, Fig. 3c defines the PR study of the SASM-LOADE technique. The result described that the SASM-LOADE system has attained highest PR performance under every class. Finally, Fig. 3d establishes the ROC investigation of the SASM-LOADE technique. The outcomes represented that the SASM-LOADE approach has resulted in proficient outcomes with maximum values of ROC under dissimilar class labels.

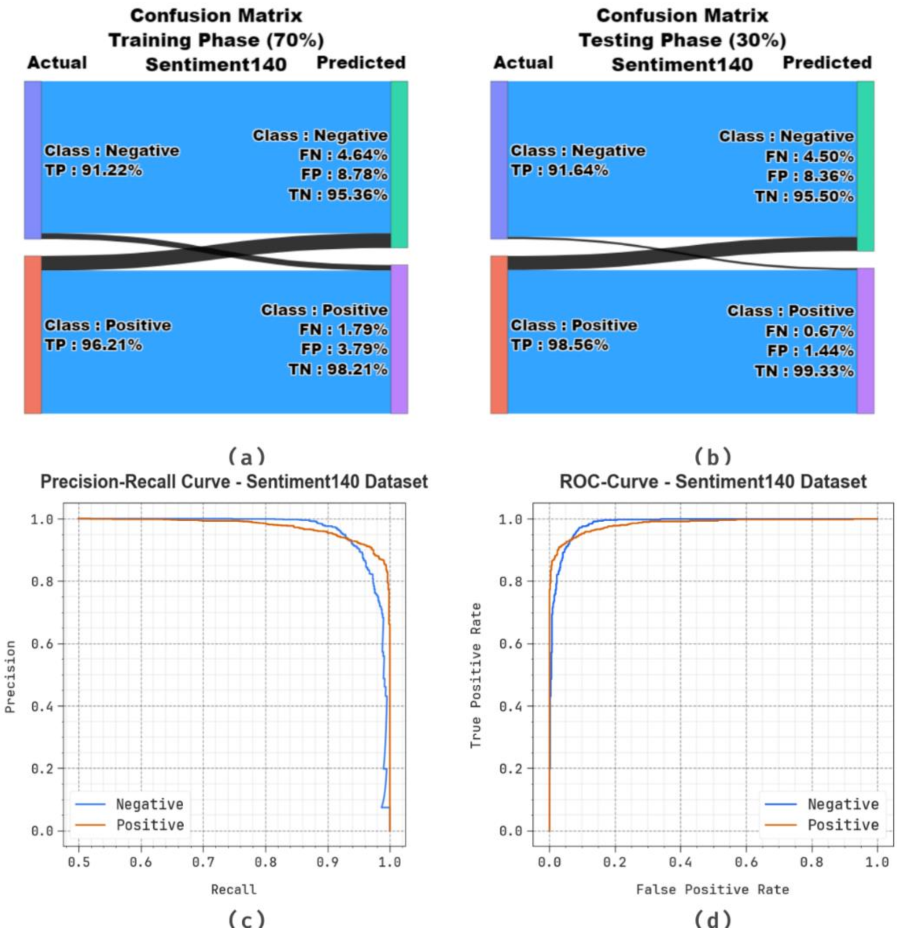

Fig. 3. Sentiment140 dataset (a-b) confusion matrices and (c-d) PR and ROC curves

Table 2 and Fig. 4 demonstrate the complete sentiment recognition outcomes of the SASM-LOADE approach on the Sentiment140 database. The outcomes epitomized that the SASM-

LOADE model correctly recognized negative and positive samples. On 70%TRAS, the SASM-LOADE system provides an average $accu_y$ of 93.57%, $prec_n$ of 93.71%, $reca_l$ of 93.57%, F_{score} of 93.57%, and MCC of 87.29%. Moreover, on 30%TESS, the SASM-LOADE method provides average $accu_y$ of 94.83%, $prec_n$ of 95.10%, $reca_l$ of 94.83%, F_{score} of 94.83%, and MCC of 89.93%.

Table 2 Sentiment recognition outcome of SASM-LOADE technique on Sentiment140 database

Class	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}	MCC
TRAS (70%)					
Negative	96.43	91.22	96.43	93.75	87.29
Positive	90.71	96.21	90.71	93.38	87.29
Average	93.57	93.71	93.57	93.57	87.29
TESS (30%)					
Negative	98.67	91.64	98.67	95.02	89.93
Positive	91.00	98.56	91.00	94.63	89.93
Average	94.83	95.10	94.83	94.83	89.93

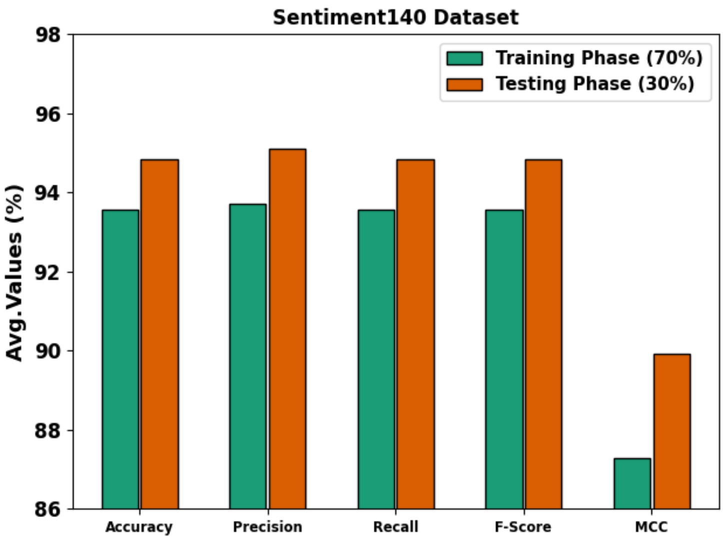

Fig. 4. Average of SASM-LOADE technique on Sentiment140 database

In Fig. 5, the training and validation accuracy outcomes of the SASM-LOADE method on Sentiment140 dataset are established. The accuracy values are calculated over a range of 0-25 epochs. The figure emphasized that the training and validation accuracy values show a rising tendency which reported the capability of the SASM-LOADE technique with enhanced performance over numerous iterations. Moreover, the training and validation accuracy remains nearer over the epochs, which specifies low marginal overfitting and displays superior performance of the SASM-LOADE approach, assuring consistent forecasts on unseen models.

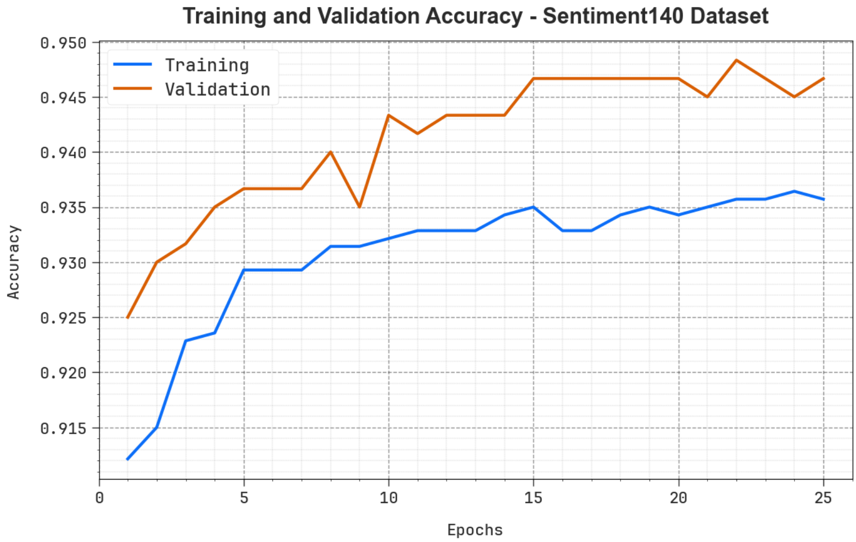

Fig. 5. $Accu_y$ curve of SASM-LOADE technique on Sentiment140 dataset

In Fig. 6, the training and validation loss graph of the SASM-LOADE approach on Sentiment140 dataset is exhibited. The loss values are computed over an interval of 0-25 epochs. It is denoted that the training and validation accuracy values demonstrate a declining tendency, reporting the capability which reported the skill of the SASM-LOADE method in balancing a trade-off among data fitting and generalization. The continual decrease in loss values also pledges the superior performance of the SASM-LOADE approach and tunes the prediction outcomes over time.

Fig. 6. Loss curve of SASM-LOADE technique on Sentiment140 database

Table 3 and Fig. 7 show the overall comparison results of the SASM-LOADE model on the Sentiment140 dataset [23]. The outcomes show that the TF-RNN and WV-CNN techniques have shown reduced performance. However, the TF-DNN, TF-CNN, WV-DNN, and WV-RNN techniques have revealed reasonable results. Although the ASASM-HHODL system has managed to extend considerable performance, the SASM-LOADE model acquires better performance with maximum $accu_y$ of 94.83%, $prec_n$ of 95.10%, $reca_l$ of 94.83%, and F_{score} of 94.83%.

Table 3 Comparative outcome of SASM-LOADE system with recent models on Sentiment140 dataset

Sentiment140 Dataset				
Methods	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}
TF-DNN	78.05	75.10	82.25	77.24
TF-CNN	82.8	79.59	83.48	75.94
TF-RNN	75.05	80.18	81.95	83.32
WV-DNN	81.93	79.95	76.47	83.48
WV-CNN	79.72	82.63	83.77	79.71
WV-RNN	80.98	75.65	78.45	75.98
ASASM-HHODL	84.25	85.83	86.37	86.13
SASM-LOADE	94.83	95.10	94.83	94.83

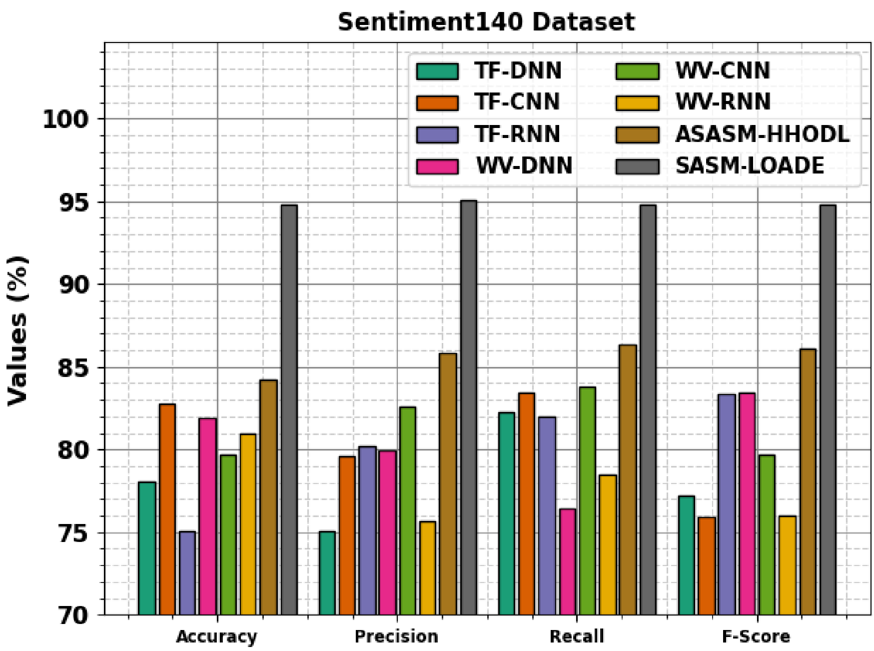

Fig. 7. Comparative outcome of SASM-LOADE technique on Sentiment140 dataset

Table 4 and Fig. 8 show the overall computational time (CT) results of the SASM-LOADE method on the Sentiment140 dataset. The outcomes specify that the TF-RNN and WV-CNN approaches have shown worst performance. At the same time, the TF-DNN, TF-CNN, WV-DNN, and WV-RNN techniques have revealed reasonable outcomes. Although the ASASM-HHODL approach has managed to extend considerable performance, the SASM-LOADE system gains enhanced performance with smaller CT of 0.71s.

Table 4 CT outcome of SASM-LOADE technique with recent models on Sentiment140 dataset

Sentiment140 Dataset	
Methods	Computational Time (sec)
TF-DNN	2.20
TF-CNN	3.19
TF-RNN	2.65
WV-DNN	2.81
WV-CNN	3.81
WV-RNN	2.99
ASASM-HHODL	4.19
SASM-LOADE	0.71

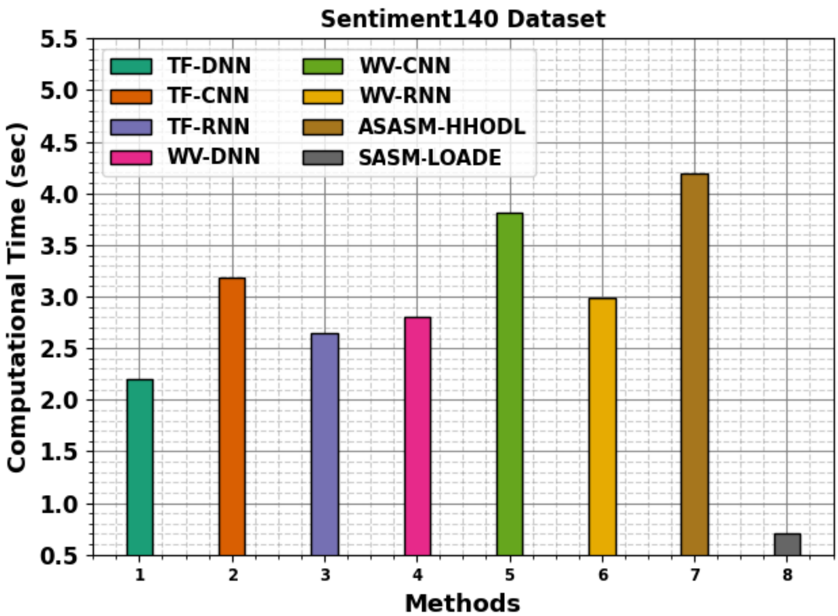

Fig. 8. CT outcome of SASM-LOADE technique on Sentiment140 dataset

Fig. 9 exhibits the classifier results of the SASM-LOADE method at Airlines dataset. Figs. 9a-9b describes the confusion matrices presented by the SASM-LOADE system on 70:30 of TRAS/TESS. The outcome signified that the SASM-LOADE method has known and classified all 2 class labels correctly.Also, Fig. 9c examines the PR outcome of the SASM-

LOADE approach. The figure conveyed that the SASM-LOADE method has attained maximum PR performance under every class. At last, Fig. 9d determines the ROC investigation of the SASM-LOADE method. The outcome portrayed that the SASM-LOADE system has resulted in proficient outcomes with maximum values of ROC under diverse classes.

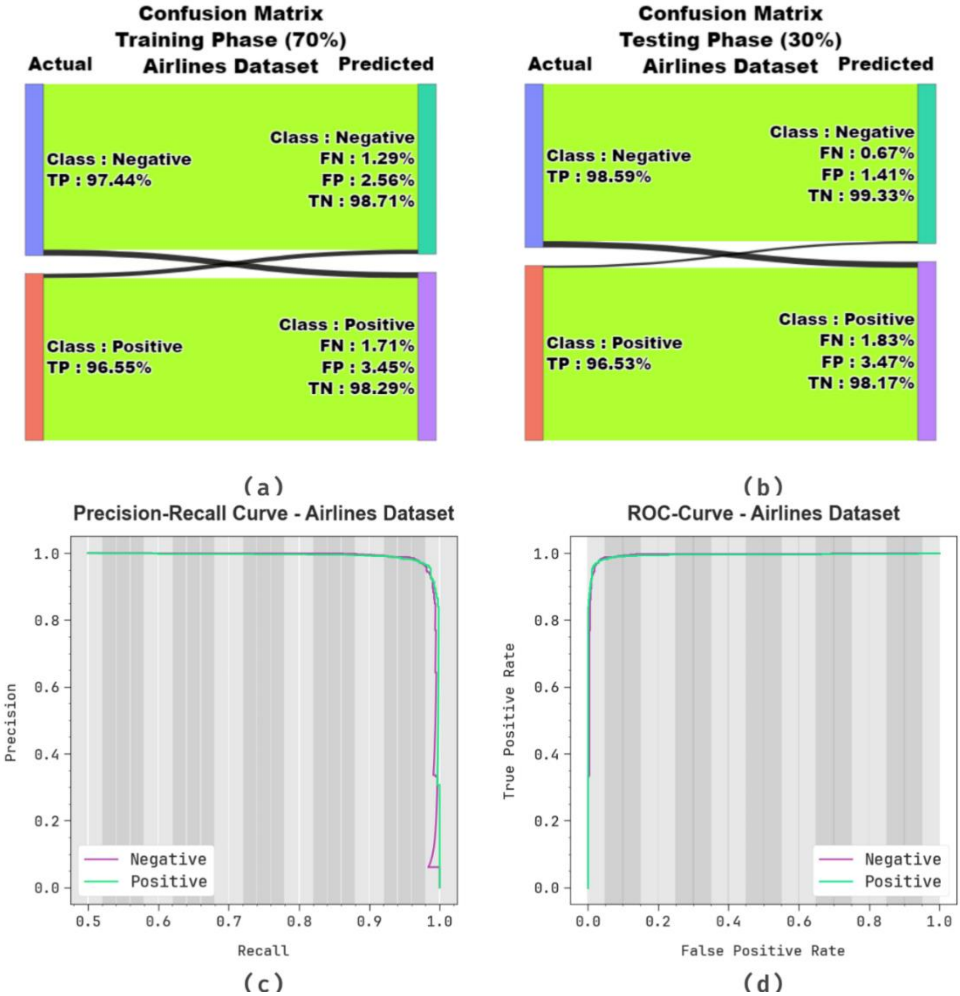

Fig. 9. Airlines dataset (a-b) confusion matrices and (c-d) PR and ROC curves

Table 5 and Fig. 10 highlight the complete sentiment recognition outcomes of the SASM-LOADE system on the airline dataset. The results denoted that the SASM-LOADE system properly recognized negative and positive samples. On 70%TRAS, the SASM-LOADE method provides average $accu_y$ of 97.01%, $prec_n$ of 97.00%, $reca_l$ of 97.01%, F_{score} of 97.00%, and MCC of 94.00%. Furthermore, on 30%TESS, the SASM-LOADE methodology gets average $accu_y$ of 97.46%, $prec_n$ of 97.56%, $reca_l$ of 97.46%, F_{score} of 97.49%, and MCC of 95.02%.

Table 5 Sentiment recognition outcome of SASM-LOADE technique on Airlines database

Class	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}	MCC
TRAS (70%)					
Negative	96.62	97.44	96.62	97.03	94.00
Positive	97.39	96.55	97.39	96.97	94.00
Average	97.01	97.00	97.01	97.00	94.00
TESS (30%)					
Negative	96.21	98.59	96.21	97.38	95.02
Positive	98.71	96.53	98.71	97.61	95.02
Average	97.46	97.56	97.46	97.49	95.02

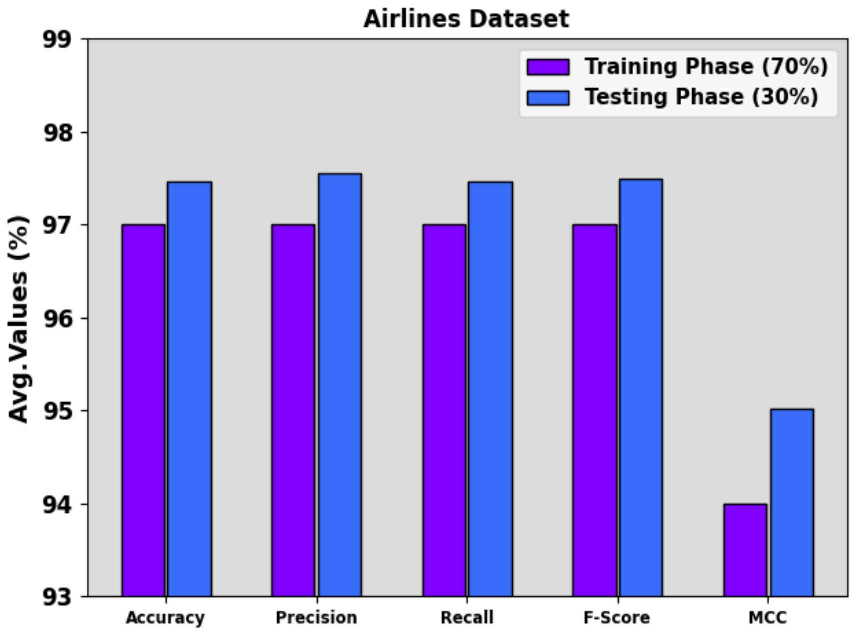

Fig. 10. Average outcome of SASM-LOADE technique on Airlines dataset

In Fig. 11, the training and validation accuracy outcomes of the SASM-LOADE approach on Airlines dataset are revealed. The accuracy values are calculated over an interval of 0-25 epochs. The outcome indicated that the training and validation accuracy values show a rising trend which alerted the capability of the SASM-LOADE system with enhanced performance over numerous iterations. Moreover, the training accuracy and validation accuracy remain closer over the epochs, which specifies low least overfitting and displays boosted performance of the SASM-LOADE system, guaranteeing reliable forecasts on unseen samples.

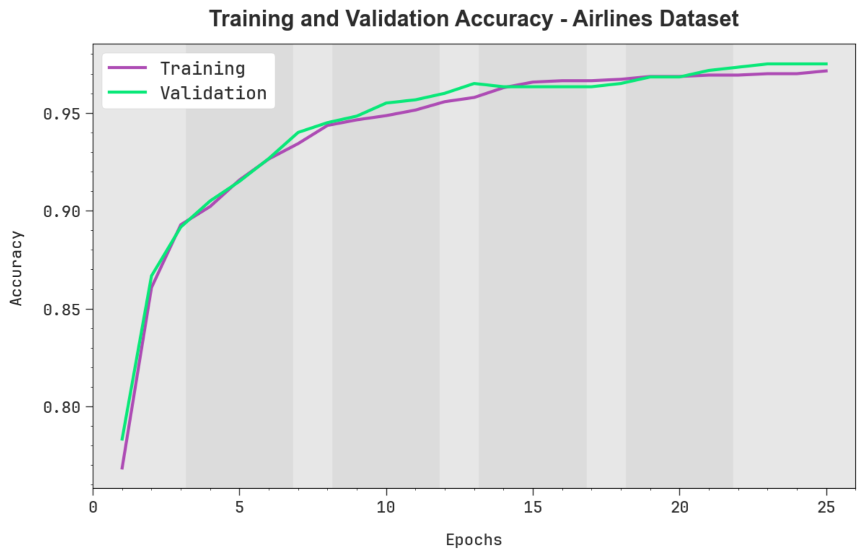

Fig. 11. $Accu_y$ curve of SASM-LOADE technique on Airlines dataset

Fig. 12. Loss curve of SASM-LOADE technique on Airlines dataset

In Fig. 12, the training and validation loss graph of the SASM-LOADE system on Airlines dataset is shown. The loss values are calculated over an interval of 0-25 epochs. It is epitomized that the training and validation accuracy values demonstrate a decreasing tendency, alerting the ability which advised the skill of the SASM-LOADE approach in balancing a trade-off amongst data fitting and generalization. The constant decrease in loss values furthermore assurances the higher performance of the SASM-LOADE methodology and tunes the prediction outcomes over time.

Table 6 and Fig. 13 direct the complete comparison outcomes of the SASM-LOADE method on the Airlines dataset. The outcomes specify that the TF-RNN and WV-CNN approaches have shown decreased performance. While, the TF-DNN, TF-CNN, WV-DNN, and WV-RNN methodologies have proven reasonable results. While the ASASM-HHODL system has achieved considerable performance, the SASM-LOADE method gains better performance with maximum $accu_y$ of 97.46%, $prec_n$ of 97.56%, $reca_l$ of 97.46%, and F_{score} of 97.49%.

Table 6 Comparative outcome of SASM-LOADE technique with recent models on Airlines database

Airlines Dataset				
Methods	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}
TF-DNN	93.77	89.67	95.08	86.87
TF-CNN	92.38	85.17	93.25	90.00
TF-RNN	93.74	87.87	90.35	86.48
WV-DNN	88.00	94.03	92.93	91.27
WV-CNN	89.83	93.77	94.24	91.96
WV-RNN	90.69	85.72	92.71	93.57
ASASM-HHODL	95.50	95.19	95.36	94.33
SASM-LOADE	97.46	97.56	97.46	97.49

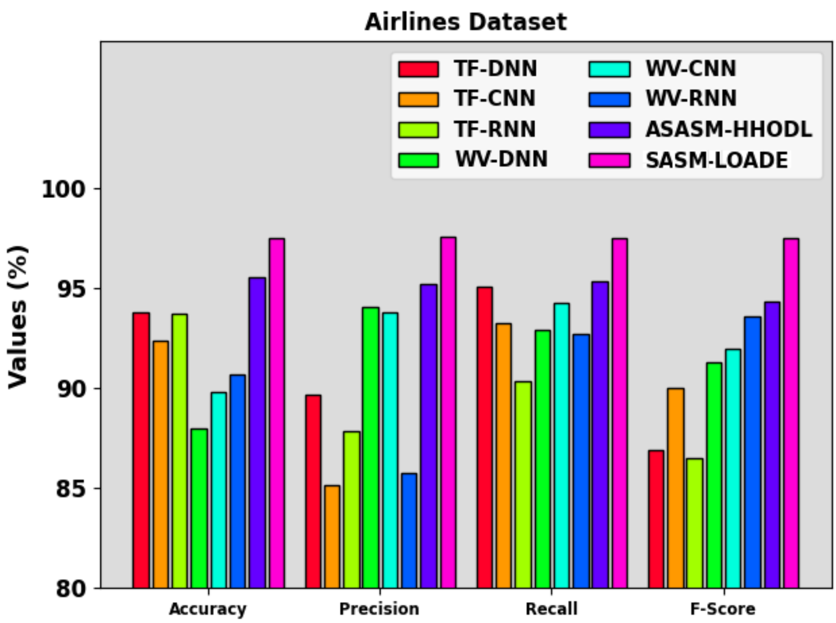

Fig. 13. Comparative outcome of SASM-LOADE technique on Airlines database

Table 7 and Fig. 14 show the overall CT results of the SASM-LOADE system on the airline dataset. The results specify that the TF-RNN and WV-CNN systems have presented worst performance. Simultaneously, the TF-DNN, TF-CNN, WV-DNN, and WV-RNN approaches have revealed reasonable outcomes. Although the ASASM-HHODL system has achieved to range of considerable performance, the SASM-LOADE model gets improved performance with smaller CT of 1.11s.

Table 7 CT outcome of SASM-LOADE technique with recent models on Airlines dataset

Airlines Dataset	
Methods	Computational Time (sec)
TF-DNN	6.19
TF-CNN	5.39
TF-RNN	3.83
WV-DNN	3.67
WV-CNN	6.53
WV-RNN	4.39
ASASM-HHODL	5.62
SASM-LOADE	1.11

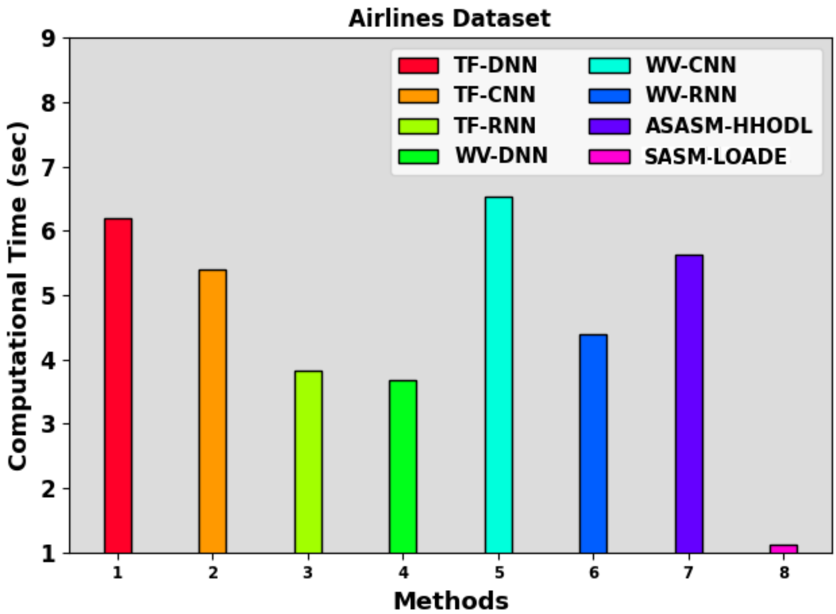

Fig. 14. CT outcome of SASM-LOADE technique on Airlines dataset

Thus, the SASM-LOADE technique has the enhanced ability to identify sentiments on social media.

5. Conclusion

In this study, we concentrate on the design and development of SASM-LOADE technique. The goal of the SASM-LOADE technique is to recognize different kinds of sentiments that exist in the social media data. Primarily, the SASM-LOADE technique takes place preprocessing involves different sub-processes. In addition, the feature extraction using TF-IDF model takes place which generates word embedding. Next, the detection of sentiments can be performed by the use of ensemble model containing3 classifiers such as GRU, RNN, and BiLSTM model. For hyperparameter tuning of the DL models, the LOA is applied in the SASM-LOADE technique. In order to demonstrate the significance of the SASM-LOADE technique, a widespread simulation analysis is performed. The experimental values illustrate that the SASM-LOADE technique reaches promising results over existing models.

References

1. Rangarajan, P.K., Gurusamy, B.M., Muthurasu, G., Mohan, R., Pallavi, G., Vijayakumar, S. and Altalbe, A., 2024. The social media sentiment analysis framework: deep learning for sentiment analysis on social media. International Journal of Electrical and Computer Engineering (IJECE), 14(3), pp.1-1x.
2. Derbentsev, V.D., Bezkorovainyi, V.S., Matviychuk, A.V., Pomazun, O.M., Hrabariev, A.V. and Hostryk, A.M., 2023. Sentiment Analysis of Electronic Social Media Based on Deep Learning. no. M3e2, pp.163-175.
3. Chandrasekaran, G., Antoanela, N., Andrei, G., Monica, C. and Hemanth, J., 2022. Visual sentiment analysis using deep learning models with social media data. Applied Sciences, 12(3), p.1030.
4. Meena, Y.R., Shrivastav, A.K. and Kharbas, V.K., 2024, March. Applying Deep Neural Networks to Sentiment Analysis in Social Media. In 2024 3rd International Conference for Innovation in Technology (INOCON) (pp. 1-5). IEEE.
5. Merayo, N., Vegas, J., Llamas, C. and Fernández, P., 2023. Social Network Sentiment Analysis Using Hybrid Deep Learning Models. Applied Sciences, 13(20), p.11608.
6. Wu, Y., Jin, Z., Shi, C., Liang, P. and Zhan, T., 2024. Research on the Application of Deep Learning-based BERT Model in Sentiment Analysis. arXiv preprint arXiv:2403.08217.
7. Rodrigues, A.P., Fernandes, R., Shetty, A., K, A., Lakshmanna, K. and Shafi, R.M., 2022. [Retracted] Real-Time Twitter Spam Detection and Sentiment Analysis using Machine Learning and Deep Learning Techniques. Computational Intelligence and Neuroscience, 2022(1), p.5211949.
8. Priya, C.S.R. and Deepalakshmi, P., 2023. Sentiment analysis from unstructured hotel reviews data in social network using deep learning techniques. International Journal of Information Technology, 15(7), pp.3563-3574.
9. Mirugwe, A., Ashaba, C., Namale, A., Akello, E., Bichetero, E., Kansiime, E. and Nyirenda, J., 2024. Sentiment Analysis of Social Media Data on Ebola Outbreak Using Deep Learning Classifiers. Life, 14(6), p.708.
10. Dominic, P., Purushothaman, N., Kumar, A.S.A., Prabagaran, A., Blessy, J.A. and John, A., 2023, January. Multilingual sentiment analysis using deep-learning architectures. In 2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT) (pp. 1077-1083). IEEE.
11. Brinda, B.M., Rajan, C., Geetha, K., Nathiya, N., Raguraman, P.J. and Srinivasan, K., 2024, March. Applying Deep Neural Networks and NLP Techniques for Sentiment Analysis in Social

Media Data. In 2024 2nd International Conference on Artificial Intelligence and Machine Learning Applications Theme: Healthcare and Internet of Things (AIMLA) (pp. 1-9). IEEE.

12. Chiranjeevi, P. and Rajaram, A., 2023. A lightweight deep learning model based recommender system by sentiment analysis. Journal of Intelligent & Fuzzy Systems, (Preprint), pp.1-14.
13. Khan, L., Amjad, A., Afaq, K.M. and Chang, H.T., 2022. Deep sentiment analysis using CNN-LSTM architecture of English and Roman Urdu text shared in social media. Applied Sciences, 12(5), p.2694.
14. Paulraj, D., Ezhumalai, P. and Prakash, M., 2024. A Deep Learning Modified Neural Network (DLMNN) based proficient sentiment analysis technique on Twitter data. Journal of Experimental & Theoretical Artificial Intelligence, 36(3), pp.415-434.
15. Kora, R. and Mohammed, A., 2023. An enhanced approach for sentiment analysis based on meta-ensemble deep learning. Social Network Analysis and Mining, 13(1), p.38.
16. Singh, N.K., Agal, S., Gadekallu, T.R., Shabaz, M., Keshta, I., Jindal, L., Soni, M., Byeon, H. and Singh, P.P., 2024. Deep Learning Model for Interpretability and Explainability of Aspect-Level Sentiment Analysis Based on Social Media. IEEE Transactions on Computational Social Systems.
17. Adam, A.Z.R. and Setiawan, E.B., 2023. Social Media Sentiment Analysis using Convolutional Neural Network (CNN) dan Gated Recurrent Unit (GRU). JurnalIlmiah Teknik ElektroKomputer dan Informatika (JITEKI), 9(1), pp.119-131.
18. Derbentsev, V.D., Bezkorovainyi, V.S., Matviychuk, A.V., Pomazun, O.M., Hrabariev, A.V. and Hostryk, A.M., 2023. A comparative study of deep learning models for sentiment analysis of social media texts. In CEUR Workshop Proceedings (pp. 168-188).
19. Peng, S., Cao, L., Zhou, Y., Ouyang, Z., Yang, A., Li, X., Jia, W. and Yu, S., 2022. A survey on deep learning for textual emotion analysis in social networks. Digital Communications and Networks, 8(5), pp.745-762.
20. Ra’ed, M., Al-qudah, N.E.A., Jawarneh, M.S. and Al-Khateeb, A., 2023. A novel improved lemurs optimization algorithm for feature selection problems. Journal of King Saud University-Computer and Information Sciences, 35(8), p.101704.
21. https://www.kaggle.com/datasets/kazanova/sentiment140
22. https://www.kaggle.com/datasets/crowdflower/twitter-airline-sentiment
23. Halawani, H.T., Mashraqi, A.M., Badr, S.K. and Alkhalaf, S., 2023. Automated sentiment analysis in social media using Harris Hawks optimisation and deep learning techniques. Alexandria Engineering Journal, 80, pp.433-443.