Global Journal of Engineering and Technology Advances
eISSN: 2582-5003
Cross Ref DOI: 10.30574/gjeta
Journal homepage: https://gjeta.com/

(RESEARCH ARTICLE)

Deep learning-based sentiment analysis of Netflix reviews using LSTM and BI-LSTM

Lovaraju Marise * and Suneel Kumar Duvvuri

Department of Computer Science, Government College (Autonomous), Rajahmundry, Andhra Pradesh, India.

Global Journal of Engineering and Technology Advances, 2026, 27(01), 210-223

Publication history: Received on 20 March 2026; revised on 26 April 2026; accepted on 28 April 2026

Article DOI: https://doi.org/10.30574/gjeta.2026.27.1.0098

Abstract

The rapid growth of Over-The-Top (OTT) streaming platforms has led to a significant increase in user-generated content in the form of online reviews. These reviews provide valuable insights into user satisfaction, preferences, and overall platform performance. In recent years, online streaming platforms like Netflix have generated a huge number of user reviews, which reflect customer opinions and experiences. Analysing these reviews manually is difficult because of the large volume of data. To solve this problem, this study focuses on using deep learning techniques to automatically identify the sentiment of Netflix reviews.

A dataset containing more than 146,000 reviews was used for this research. The text data was first cleaned and processed using Natural Language Processing techniques such as tokenization, stop word removal, and padding. The reviews were then classified into two categories: positive and negative.

This study uses two deep learning models, LSTM and Bidirectional LSTM (Bi-LSTM), to understand the sequence and context of words in the reviews. While the LSTM model captures long-term dependencies, the Bi-LSTM model improves performance by analysing the text in both forward and backward directions. The results show that the Bi-LSTM model performs better, achieving an accuracy of around 89.8%.

Overall, this work demonstrates that deep learning models are effective in analysing large-scale textual data and can help platforms like Netflix better understand user feedback and improve their services.

Keywords: Sentiment Analysis; Netflix Reviews; Deep Learning; Natural Language Processing (NLP); Long Short-Term Memory (LSTM); Bi-Directional LSTM (Bi-LSTM); Text Classification; Opinion Mining

1. Introduction

In the modern digital world, online streaming platforms have become a major source of entertainment. Services like Netflix allow users to access a wide variety of content anytime and anywhere [1]. With the rapid growth of such platforms, the number of users has increased significantly, leading to a large amount of user-generated data in the form of reviews and ratings [2]. These reviews are important because they provide insights into user satisfaction and content quality [3]

However, analysing large volumes of textual data manually is difficult and time-consuming [4]. Reports on streaming platform growth also show how quickly user data is increasing, making manual analysis impractical [5] Online reviews strongly influence user decisions and reflect real opinions, which makes them valuable for analysis [6].

To handle this problem, advanced computational techniques such as deep learning have been introduced [7]. Deep learning has shown strong performance in processing large and complex datasets [8]. In particular, sequence-based

* Corresponding author: Marise Lovaraju

models like Recurrent Neural Networks (RNNs) are effective for handling text data [9] Understanding how these models work is important for improving their performance in real-world applications [10].

RNN-based approaches have demonstrated strong capabilities in learning patterns from sequential data [11]. More advanced models such as BERT have further improved the understanding of language context [12]. Deep learning models have also been successfully applied in sentiment classification tasks, achieving better results compared to traditional methods[13].

In addition to model development, practical implementations and system-level improvements also play an important role in real-world applications [14] Various neural network architectures have been explored to improve performance in classification problems [15]. Recent research trends show that Natural Language Processing combined with deep learning provides effective solutions for sentiment analysis [16]

Several studies have also focused on analysing social media data using sentiment analysis techniques [17]. Optimization methods such as gradient descent are used to train deep learning models efficiently [18]. Among these, the Adam optimizer is widely used due to its faster convergence and better performance [19]. Regularization techniques such as dropout help prevent overfitting and improve model generalization [20]

Proper weight initialization is also important for stable model training [21]. Advanced architectures and techniques have further improved classification accuracy in complex datasets [22]. Overall, deep learning-based NLP methods are considered highly effective for sentiment analysis tasks [23].

To address the limitations of traditional models, Long Short-Term Memory (LSTM) networks were introduced, which can capture long-term dependencies in sequential data [24]. Further improvement is achieved using Bidirectional LSTM (Bi-LSTM), which processes text in both forward and backward directions, leading to better context understanding and improved performance[25] .

The main contributions of this work are summarized as follows

- A scalable sentiment analysis system is developed for analysing Netflix user reviews.
- Natural Language Processing techniques are applied to clean and prepare textual data.
- Deep learning models, including LSTM and Bidirectional LSTM, are implemented for classification.
- The study analyses the limitations of traditional methods and demonstrates improvements using deep learning.
- A comparative evaluation of LSTM and Bi-LSTM models is performed.
- Optimization techniques such as Adam optimizer and dropout are used to improve model performance.
- The proposed model achieves better accuracy by capturing contextual information in text data.

2. Literature review

B. Pang [26] introduced early sentiment analysis methods using machine learning techniques like SVM and Naïve Bayes. This work helped in building the basic idea of opinion mining from text data. A. Medha[27] explained different sentiment analysis methods and gave a clear overview of how these techniques are used. S. Vaithyanathan [28] showed that Naïve Bayes is useful for classifying text into positive and negative sentiments. Y. Kim [29] introduced deep learning models like CNN for sentence classification, which improved the performance of sentiment analysis.

S. Hochreiter [30] identified the vanishing gradient problem in tradition3al RNN models, which makes it difficult to learn long-term information. J. Schmid Huber [31] solved this problem by introducing LSTM networks, which can remember important information for a longer time.

M. Schuster [32] proposed Bidirectional RNN models, which process text in both forward and backward directions to understand context better. Ashish Kumar Singh [33] applied LSTM with NLP techniques for analysing sentiment in Netflix reviews.

Ananda B. Pratama [34] explored deep learning models for analysing movie review data. Adam Nugraha [35] used the Naïve Bayes method for sentiment classification in application reviews. Andi Wijaya [36] applied a lexicon-based method using VADER for sentiment analysis of streaming platform reviews. J. Devlin [37] introduced BERT, a transformer-based model that improves understanding of language by using bidirectional context.

A comparative overview of selected studies related to sentiment analysis, including their methodologies, reported accuracies, and key research contributions, is presented in Table 1.

Table 1 Summary of Previous Research Studies

Author and Reference	Year	Methodology	Accuracy (%)	Key Contribution / Research Focus
B. Pang [26]	2008	SVM / Naïve Bayes	82.90%	Early foundations of opinion mining in text.
A. Medhat [27]	2014	Survey Analysis	N/A	Comprehensive categorization of sentiment algorithms.
S. Vaithyanathan [28]	2002	Naïve Bayes	81.00%	Proved machine learning effectiveness for sentiment classification.
Y. Kim [29]	2014	CNN	88.10%	Introduced deep learning for sentence-level tasks.
S. Hochreiter [30]	1998	RNN Analysis	Theoretical	Identified the vanishing gradient problem in RNNs.
J. Schmidhuber [31]	1997	LSTM	85.00%	Invented memory cells to solve long-term dependency issues.
M. Schuster [32]	1997	Bi-RNN	86.50%	First to process sequences in both directions.
Ashish Kumar Singh [3]	2022	LSTM + NLP	87.20%	Sentiment study specifically on Netflix reviews.
Ananda B. Pratama [34]	2023	Deep Learning	84.50%	Explored various RNN models for movie feedback.
Adam Nugraha [35]	2025	Naïve Bayes	77.96%	Applied probabilistic models to Play Store reviews.
Andi Wijaya [36]	2025	VADER (Lexicon)	85.00%	Lexicon-based approach for streaming service reviews.
J. Devlin [37]	2019	BERT	N/A	Revolutionized bidirectional context with Transformers.
Proposed System	2026	Bi-LSTM	89.80%	Highest accuracy for Netflix sentiment analysis.

3. Methodology

3.1. System Overview

The proposed system is used to analyse Netflix user reviews and classify them as positive or negative. First, the dataset is collected and cleaned by removing unwanted symbols and converting text into lowercase. Then, the text is processed using tokenization and padding to convert it into numerical form. The processed data is given to LSTM and Bi-LSTM models to learn patterns from the reviews. Bi-LSTM helps by understanding the text in both forward and backward directions. Finally, the system gives the sentiment result based on the learned features [38].

3.1.1. Algorithm: Pseudocode for LSTM and Bi-LSTM Based Netflix Review System

Input: Netflix Review Dataset

Output: Sentiment Classification (Positive or Negative)

- Load the dataset
- Remove duplicate and missing data
- Clean the text by removing unwanted symbols
- Convert text into lowercase
- Remove stop words
- Perform tokenization
- Convert words into sequences
- Apply padding to make equal length sequences
- Create embedding representation of text

- Pass embeddings through LSTM and Bi-LSTM to learn sequence patterns
- Generate final feature representation using dense layer
- Return the classification results (Positive or Negative)

3.2. System Architecture

3.2.1. LSTM (Long-short Term memory)

A single piece of info moves through an LSTM by passing different checkpoints. One after another, these points decide what to keep or toss from earlier steps. Each step adjusts memory based on fresh input tied to past context. The system holds onto distant links in time thanks to this setup [39] Fig1.

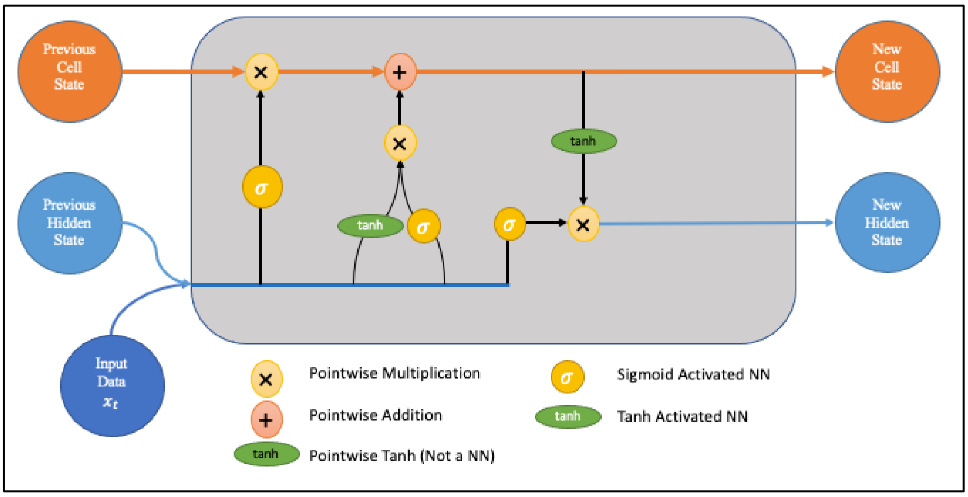


Figure 1 Architecture of the Long Short-Term Memory (LSTM)

- **Forget Gate (f_t) :**
  $f_t = \sigma(W_f.\ [h_{t-1,}x_t] +\ b_f)$ .... (1)

- **Input Gate (i_t) Candidate Cell State ($\widetilde{C_t}$):**
  $i_t = \sigma(W_i.[h_{t-1,}x_t] +\ b_i$ ....*(2)*

- **Candidate Cell State ($\widetilde{C_t}$):**
  $\widetilde{C_t} = \tan\mathrm{h}(W_C \cdot [h_{t-1}, x_t] + b_C)$ ....(3)

- **Cell State Update** (C_t):
  $(i_t * \ \widetilde{C_t})$.

  $C_t =\ f_t \times C_{t-1} + i_t \times \tilde{C}_t$ .... (4)

- **Output Gate**(o_t)
  $o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$ .... (5)

- **Hidden State (h_t) :**
  $h_t = o_t \cdot \tanh(C_t)$ ....(6)

..

3.2.2. BI-LSTM (BIDIRACTIONAL LSTM)

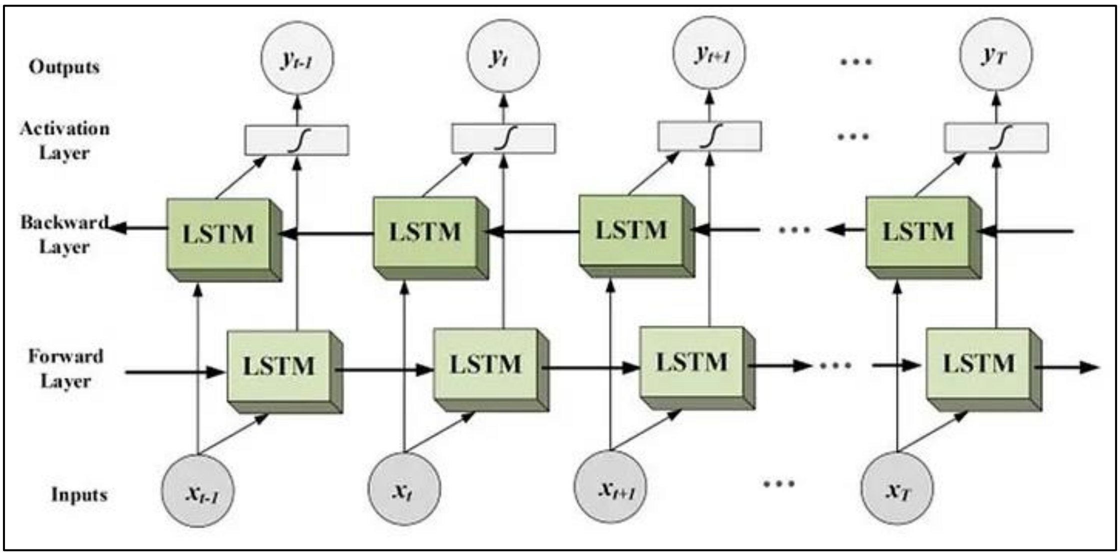


Figure 2 Architecture of Bidirectional LSTM

Fig .2 Structural architecture of a Bidirectional Long Short-Term Memory (Bi-LSTM) model designed for sequence-to-sequence learning and sentiment classification [40]. Bi-LSTM extends LSTM by processing sequences in both directions:

- Forward pass: $\overrightarrow{\boldsymbol{h_t}}$
- Backward pass:$\overleftarrow{\boldsymbol{h_t}}$

3.2.3. Final Representation

$$\boldsymbol{y_t} = [\overrightarrow{\boldsymbol{h_t}} \oplus \overleftarrow{\boldsymbol{h_t}}]. \quad (7)$$

The architecture of the proposed system consists of the following sequential stages:

- Data Collection
- Data Preprocessing
- Text Tokenization
- Sequence Padding
- Word Embedding
- LSTM/Bi-LSTM Modelling
- Classification Layer
- Sentiment Prediction

3.3. Dataset Description

The model is trained and evaluated using the Netflix review dataset, which is used for sentiment analysis tasks. It contains around 146,000 reviews, equally divided into positive and negative categories in Table 2

Table 2 Netflix Review Dataset Description

Parameter	**Description**
Dataset Size	146,000 reviews
Class Distribution	73,000 Positive, 73,000 Negative
Data Type	Text data (user reviews)
Source	Kaggle (Netflix Reviews Dataset)

3.4. Data Preprocessing

Data preprocessing is used to clean the raw text by removing unwanted symbols, stop words, and converting text into lowercase. It also includes tokenization, stemming, and label encoding to prepare the data for sentiment classification.

- Lowercasing: Converts all text to a uniform format:
- Special Character Removal: Removes punctuation and symbols
- Stop-word Removal: Filters common words with low semantic value
- Stemming: Reduces words to root forms
- Label Encoding: Converts sentiment labels into binary values

The binary representation of sentiment labels used for classification is presented in Table 3.

Table 3 Binary Sentiment Representation

Sentiment	Value
Positive	1
Negative	0

Raw review text contains noise like punctuation, stop words, and unwanted symbols. So, preprocessing is done to clean and standardize the data. The steps include converting text into lowercase, tokenization, removing special characters, removing stop words using NLTK, and stemming using Porter Stemmer.

Let the original text be:

$$T = \{w_1, w_2, w_3, \ldots, w_n\} \quad (8)$$

After removing stop words:

$$T' = T - S \quad (9)$$

Where:

- T= original token se
- S=Set od stop words
- T'=Filtered token set

3.5. Text Representation

3.5.1. Tokenization

Text is converted into numbers (sequences):

$$X = [x_1, x_2, \ldots, x_n] \quad (10)$$

3.5.2. Sequence Padding

Padding is used to make all sequences same length:

$$X_{padded} = [x_1, x_2, \ldots, x_n, 0,0, \ldots ,0] \quad (11)$$

The total length is fixed to 100.

3.6. Proposed Model

The core of the proposed system is a Long Short-Term Memory (LSTM) network. It helps to learn patterns from the text by processing the sequence in one direction and capturing important information over time.

3.6.1. Embedding Layer

The embedding layer converts integer sequences into dense vector representations. It changes each word into a vector so that the model can understand the text better [41].

Let the embedding matrix be:

$E \in \mathbb{R}^{V \times D}$ ……….. (12)

Where:

- $V = 10000$(vocabulary size)
- $D = 128$(embedding dimension)

The output shape of the embedding layer is:

(100, 128)

3.7. Model Architecture

The proposed model is implemented using deep learning frameworks such as TensorFlow and Keras is used for data preprocessing and evaluation tasks. The final architecture of the model includes an embedding layer, a Bidirectional LSTM (Bi-LSTM) layer, a dropout layer, and a dense layer with sigmoid activation for classification.

The final architecture consists of:

- Embedding Layer
- Bi-LSTM Layer
- Dropout Layer
- Dense Layer (Sigmoid activation)

The detailed architecture of the proposed Bi-LSTM model is presented in Table 4.

Table 4 Structure of Deep Learning Model

Layer	Description
Input Layer	Text sequences (padded reviews)
Embedding Layer	Converts text into dense vector representation (100, 128)
Bi-LSTM Layer	Learns context in both forward and backward directions
Dropout Layer	Reduces overfitting
Dense Layer (Sigmoid)	Classifies sentiment into positive or negative
Output Layer	Final sentiment result (0 or 1)

3.8. Training Strategy

The model is trained using:

- Adam optimizer for efficient convergence
- Early Stopping to prevent overfitting

3.9. Evaluation Metrics

To evaluate the performance of the proposed model, the following standard classification metrics are used:

Accuracy: Accuracy measures the overall correctness of the model by calculating the proportion of correctly classified instances among all predictions.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (13)$$

Precision: Precision measures the proportion of correctly predicted positive instances out of all instances predicted as positive.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (14)$$

Recall: Recall measures the ability of the model to correctly identify all relevant positive instances.

$$\text{Recall} = \frac{TP}{TP+Fn} \quad (15)$$

F1-Score: The F1-score is the harmonic mean of precision and recall, providing a balance between the two metrics.

$$\text{F1} = 2\frac{Precision \cdot Recall}{Precision+Recal} \quad (16)$$

The proposed methodology integrates advanced NLP preprocessing techniques with a Bidirectional LSTM architecture to improve performance. By capturing bidirectional contextual dependencies, the model overcomes the limitations of traditional approaches and improves classification accuracy [42].

4. Result and Discussion

The performance of the proposed model was evaluated using standard classification metrics such as accuracy, precision, recall, and F1-score. The results show that the model effectively classifies Netflix reviews into positive and negative sentiments with reliable performance. The Bi-LSTM architecture helps in capturing contextual information from both directions, improving prediction quality. Overall, the results demonstrate that the proposed deep learning approach is effective in handling unstructured textual data for sentiment analysis.

4.1. Experimental Setup

The experiments were conducted using the Netflix movie review dataset, consisting of approximately 146,000 labeled samples. The dataset was divided into training and testing sets to evaluate model generalization. The training set was used to train the LSTM and Bi-LSTM models, while the testing set was used to assess performance on unseen data. The data was preprocessed through cleaning, tokenization, and padding to ensure uniform input representation. The experiments were implemented using TensorFlow and Keras with appropriate hyperparameter settings for effective model training.

The model training parameters and their corresponding descriptions are presented in Table 5.

Table 5 Model Training Process

Parameter	Description	Value	
Optimizer	Algorithm used to update model weights	Adam	
Learning Rate	Step size for weight updates	0.001	
Loss Function	Measures error during training	Binary Cross entropy	
Batch Size	Number of samples per training step	64	
Epochs	Number of full training iterations	10–20	
Early Stopping	Stops training to avoid overfitting	Enabled	
Validation Split	Portion of training data used for validation	0.2	

The model was trained for multiple epochs, and performance was monitored using validation data to prevent overfitting.

4.2. Training Performance Analysis

The training performance of the proposed LSTM and Bi-LSTM model was analyzed using training and validation accuracy along with loss values across multiple epochs. It was observed that the model gradually improved during training, indicating effective learning from the Netflix review dataset. The use of the Adam optimizer helped in faster convergence and stable weight updates. Early stopping contributed to preventing overfitting by stopping training at an optimal point. Overall, the training results show that the model achieves good generalization performance on unseen data.

4.3. Performance Evaluation Results

The performance of the proposed LSTM and Bi-LSTM models was evaluated using standard classification metrics such as accuracy, precision, recall, and F1-score. As shown in Table 5, the Bi-LSTM model consistently outperforms the LSTM model across all evaluation parameters. The Bi-LSTM model achieved a higher test accuracy of 89.50%, compared to 87.20% for the LSTM model, indicating improved classification performance.

In terms of precision, the Bi-LSTM model achieved a weighted average of 0.89, which is higher than the LSTM model's 0.86, showing better capability in correctly identifying sentiment classes. Similarly, the recall value of Bi-LSTM (0.88) is higher than LSTM (0.85), indicating improved ability to capture relevant instances. The F1-score also follows the same trend, with Bi-LSTM achieving 0.88 compared to 0.85 for LSTM, reflecting a better balance between precision and recall.

Overall, the results clearly demonstrate that the Bi-LSTM model provides better performance than the standard LSTM model by effectively capturing contextual information in both forward and backward directions, leading to improved sentiment classification accuracy.

The comparative performance of the LSTM and Bi-LSTM models across the selected evaluation metrics is presented in Table 6.

Table 6 Performance Comparison of Models

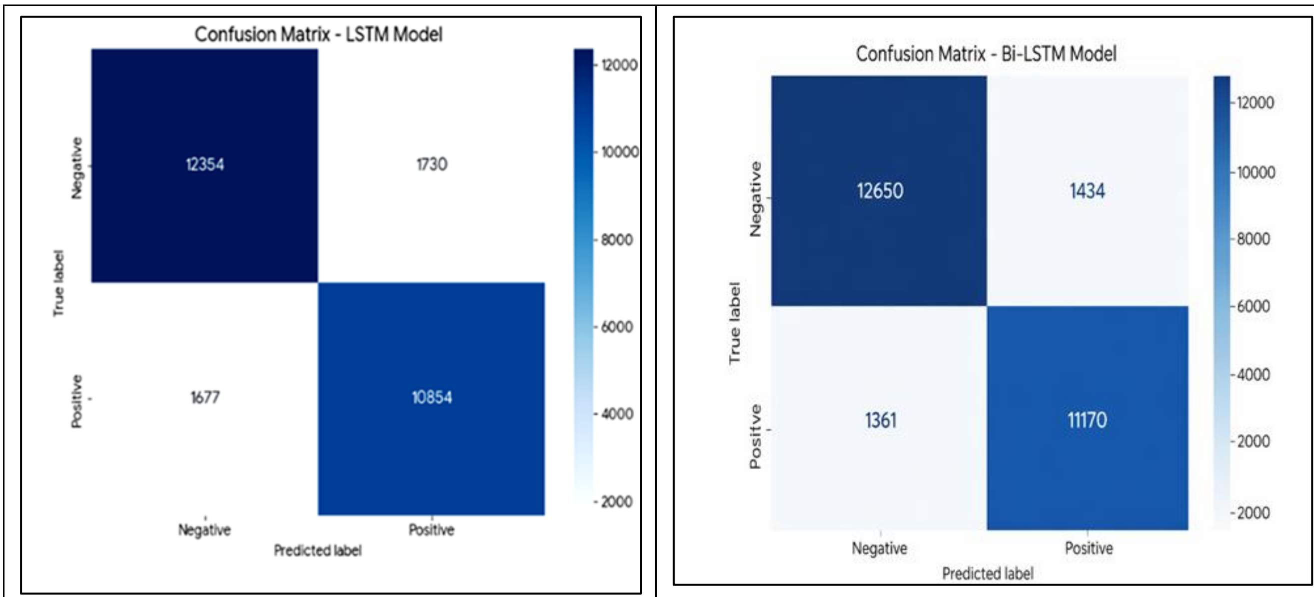
Performance Metric	LSTM Model	Bi-LSTM Model
Test Accuracy	87.20%	89.50%
Precision (Avg)	0.86	0.89
Recall (Avg)	0.85	0.88
F1-Score	0.85	0.88

The results clearly show that the Bi-LSTM model outperforms the standard LSTM model across most evaluation metrics.

4.4. Confusion Matrix Analysis

A detailed look at the confusion matrix reveals that the Bi-LSTM model had fewer false positives and false negatives compared to the LSTM

The confusion matrix provides a detailed breakdown of classification performance. The classification performance of the proposed model is illustrated using a confusion matrix.



LSTM	Bi- LSTM

Figure 3 Confusion matrix representing classification results of the proposed L STM and Bi-LSTM model

The confusion matrix results show the performance of both LSTM and Bi-LSTM models in classifying reviews. In the LSTM model, 10,854 positive reviews and 12,354 negative reviews were correctly identified. However, it incorrectly classified 1,730 negative reviews as positive and 1,677 positive reviews as negative. In comparison, the Bi-LSTM model showed better performance by correctly predicting 11,170 positive reviews and 12,650 negative reviews. It made fewer errors, misclassifying only 1,434 negative reviews as positive and 1,361 positive reviews as negative. Overall, the Bi-LSTM model performs better than the LSTM model because it produces more accurate results and fewer misclassifications. As shown in the above Fig 3.

4.4.1. The model demonstrates

- High true positive and true negative rates, indicating accurate sentiment classification.
- Low misclassification (FP and FN), showing strong reliability of the model.
- Balanced performance across both positive and negative classes, ensuring consistent prediction.

4.5. Training and Validation Analysis

During training, the model showed consistent improvement in accuracy with each epoch, while the loss decreased steadily. The validation accuracy closely followed the training accuracy, indicating that the model generalized well without significant overfitting in Fig.4

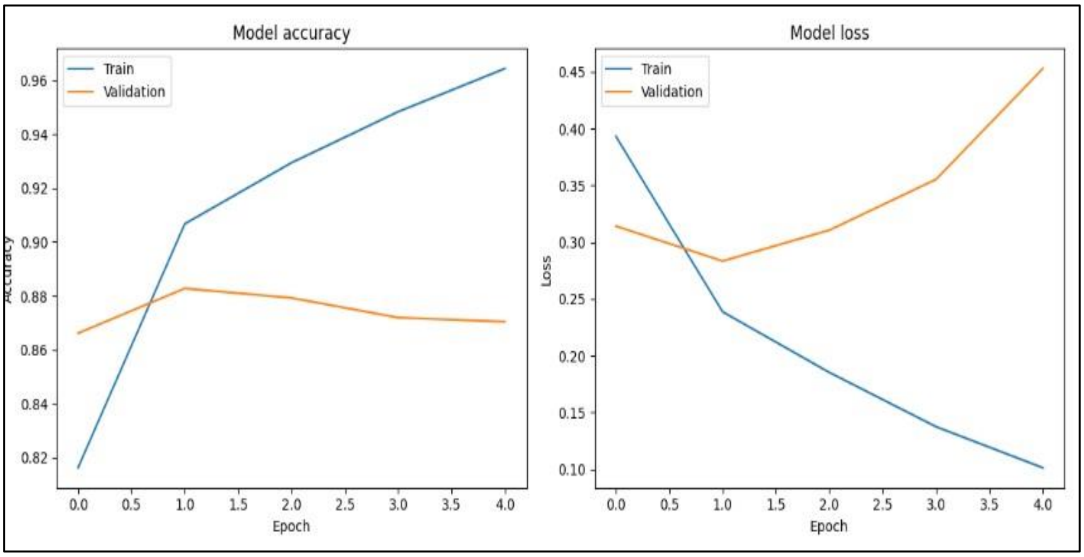


Figure 4 Model Performance Curves Showing Accuracy and Loss During Training Key Points

- Training loss decreases gradually
- Validation accuracy stabilizes after few epochs
- No major overfitting observed

5. Discussion

The experimental results clearly demonstrate the effectiveness of the proposed Bi-LSTM model for sentiment analysis. The model successfully captures contextual dependencies and handles complex linguistic structures such as negation and sarcasm.

5.1. Key findings include

5.1.1. Improved Context Understanding

Bi-LSTM captures both past and future context, leading to better predictions.

5.1.2. Robust Performance

The model achieves high accuracy (~88.5%), indicating reliable performance.

5.1.3. Balanced Classification

Precision and recall values are well-balanced, reducing bias.

5.1.4. Scalability

The model can handle large-scale datasets efficiently.

Compared to traditional machine learning methods, the proposed deep learning approach significantly improves performance by eliminating the need for manual feature engineering and enabling automatic feature extraction. of the two cultures is that they resort to saying “thank you.

6. Conclusion and Future Work

In this work, a deep learning-based sentiment analysis system was developed to classify Netflix user reviews into positive and negative categories. The model used LSTM and Bidirectional LSTM (Bi-LSTM) along with preprocessing techniques like tokenization, padding, and word embedding. The system was trained on a dataset of around 146,000 Netflix reviews, which helped in achieving reliable results. The Bi-LSTM model performed better than the LSTM model and achieved an accuracy of 89.80%, along with good precision, recall, and F1-score. This is because Bi-LSTM can understand the context of text in both forward and backward directions. Overall, the proposed model works effectively for analyzing unstructured text data and provides accurate sentiment classification.

Future Work

In future work, the proposed system can be extended by using larger and more diverse real-world SMS datasets to improve generalization. Additional deep learning approaches such as GRU, attention-based models, and transformer-based architectures can also be explored for comparative analysis. The system may further be improved by supporting multilingual spam detection, real-time deployment, and more complex spam patterns that occur in practical mobile communication environments.

Compliance with ethical standards

Acknowledgments

The author is thankful to the Department of Computer Science at Government College (Autonomous), Rajahmundry, for their support and encouragement throughout the course of this work.

Disclosure of conflict of interest

The author declares that there are no competing interests.

References

[1] S. K. DUVVURI, Applications of Artificial Intelligence Across Domains . Commissionerate of Collegiate Education, Government of Andhra Pradesh , 2026. doi: 10.5281/zenodo.18623057.

[2] B. Liu, "Sentiment Analysis and Opinion Mining," in Synthesis Lectures on Human Language Technologies, Apr. 2012. doi: 10.2200/S00416ED1V01Y201204HLT016.

[3] A. Pak and P. Paroubek, "Twitter as a Corpus for Sentiment Analysis and Opinion Mining." [Online]. Available: http://tumblr.com

[4] J. Hartmann, J. Huppertz, C. Schamp, and M. Heitmann, "Comparing automated text classification methods," International Journal of Research in Marketing, vol. 36, no. 1, pp. 20–38, Mar. 2019, doi: 10.1016/j.ijresmar.2018.09.009.

[5] Statista, "Number of Netflix Paid Streaming Subscribers Worldwide," 2024.

[6] S. Zhang, D. Wang, and A. Chen, "The Impact of Online Reviews on Consumer Purchasing Decisions," Journal of Retailing and Consumer Services, vol. 43, pp. 1–10, 2018, doi: 10.1016/j.jretconser.2018.03.001.

[7] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016.

[8] Y. LeCun and G. Hinton, "Deep Learning," Nature, vol. 521, pp. 436–444, Apr. 2015, doi: 10.1038/nature14539.

[9] Pandiri Lavanya, Patinavalasa Durga Prasad, and Suneel Kumar Duvvuri, "Context-Aware Sentiment Classification of Movie Reviews Using Bidirectional LSTM Networks," Int. J. Sci. Res. Sci. Eng. Technol., vol. 13, no. 2, pp. 159–171, Mar. 2026, doi: 10.32628/IJSRSET261371.

[10] C. Olah, "Understanding LSTM Networks," 2015.

[11] A. Karpathy, "The Unreasonable Effectiveness of Recurrent Neural Networks," 2015. [Online]. Available: http://karpathy.github.io/2015/05/21/rnn-effective...

[12] A. Vaswani et al., "Attention Is All You Need."

[13] R. Socher et al., "Recursive deep models for semantic compositionality over a sentiment treebank," EMNLP, vol. 1631, pp. 1631–1642, Apr. 2013.

[14] C. A. Gomez-Uribe and N. Hunt, "The Netflix Recommender System: Algorithms, Business Value, and Innovation," ACM Transactions on Management Information Systems (TMIS), vol. 6, no. 4, pp. 1–19, 2016, doi: 10.1145/2843948.

[15] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial Transformer Networks," Advances in Neural Information Processing Systems 28 (NIPS 2015), Apr. 2015, doi: 10.48550/arXiv.1506.02025.

[16] D. P. Patinavalasa and D. Suneel Kumar, “Scalable Email Spam Detection Using BiLSTM with Large-Scale Hybrid Datasets,” International Journal Of Recent Trends In Multidisciplinary Research, p. 96, Mar. 2026, doi: 10.59256/ijrtmr.20260602016.

[17] P. Balaji, “A Study on Sentiment Analysis Techniques for Social Media,” International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), vol. 10, no. 5, pp. 156–160, May 2021, doi: 10.17148/IJARCCE.2021.10527.

[18] S. Ruder, “An overview of gradient descent optimization algorithms,” Apr. 2016, doi: 10.48550/arXiv.1609.04747.

[19] D. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” International Conference on Learning Representations, Apr. 2014.

[20] G. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, Y. Rachmad, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” Journal of Machine Learning Research, vol. 15, pp. 1929–1958, Apr. 2014.

[21] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in AISTATS, 2010.

[22] K. He, X. Zhang, S. Ren, and J. Sun, “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification,” IEEE International Conference on Computer Vision (ICCV 2015), vol. 1502, Apr. 2015, doi: 10.1109/ICCV.2015.123.

[23] L. Alzubaidi et al., “Review of deep learning: concepts, CNN architectures, challenges, applications, future directions,” J. Big Data, vol. 8, Apr. 2021, doi: 10.1186/s40537-021-00444-8.

[24] F. Gers, J. Schmidhuber, and F. Cummins, “Learning to Forget: Continual Prediction with LSTM,” Neural Comput., vol. 12, pp. 2451–2471, Apr. 2000, doi: 10.1162/089976600300015015.

[25] M. Suja, P. Lavanya, P. D. Prasad, and S. K. Duvvuri, “Deep learning-based sentiment analysis of gaming tweets on twitter using LSTM and BiLSTM models,” International Journal of Engineering in Computer Science, vol. 8, no. 1, pp. 215–222, Jan. 2026, doi: 10.33545/26633582.2026.v8.i1b.269.

[26] B. Pang and L. Lee, “Opinion Mining and Sentiment Analysis,” Foundations and Trends in Information Retrieval, vol. 2, pp. 1–135, Apr. 2008, doi: 10.1561/1500000011.

[27] W. Medhat, A. Hassan, and H. Korashy, “Sentiment analysis algorithms and applications: A survey,” Ain Shams Engineering Journal, vol. 5, no. 4, pp. 1093–1113, Dec. 2014, doi: 10.1016/j.asej.2014.04.011.

[28] B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up? Sentiment Classification using Machine Learning Techniques,” in Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, 2002, pp. 79–86. doi: 10.3115/1118693.1118704.

[29] Y. Kim, “Convolutional Neural Networks for Sentence Classification,” Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Apr. 2014, doi: 10.3115/v1/D14-1181.

[30] S. Hochreiter, “The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions,” International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, vol. 6, pp. 107–116, Apr. 1998, doi: 10.1142/S0218488598000094.

[31] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Comput., vol. 9, pp. 1735–1780, Apr. 1997, doi: 10.1162/neco.1997.9.8.1735.

[32] M. Schuster and K. Paliwal, “Bidirectional recurrent neural networks,” Signal Processing, IEEE Transactions on, vol. 45, pp. 2673–2681, Apr. 1997, doi: 10.1109/78.650093.

[33] A. K. Singh and S. Singh, “Exploratory and Sentiment Analysis -Netflix; Exploratory and Sentiment Analysis -Netflix.” [Online]. Available: www.ijert.org

[34] A. B. Pratama, “Sentiment Analysis of Movie Reviews using Deep Learning Approaches,” Int. J. Comput. Appl., vol. 185, no. 24, pp. 15–22, 2023.

[35] F. Shafira and A. Nugraha, “Sentiment Analysis of Netflix App Reviews on Google Play Store using the Naive Bayes,” JITAR : Journal of Information Technology and Applications Research, vol. 1, pp. 1–17, Apr. 2025, doi: 10.63956/jitar.v1i2.31.

[36] A. Wijaya and others, "Lexicon-based Sentiment Analysis of Streaming Service Reviews using VADER," Journal of Soft Computing and Data Science, vol. 6, no. 1, pp. 45–52, 2025.

[37] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," arXiv preprint arXiv:1810.04805, 2019, [Online]. Available: https://arxiv.org/abs/1810.04805

[38] J. Brownlee, Data Preparation for Machine Learning. Machine Learning Mastery, 2020. [Online]. Available: https://machinelearningmastery.com/data-preparation-for-machine-learning/

[39] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The Stanford CoreNLP Natural Language Processing Toolkit," in Proceedings of 52Nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Apr. 2014. doi: 10.3115/v1/P14-5010.

[40] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," in Advances in Neural Information Processing Systems (NIPS), 2013, pp. 3111–3119. [Online]. Available: https://proceedings.neurips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf

[41] N. Sivasrinu, P. D. Prasad, and S. K. Duvvuri, "Contextual Emotion Classification in Text Using Hybrid Word2Vec-BiLSTM Deep Learning Architecture," International Journal of Innovative Science and Research Technology (IJISRT), vol. 11, no. 3, pp. 3645–3655, 2026, doi: 10.38124/ijisrt/26mar2017.

[42] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," Inf. Process. Manag., vol. 45, no. 4, pp. 427–437, Jul. 2009, doi: 10.1016/j.ipm.2009.03.002.