International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(3s) 2026 pp. 171-181
https://doi.org/10.70917/ijcisim-2026-2321
Article

Performance analysis using deep learning approaches for sentiment classification of tweets

Geetha V[1], Sujatha N[2]

[1]Department of Computer Science, Lady Doak College, Madurai Kamaraj University, Madurai Tamil Nadu, India, geetha@ldc.edu.in
[2]P.G & Research Department of Computer Science, Sri Meenakshi Govt. Arts College for Women (Autonomous), Madurai Kamaraj University, Madurai, Tamil Nadu, India, sujamurugan@gmail.com

Abstract: Social media refers to a collection of websites and programs that enable users to express and disseminate their opinions among diverse communities. For instance, popular social media platforms include Facebook, Instagram, and X (Twitter). Instantaneous communication, text, video, and idea sharing is possible with people all over the world. However, overuse can have detrimental effects like mental health issues, decreased productivity, and cyberbullying. In the end, social media is a great tool for networking, self-expression, and international connection when utilized appropriately. A text mining technique called sentiment analysis is used to identify and categorize the emotional tone of textual material. Predicting a sentence's sentiment—whether good, negative, or neutral—assists companies in improving the quality of their services through feedback, reviews, and social media comments. This research article aims to sort of recurrent neural network called a long short-term memory (LSTM) is specifically made to keep the neural network output for a particular input from either exploding or fading as it cycles through the feedback loops. An embedding layer, a single LSTM layer, and a dense layer at the end make up this architecture. To reduce over-fitting, dropout methods are used in between the LSTM layers.
Keywords: Social Media Networks, Twitter, Sentiment Analysis, Deep Learning, LSTM.

1. INTRODUCTION

Social media is commonly utilized to quickly and simply spread any type of information at any time [1]. In today's networking society, most people use one of the social media networking platforms. For example, there are millions of active users on Facebook and Twitter. Both are used as strong tools for practically instantaneous message distribution to a broad variety of audiences [2]. Twitter is a major social media network that incorporates public opinion, therefore the content is tough for computers to analyze. People's opinions about a given service, product, or situation can be acquired by processing and analyzing data from Twitter [3]. Twitter is a well-known microblog where we may categorize tweets according to their attitude, which includes neutral, negative, and positive [4]. A few applications of deep learning are given below [5]:

Table 1: Applications of deep learning models

Model Name	Problem	Description
CNN	Image Classification	In CNN, each neuron receives some inputs, does a dot product, and optionally applies non-linear activations. Every pixel in the image is assigned a value between 0 and 255. It takes considerable preprocessing to identify patterns that distinguish one image from another.
3D-CNN	Visual Speech Recognition	The technique involves the classification of visemes in texts. The first module extracts lip characteristics from the input, while the

		second module trains a neural network system to analyze lip visemes and identify them as text.
RNN	Stock Market Prediction	Predicting stock prices is an essential task. Analysis entails properly predicting the future closing values of a stock using historical data.
RNN-LSTM	Next Word Prediction	The task at hand is word prediction. This issue aids in lowering the keystroke count.
CRNN	Tamil Handwritten character optical recognition	It is possible to digitize historical Tamil-language files from several organizations.

2. LITERATURE REVIEW

The goal of sentiment analysis (SA) is to extract and identify people's opinions from the text. Over time, machine learning-based techniques have been used to achieve this. However, due to their improved performance, deep learning-based methods have recently taken the lead [6]. Convolutional Neural Networks (CNNs), Google BERT, and attention-based bidirectional LSTM were modified for the sentiment classification of tweets [7]. The effectiveness of transformer-based models, such as BERT and RoBERTa, is greatly impacted by the type of dataset, pre-processing techniques, and application domain. These models have strong abilities to handle complex linguistic patterns and achieve high performance in sentiment classification tasks [8]. Sentiment dictionary, BERT model, CNN model, BiGRU model, and attention mechanism were used to build the model SLCABG for sentiment analysis on product reviews. In the first step, the sentiment lexicon was used to improve the sentiment features in the reviews. In the second step, the primary sentimental and contextual characteristics of the reviews were extracted using CNN and GRU networks, and they were then weighted using an attention mechanism [9]. To demonstrate how to attain high emotion classification accuracy, the experiment was carried out using RNN and LSTM models on three distinct datasets. With 88.47% accuracy for positive/negative classification and 89.13% and 91.3% accuracy for positive and negative subclass, respectively, the final system's emotion prediction on the LSTM model [10]. With scores of 94.9%, 91.77%, and 89.81% on the IMDb, Twitter US Airline Sentiment dataset, and Sentiment140 dataset, respectively, the experimental results show that the ensemble hybrid deep learning model with majority voting outperforms all other approaches [11]. The experimental findings demonstrate that, on three French datasets, the combination of GRU and CNN with XLNet appears to improve upon the state of the art: 1) French Amazon Customer Reviews, 2) AlloCị Dataset, and 3) French Twitter Sentiment Analysis; the corresponding accuracy values are 96.5%, 90.1%, and 89.6% [12]. The suggested model had an accuracy of 87%. The higher accuracy of the suggested hybrid model when compared to basic deep learning models shows that it is superior for sentiment analysis. The methodology made it easier to categorize both positive and negative viewpoints seen in film reviews [13]. The study combines a hybrid deep learning model that combines Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks with sophisticated feature extraction methods, including Word2Vec. With Word2Vec characteristics added, the suggested model obtains 92% accuracy, 91% precision, 89% recall, and 90% F1 score, demonstrating how well it captures the dynamic nature of Twitter data. These results showed the hybrid model's higher robustness and predictive performance [14]. About 1280000 tweets were used to train the full model, and 320000 tweets were used for testing. During training, cross validation was used, and the accuracy was approximately 80.99%. Prediction accuracy is significantly lower for the same data parameters using alternative models, such as CNN and RNN. CNN model accuracy is 58.69%, whereas RNN model accuracy is 77.73% [15].

3. METHODOLOGY

The work flow of the suggested methodology is shown in the following figure:

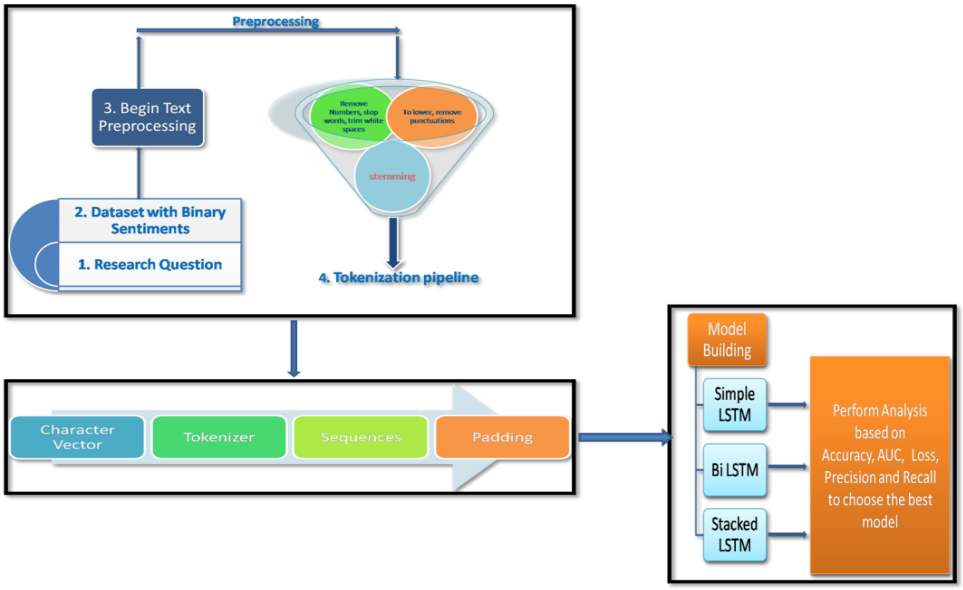


Fig 1 : Architecture for the Proposed Methodology

3.1 Procedure to working on the models

Procedure: to apply deep learning models in sentiment classification

1. Install Packages
2. Load Dataset
3. Preprocessing text (to lower, remove numbers, remove punctuation, stop words, trim white space)
4. Tokenization pipeline is as follows:

   character vector
   ↓
   tokenizer
   ↓
   sequences
   ↓
   padding
   ↓
   Apply model
5. Printing sample records
6. Standardizing input length by padding and truncating
7. Build simple LSTM sentiment classifier architecture

   Proceed to step 10
8. Build Bidirectional LSTMs sentiment classifier architecture

   Proceed to step 10
9. Build Stacked Recurrent sentiment classifier architecture

   Proceed to step 10
10. Steps to follow:
    1. build the model
    2. summary of the model
    3. compilation of the model

For binary sentiment classification:

$$\hat{y}=\sigma(W_x+b) \quad (1)$$

-where sigmoid outputs probabilities between 0 and 1.

4. fitting model
5. predicting and evaluating for test data
6. plot the results

11. Choosing the best model for the chosen data by comparing measures

3.2 About the dataset

The Twitter_data.csv dataset was utilized for this research article [16]. There were 162980 x 2 records with the clean_text and category characteristics. In order to properly evaluate text data, the "0" (55205) labeled and "1" (72242) labeled opinions were taken into account, resulting in a total of 1,27,447 entries for this research paper. The sample two records are shown in the following figure:

A data.frame: 2 × 2

	clean_text	category
	<chr>	<int>
2	talk all the nonsense and continue all the drama will vote for modi	0
3	what did just say vote for modi welcome bjp told you rahul the main campaigner for modi think modi should just relax	1

Fig 2 : Sample records from the dataset

3.3 Text Preprocessing

The R packages tidyverse, tm, and caret were used for text processing. Text preprocessing includes stopwords, stemDocument, removePunctuation, removeNumbers, stripWhitspace, and Tolower. Following preprocessing, the two records are displayed below:

A data.frame: 2 × 2

	clean_text	category
	<named list>	<int>
2	talk nonsense continue drama will vote modi	0
3	just say vote modi welcome bjp told rahul main campaigner modi think modi just relax	1

Fig 3 : Sample records after text pre-processing

3.4 *Split Dataset and Training the model*

The data set was split into 80:20 ratios for training and testing, respectively. The text data sample is as follows:

Data Split (80:20) :

chr [1:101958] "talk nonsense continue drama will vote modi"- (train text)

num [1:101958] 0 1 1 0 0 1 1 1 0 0 - (train label)

The model was trained using 80% of the data (101,958 records), with the remaining 20% set aside for testing and evaluation. Tokenization was used to transform textual content into a machine-readable numerical representation once the data had been separated. The most common words in the corpus were chosen for tokenization, with a maximum vocabulary size of 5000 words. The text-to-sequences method was then used to convert each text document into an integer sequence, with each distinct word given a numerical index. Padding was used with a maximum sequence length of 50 to guarantee consistent input size for the deep learning model. In order

to provide fixed-length input vectors appropriate for models like Simple LSTM, Bidirectional LSTM, and stacked LSTM, shorter sequences were padded with zeros while larger sequences were trimmed. The following is an example of the resulting matrix:

A matrix: 3 × 50 of type int																				
0	0	0	0	0	0	0	0	0	0	⋯	0	0	0	228	1056	595	743	3	11	1
0	0	0	0	0	0	0	0	0	0	⋯	4	423	22	363	3308	1	54	1	17	3559
0	0	0	0	0	0	0	0	0	0	⋯	304	726	637	71	75	66	695	3399	1	82

Fig 4 : Sample machine readable(numerical) form of text data

3.4.1 Simple LSTM

An embedding layer, an LSTM layer, a dropout layer, and a dense output layer make up the suggested LSTM-based sentiment classification model. By converting input words into 64-dimensional dense vector representations, the embedding layer lowers the sparsity of textual data while allowing the model to learn semantic associations between words. An LSTM layer with 32 units then processes these word embeddings, capturing long-term relationships and sequential patterns in the text, making it appropriate for comprehending the contextual meaning of tweets. During training, a Dropout layer with a rate of 0.5 randomly deactivates 50% of the neurons to enhance the model's capacity for generalization and avoid overfitting. Finally, for binary sentiment classification, a Dense layer with a single neuron and a sigmoid activation function generates a probability value between 0 and 1, indicating whether the tweet represents positive or negative sentiment. This architecture maintains good performance on sentiment analysis tasks while striking an efficient balance between computational efficiency and model complexity. The architecture and parameters of this model are depicted in Figure 5 and Figure 6 respectively.

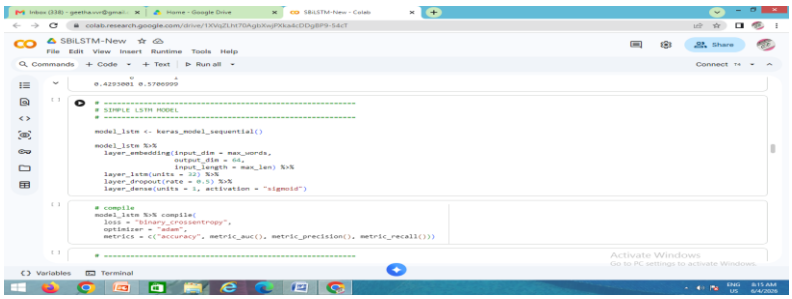


Fig 5 : Simple LSTM sentiment classifier architecture

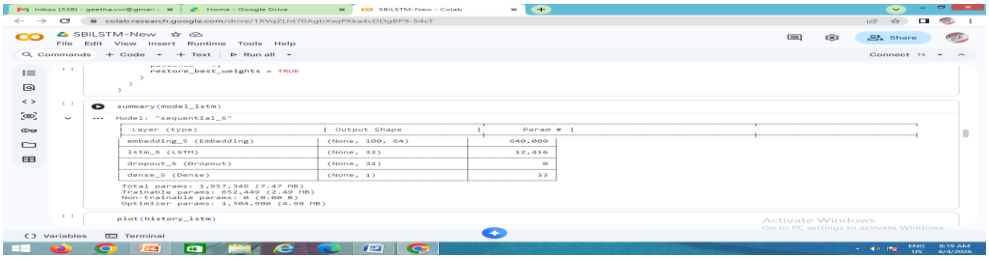


Fig 6 : Simple LSTM sentiment classifier parameters

The training and validation performance of the basic LSTM over several epochs is shown in Figure 7. The model is learning efficiently without any over-fitting, as seen by the training accuracy's steady growth from roughly 0.92 to 0.96 and the validation accuracy's constant high level of 0.95. Strong classification capacity is also demonstrated by the AUC values for both training and validation, which stay at 0.97–0.98. While the validation loss stays comparatively constant, indicating strong generalization performance, the loss curve displays a continual drop in training loss over epochs. The model's capacity to accurately identify positive samples with fewer false predictions is demonstrated by the continuous improvement of precision and recall measures during training. Overall, the graph shows that on both training and validation datasets, the model provides consistent and dependable performance with balanced learning behavior.

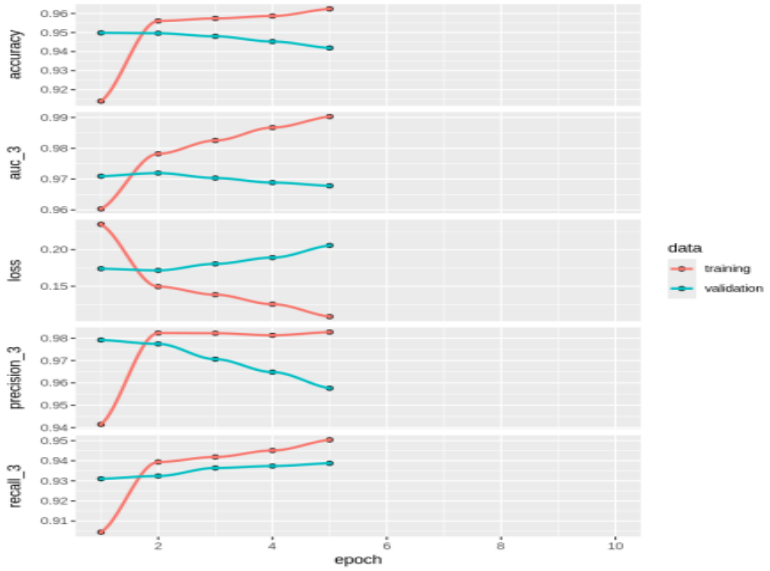


Fig 7 : Simple LSTM sentiment classifier performance

3.4.2 Bidirectional LSTM

The suggested Bidirectional LSTM (BiLSTM) architecture is quite successful for sentiment classification tasks since it is made to capture contextual information from both past and future words in a text sequence. The model starts with an Embedding layer that allows the network to learn semantic links between words by converting input word indices into 128-dimensional dense vector representations. After that, a 128-unit Bidirectional LSTM layer processes these embeddings by concurrently analyzing the sequence in both forward and backward directions. By taking into account both the preceding and succeeding words in a sentence, this bidirectional processing enables the model to comprehend context more fully. After that, a Dropout layer with a rate of 0.3 is used to improve generalization by randomly deactivating some neurons during training in order to lessen over-fitting. Lastly, for binary sentiment classification, a Dense output layer with a single neuron and a sigmoid activation function produces a probability score between 0 and 1 that indicates whether a tweet conveys positive or negative sentiment. The intricate linguistic patterns and contextual relationships found in social media text are especially well captured by this design. The architecture and parameters of this model are depicted in Figure 8 and Figure 9 respectively.

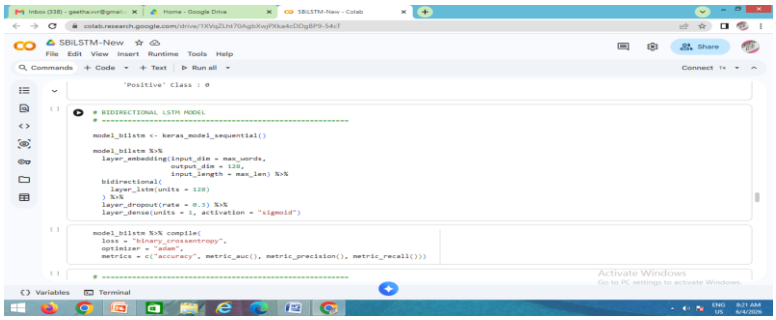


Fig 8 : Bidirectional LSTM sentiment classifier architecture

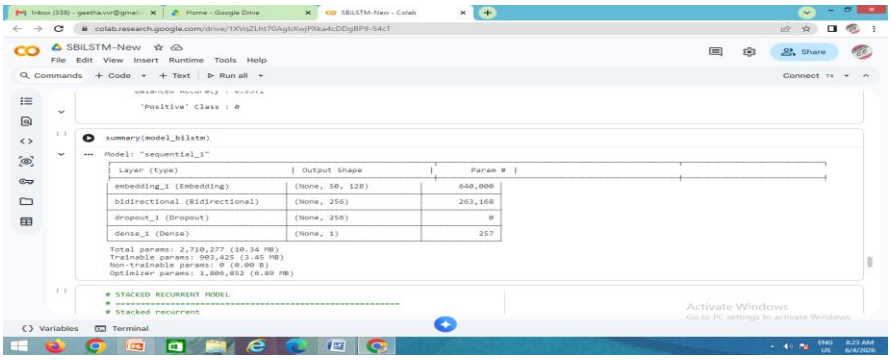


Fig 9 : Bidirectional LSTM sentiment classifier parameters

The deep learning model's training and validation performance throughout several epochs is shown in Figure 10. Effective learning and strong model stability are indicated by the training accuracy's steady growth from 0.91 to 0.96 and the validation accuracy's stability around 0.95. Both datasets' AUC values remain above 0.96, indicating a high degree of ability to differentiate between the target classes. The model appears to be converging correctly without experiencing severe over-fitting, as evidenced by the loss curve's progressive decrease in training loss and comparatively controlled validation loss. While recall gradually increases over epochs, demonstrating an improved capacity to properly identify pertinent instances, precision values constantly remain high for training and validation data, indicating accurate positive predictions. Overall, the model performs in a balanced and dependable manner and has a great capacity to generalize to new data.

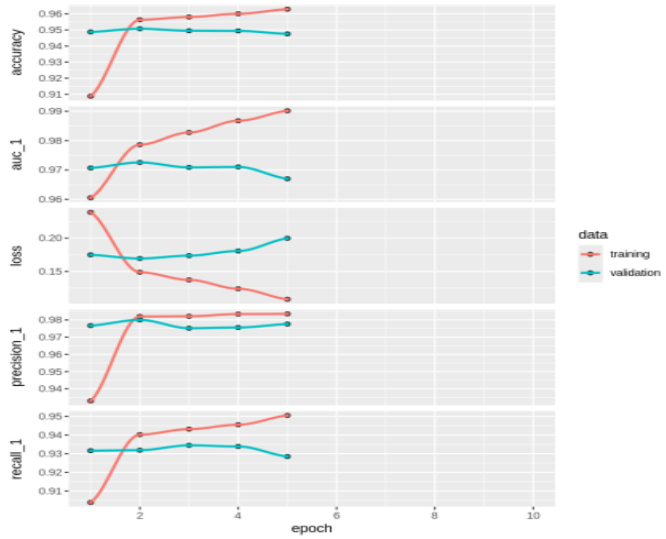


Fig 10: Bidirectional LSTM sentiment classifier performance

3.4.3 Stacked LSTM

An embedding layer, two LSTM layers, and a dense output layer make up the suggested stacked LSTM architecture for binary sentiment categorization. The Embedding layer reduces the sparsity of textual data while capturing semantic links between words by converting each word index into a 128-dimensional dense vector representation. The first LSTM layer can send the entire sequence of learnt temporal characteristics to the subsequent LSTM layer because it has 128 memory units and is set up with return_sequences = TRUE. Only the final hidden state (return_sequences = FALSE) is output by the second LSTM layer, which has 64 units and further extracts long-term dependencies and higher-level sequential patterns from the text. Lastly, a sigmoid activation function and a single neuron in a dense layer generate a probability value between 0 and 1 that indicates whether the input tweet falls into the positive or negative sentiment class. The model's capacity to capture intricate contextual information and hierarchical linguistic patterns is enhanced by this stacked architecture, resulting in more precise sentiment classification. The architecture and parameters of this model are depicted in Figure 11 and Figure 12 respectively.

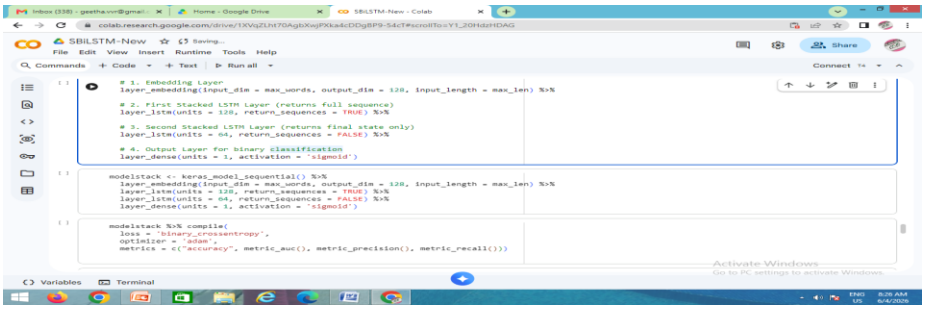


Fig 11 : Stacked LSTM sentiment classifier architecture

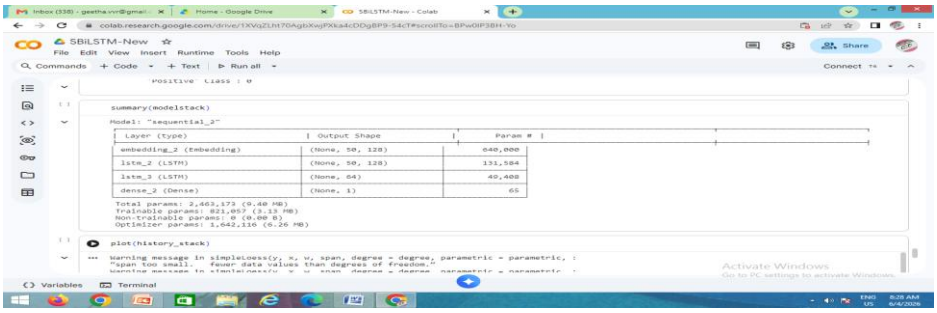


Fig 12: Stacked LSTM sentiment classifier parameters

The deep learning model's training and validation performance over several epochs is shown in Figure 13 utilizing important assessment criteria. While the validation accuracy first increases and then stays mostly consistent, the training accuracy shows a steady increase from about 0.91 to above 0.96, suggesting effective learning during the training procedure. Although the validation AUC gradually declines in later epochs, indicating the start of modest over-fitting, the AUC values show high classification performance. The model may begin memorizing the training data rather than fully generalizing, as the loss curve shows a steady decrease in training loss while the validation loss progressively rises after specific epochs. While validation precision gradually decreases across epochs, indicating less prediction consistency on unseen data, training data precision continually improves. The model's capacity to accurately detect positive instances is demonstrated by the recall values for both training and validation datasets, which stay high and progressively increase. With only minor indications of over-fitting in subsequent epochs, the model often achieves excellent predictive performance.

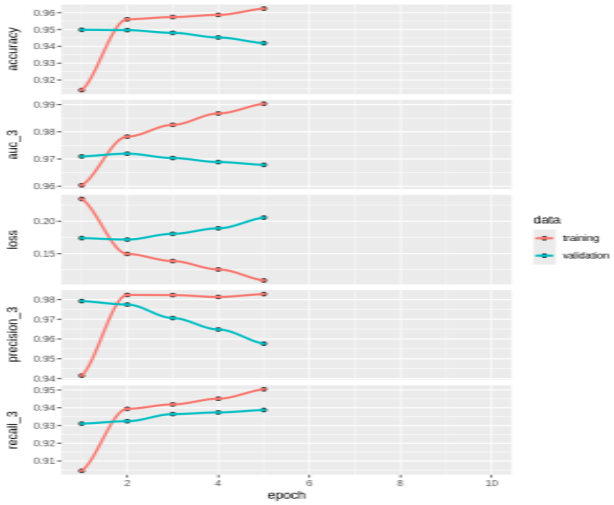


Fig 13 : Stacked LSTM sentiment classifier performance

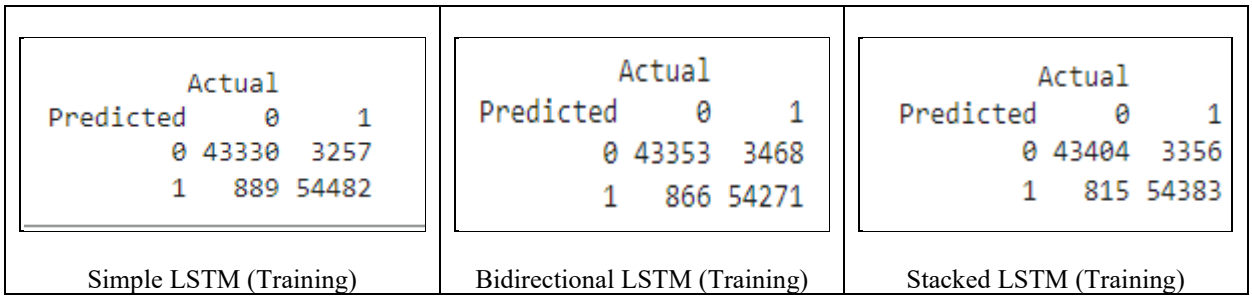
4. RESULTS AND DISCUSSION

The setup and outcomes of the three models during data training are shown in the following table:

Table 2: Results for Train data

Model Name	Max Words	Max Size	Embedding Dim	Epoch	Batch size	Accuracy	AUC	Loss	Precision	Recall
Simple LSTM	5000	50	128	10	128	0.9593	0.9851	0.1339	0.9839	0.9436
BiLSTM	5000	50	128	10	128	0.9574	0.9831	0.1379	0.9842	0.9399
Stacked recurrent	5000	50	128	10	128	0.9590	0.9855	0.1278	0.9852	0.9418

According to the training results, all three deep learning models demonstrated good predictive capacity by achieving high classification performance with accuracy values above 95% and AUC values above 0.98. In comparison to the other models, the BiLSTM produced somewhat lower accuracy and AUC, while the Simple LSTM achieved an accuracy of 0.9593 with acceptable precision and recall values. Among the three models, the Stacked Recurrent model demonstrated superior overall learning and classification performance with the greatest AUC (0.9855), highest accuracy (0.9852), and lowest loss (0.1278). All models worked well overall, while the Stacked Recurrent model produced somewhat better outcomes.



Simple LSTM (Training) | Bidirectional LSTM (Training) | Stacked LSTM (Training)

Fig 14: confusion matrix during training the models

The confusion matrices derived from the training outcomes of three deep learning models—Simple LSTM, Bidirectional LSTM (BiLSTM), and Stacked LSTM—are shown in Figure 14. By contrasting the expected and actual values, a confusion matrix is used to assess categorization performance. The diagonal values in each of the three models show occurrences that were successfully classified, whereas the off-diagonal values show examples that were incorrectly classified. 54,482 positive instances and 43,330 negative instances were accurately classified by the Simple LSTM model, however 3,257 positive samples were incorrectly predicted as negative and 889 negative samples were incorrectly predicted as positive. Strong prediction skill with a large number of true classifications is indicated here. The Bidirectional LSTM model accurately detected 54,271 positive and 43,353 negative cases. False positives decreased to 866, but false negatives slightly increased to 3,468. With just small prediction errors, the model continues to perform well in categorization. Out of the three models, the Stacked LSTM model performed the best overall. In comparison to the other models, it produced the lowest false positive value of 815 and less false negatives, properly classifying 43,404 negative occurrences and 54,383 positive examples. The Stacked LSTM model offers superior learning capacity and more precise predictions on the training dataset, as evidenced by the decreased misclassification values. All three models worked well overall, but the Stacked LSTM demonstrated significantly higher classification accuracy and dependability. The setup and outcomes of the three models during data testing are shown in the following table:

Table 3: Results for test data

Model Name	Max Words	Max Size	Embedding Dim	Epoch	Batch size	Accuracy	AUC	Loss	Precision	Recall
Simple LSTM	5000	50	128	10	128	0.9532	0.9747	0.1615	0.9776	0.9393
BiLSTM	5000	50	128	10	128	0.9546	0.9759	0.1571	0.9802	0.9391
Stacked recurrent	5000	50	128	10	128	0.9547	0.9769	0.1541	0.9804	0.9391

The test data findings show that all three deep learning models performed well and consistently when classifying unknown data. With an accuracy of 95.32% and an AUC of 0.9747, the Simple LSTM model demonstrated strong class discrimination and prediction performance. In comparison to the Simple LSTM, the Bidirectional LSTM model demonstrated superior contextual learning from both forward and backward sequences, with an accuracy of 95.46%, a higher precision of 0.9802, and a smaller loss. With the highest accuracy (95.47%), highest AUC (0.9769), highest precision (0.9804), and lowest loss (0.1541), the Stacked Recurrent model outperformed the other two models. The Stacked Recurrent model showed superior generalization and classification ability on the test dataset, despite the recall values for all models remaining almost same at about 93.9%. All models performed well overall, according to the data, however the Stacked Recurrent model's predictive performance was slightly better.

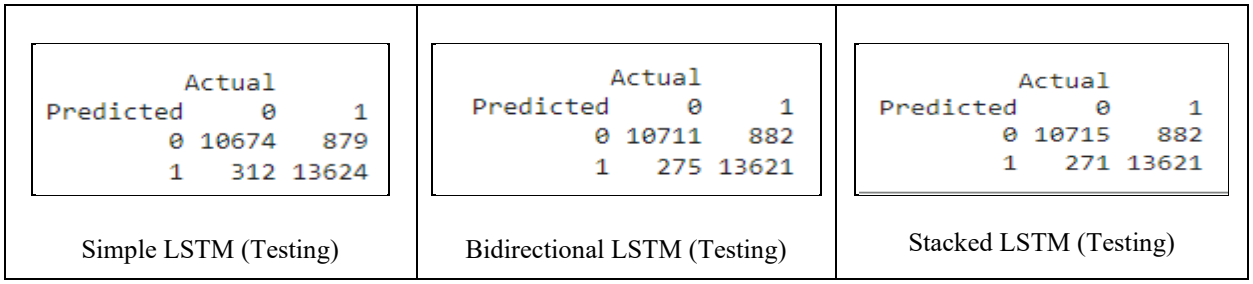


Fig 15: confusion matrix during testing the models

The confusion matrices of the Simple LSTM, Bidirectional LSTM (BiLSTM), and Stacked LSTM models assessed on the testing dataset are shown in Figure 15. By contrasting expected and actual class labels, a confusion matrix aids in assessing model performance. The diagonal values in each matrix show occurrences that were successfully classified, whereas the off-diagonal values show misclassifications like false positives and false negatives. 10,674 negative samples and 13,624 positive samples were accurately categorized using the Simple LSTM model. Nevertheless, 312 negative instances were mistakenly projected as positive (false positives) while 879 positive examples were mistakenly predicted as negative (false negatives). These findings show a high degree of predictive power with few categorization errors. 10,711 negative and 13,621 positive cases were accurately recognized in the Bidirectional LSTM model. While the number of false negatives marginally increased to 882, the number of false positives decreased to 275. This demonstrates how the BiLSTM model maintained good classification accuracy on unknown data while increasing forecast precision. Out of the three models, the Stacked LSTM model performed the best overall during testing. It produced the lowest false positive value of 271 and maintained a low false negative count of 882, properly classifying 10,715 negative instances and 13,621 positive instances. The lower number of misclassifications indicates that the Stacked LSTM model acquired better prediction capabilities and generalized more successfully on the testing dataset. All three models performed well overall, with low error values and high correct classification rates. The Stacked LSTM model outperformed the others in terms of generalization performance, dependability, and classification accuracy on test data that had not yet been seen.

5. CONCLUSION

The experimental findings show that on both training and testing datasets, the three deep learning models—Simple LSTM, Bidirectional LSTM, and Stacked LSTM—performed well for sentiment classification on Twitter with excellent accuracy, AUC, precision, and recall values. Textual input has been transformed into appropriate numerical representations for deep learning models with the aid of preprocessing techniques including tokenization, sequence creation, and padding. With the highest accuracy, AUC, and precision as well as the lowest loss value among the assessed models, the Stacked LSTM demonstrated superior learning capacity and more robust generalization on unknown data. The Stacked LSTM generated the fewest misclassifications when compared to the other models, according to the confusion matrix study. The Stacked LSTM showed substantially better prediction power for twitter sentiment categorization, even though all models displayed steady and dependable performance. As a result, the study comes to the conclusion that deep learning techniques—in particular, the Stacked LSTM architecture—are very successful at managing text-based sentiment analysis tasks and can produce precise(0.9547) classification outcomes for huge textual datasets. In this article, Quillbot.com's paraphrase tool has been utilized to reword the text without changing the author's opinion.

References

1. Zafarani, R., Abbasi, M.A., Liu, H. 2014. Social Media Mining: An Introduction. Cambridge University Press. Cambridge University Press. ISBN 978-1-107-01885-3.
2. Nathan Danneman, Richard Heimann. 2014. Social Media Mining with R. Packt Publishing.
3. Shiyang Liao, Junbo Wang, Ruiyun Yu, Koichi Sato, Zixue Cheng. 2016. CNN for situations understanding based on sentiment analysis of twitter data. 8th International Conference on Advances in Information Technology. IAIT2016. pp. 19-22.
4. Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, Rebecca Passonneau. 2011. Sentiment Analysis of Twitter Data. Proceedings of the Workshop on Language in Social Media (LSM 2011). Pages 30–38.
5. Dr.S.Lovelyn Rose, Dr.L.Ashok Kumar, Dr.D.Karthika Renuka. 2019. “Deep Learning with Python, (1st ed.)”, New Delhi: Wiley India Pvt Ltd, 2019.
6. L. Chaudhary, N. Girdhar, D. Sharma, J. Andreu-Perez, A. Doucet and M. Renz. 2020. "A Review of Deep Learning Models for Twitter Sentiment Analysis: Challenges and Opportunities," in IEEE Transactions on Computational Social Systems, vol. 11, no. 3, pp. 3550-3579, June 2024, doi: 10.1109/TCSS.2023.3322002.
7. A. Roy and M. Ojha.2020. "Twitter sentiment analysis using deep learning models," 2020 IEEE 17th India Council International Conference (INDICON), New Delhi, India, pp. 1-6, doi: 10.1109/INDICON49873.2020.9342279.
8. A. Albladi, M. Islam and C. Seals. 2025. "Sentiment Analysis of Twitter Data Using NLP Models: A Comprehensive Review," in IEEE Access, vol. 13, pp. 30444-30468, doi: 10.1109/ACCESS.2025.3541494.
9. L. Yang, Y. Li, J. Wang and R. S. Sherratt. 2020. "Sentiment Analysis for E-Commerce Product Reviews in Chinese Based on Sentiment Lexicon and Deep Learning," in IEEE Access, vol. 8, pp. 23522-23530, doi: 10.1109/ACCESS.2020.2969854.
10. P. C. Shilpa, R. Shereen, S. Jacob and P. Vinod. 2021. "Sentiment Analysis Using Deep Learning," 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India, pp. 930-937, doi: 10.1109/ICICV50876.2021.9388382.
11. K. L. Tan, C. P. Lee, K. M. Lim and K. S. M. Anbananthen. 2022. "Sentiment Analysis With Ensemble Hybrid Deep Learning Model," in IEEE Access, vol. 10, pp. 103694-103704, doi: 10.1109/ACCESS.2022.3210182.
12. N. Habbat, H. Anoun and L. Hassouni. 2023. "Combination of GRU and CNN Deep Learning Models for Sentiment Analysis on French Customer Reviews Using XLNet Model," in IEEE Engineering Management Review, vol. 51, no. 1, pp. 41-51, doi: 10.1109/EMR.2022.3208818.
13. I. Kanwal, F. Wahid, S. Ali, A. -U. Rehman, A. Alkhayyat and A. Al-Radaei. 2023. "Sentiment Analysis Using Hybrid Model of Stacked Auto-Encoder-Based Feature Extraction and Long Short Term Memory-Based Classification Approach," in IEEE Access, vol. 11, pp. 124181-124197, doi: 10.1109/ACCESS.2023.3313189.
14. B. R. Babu, S. Ramakrishna and S. K. Duvvuri. 2025. "Advanced Sentiment and Trend Analysis of Twitter Data Using CNN-LSTM and Word2Vec," 2025 4th International Conference on Sentiment Analysis and Deep Learning (ICSADL), Bhimdatta, Nepal, pp. 1536-1543, doi: 10.1109/ICSADL65848.2025.10933031.
15. Aniket Kale, Chetan Bawankule, Payal Singanjude, Ganesh Wattamwar, Dr. Simran Khiani. 2021. “Twitter Sentiment Analysis using LSTM Algorithm”, International Journal of Creative Research Thoughts (IJCRT), ISSN: 2320-2882, pp. 15-18.
16. Reference Link: https://github.com.