International Journal of Engineering Research and Development
e- ISSN: 2278-067X, p-ISSN: 2278-800X, www.ijerd.com
Volume 22, Issue 2 (February 2026), PP 20-26

Sentiment Classification Analysis Using Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) on the Weibo_Senti_100k Dataset

Jie Zeng,Zhan Wen*,Xu Gao,Meizi Luo
[1]*School of Communication Engineering, Chengdu University of Information Technology*
Chengdu, 610225, China
Corresponding Author: Zhan Wen.

ABSTRACT
The rapid growth of e-commerce has generated a massive volume of product reviews, which serve as valuable resources for analyzing consumer satisfaction and product feedback. Sentiment analysis techniques can efficiently extract emotional tendencies from such unstructured texts, providing critical insights for businesses to optimize products and formulate marketing strategies. This study aims to investigate the performance differences between Recurrent Neural Networks (RNN) and their enhanced variant—Long Short-Term Memory (LSTM)—in the task of sentiment classification of product reviews. Using the PyTorch Lightning framework, we constructed RNN and LSTM models, respectively, and trained and evaluated them on the Weibo_Senti_100k dataset. Experimental results demonstrate that the LSTM model significantly outperforms the traditional RNN model across key metrics such as test accuracy, recall, and F1-score. The LSTM model achieved a test accuracy of 91.5%, a notable improvement over the RNN model's 88.9%. This improvement is primarily attributed to the gating mechanism of LSTM, which effectively captures long-range dependencies in reviews. This study provides practical guidance for model selection and implementation in consumer sentiment analysis based on deep learning.
Keywords: *sentiment analysis,RNN, LSTM,product reviews,PyTorch Lightning.*

Date of Submission: 24-01-2026 Date of acceptance: 06-02-2026

I. INTRODUCTION

The global e-commerce sector has experienced sustained, rapid growth in recent years, generating a vast volume of unstructured data in the form of product reviews. These texts provide direct insights into users' emotional tendencies and subjective evaluations of products, services, and consumption experiences. Consequently, automated sentiment analysis (classifying text as positive or negative) has become essential for businesses to informmarket strategies, optimize products, and enhance customer satisfaction.

This study conducts a comparative analysis of two neural network architectures—Recurrent Neural Networks (RNN) and Long Short-Term Memory networks (LSTM)—for sentiment classification of product reviews. Models are trained and evaluated using the Weibo_Senti_100k dataset, a collection of social media texts labeled as positive or negative. Our objective is to identify the superior model by evaluating key metrics including accuracy, recall, F1-score, training time, and loss.

Although deep learning models have proven effective for sentiment analysis, systematic comparisons between the fundamental RNN and its enhanced variant, LSTM, are still lacking, particularly in the domain of product reviews. To address this gap, the key contributions of our work are:

- This study preprocesses the Weibo_Senti_100k dataset to ensure high-quality input for model training, including cleaning, word segmentation, stop-word removal, and converting the data into a numerical format suitable for neural network input. The dataset is also split into 80% training set and 20% validation set to enable accurate performance evaluation of the models.
- This study trains two deep learning models: a Recurrent Neural Network (RNN) and a Long Short-Term Memory network (LSTM). Both models are developed and trained using the PyTorch Lightning framework. They are trained under similar configurations and evaluated using metrics such as accuracy, recall, F1-score, loss function, and training time.

II. RELATED WORK

Sentiment analysis, a crucial branch of Natural Language Processing (NLP), is widely applied in fields such as opinion mining, product review classification, and market trend analysis. Early research primarily relied on rule-based systems using sentiment lexicons and traditional machine learning algorithms like Naïve Bayes, Support Vector Machines (SVM), and decision trees. Although effective in certain scenarios, these approaches have limited feature extraction capabilities and struggle to capture the complex semantic structures and contextual relationships within text. For instance, traditional methods based on the Bag-of-Words model often hit an accuracy plateau in sentiment classification tasks due to their neglect of word order and semantic relationships, typically underperforming compared to deep learning models.

With the advancement of deep learning, Recurrent Neural Networks (RNNs) have been introduced to text sentiment analysis tasks due to their inherent ability to process sequential data. By modeling temporal dependencies through their recurrent structure, RNNs can theoretically capture contextual relationships between words in a sentence. Recent studies further validate the applicability of RNNs across different modalities of sequential data analysis, demonstrating strong sequence modeling capabilities in tasks such as speech emotion recognition [1] and real-time chord perception for music analysis [2]. Furthermore, in non-text domains such as wireless sensor networks [3] and rainfall forecasting [4], RNNs have also demonstrated their versatility in modeling time-series data.

However, standard RNNs are prone to vanishing or exploding gradient problems during training, which severely limits their capacity to retain long-range contextual information. Recent research similarly observed the limitations of basic RNNs in modeling long-term dependencies. For example, in short-term electric load forecasting, RNNs were found to underperform compared to more advanced recurrent network variants [5].

To overcome these shortcomings, Long Short-Term Memory (LSTM) networks were proposed. By incorporating gating mechanisms (input, forget, and output gates), LSTMs can selectively retain or discard information, significantly enhancing the model's ability to capture long-range semantic dependencies. This advantage has been validated across multiple domains: in speech emotion recognition tasks, LSTMs achieved approximately a 5% performance improvement over basic RNNs [1]; in power load forecasting, LSTM's prediction accuracy was about 7.5% higher than that of RNNs [5]. In the field of text sentiment analysis, numerous studies have confirmed the superior performance of LSTM. Notably, Atlas et al. [6] systematically compared various deep learning models in modern product review sentiment analysis and found that RNN-based LSTM models excelled in capturing fine-grained sentiment features.

Examining the current research landscape, the performance advantages of LSTM are well-established, but related studies exhibit two distinct tendencies: first, a dominance of hybrid models, with most research focusing on optimizing combinations of LSTM with other complex architectures. For instance, the BiGRU and RNN-based LSTM hybrid model proposed by Atlas et al. [6] achieved optimal performance in product review sentiment analysis. Second, there is a trend toward scenario-specific specialization, such as integrating attention mechanisms with LSTM for product review analysis or developing multimodal LSTM frameworks for processing social media data. However, these studies have not typically employed basic RNN as a direct comparative baseline.

More critically, systematic comparative studies between basic RNN and LSTM models remain relatively scarce in the specific application context of product review sentiment analysis. Existing literature that attempts such comparisons often has notable limitations. For example, some studies employ different word embeddings (e.g., Word2Vec vs. GloVe) or hyperparameter configurations, making it impossible to attribute performance differences solely to model architecture. Although the study by Atlas et al. [6] provides comparative data for RNN and LSTM in modern product review analysis, it does not delve into a granular comparison of their capability to capture sentiment associated with specific product attributes or contextual nuances.

This paper aims to address this research gap by conducting a fair and controlled comparison between RNN and LSTM under a unified experimental environment—using identical preprocessing pipelines, hyperparameter settings, and evaluation metrics. The experiment is based on the Weibo_Senti_100k dataset, focusing on evaluating the performance of both models in capturing the sentiment polarity of user reviews, thereby providing empirical evidence to guide model selection in practical applications.

III. METHODOLOGY

3.1 Data and Preprocessing

The Weibo_Senti_100k dataset used in this study consists of labeled text samples indicating either positive or negative sentiment. As a widely adopted benchmark dataset in the field of Chinese Natural Language Processing (NLP) specifically designed for sentiment analysis tasks, Weibo_Senti_100k contains approximately 100,000 Chinese comments crawled from Sina Weibo. Each comment is automatically annotated as either "positive" or "negative", based on the emotional emojis attached by users when posting. The dataset's most significant characteristic is its authentic reflection of the Chinese internet language ecosystem, containing

abundant informal expressions, internet neologisms, and colloquial sentence structures. This makes it particularly suitable for training and evaluating models for short-text sentiment classification in real-world social media scenarios.

All textual inputs underwent a standardized preprocessing pipeline to ensure data consistency. The workflow involved: filtering out all non-Chinese characters; removing unwanted particles and stop words using a stop word lexicon; eliminating punctuation and special characters; performing word segmentation using the Jieba tool; and finally, converting the tokens into vector representations.

To facilitate supervised learning and model validation, the preprocessed data was split into training and test sets according to an 8:2 ratio, with 80% used for training and 20% for testing. This division ensures the model has sufficient data to learn patterns while maintaining a representative set for evaluating model performance. The preprocessing pipeline is illustrated in Figure 1.

Figure1: Data Preprocessing Pipeline

3.2 Model Architecture

Two types of deep learning models were implemented using PyTorch Lightning: an RNN model and an LSTM model.

The RNN model begins with an embedding layer that transforms the input word sequence into dense vector representations. Subsequently, the recurrent neural network layer processes each word in the text sequentially, capturing temporal dependencies and contextual information between words through its recurrent connections. The hidden state of the RNN layer is updated at each time step, enabling the modeling of sequential patterns within the text. The final output layer consists of a fully connected Softmax classifier, which generates predictions for sentiment categories based on the hidden state of the final time step of the RNN. The RNN architecture is illustrated in Figure 2.

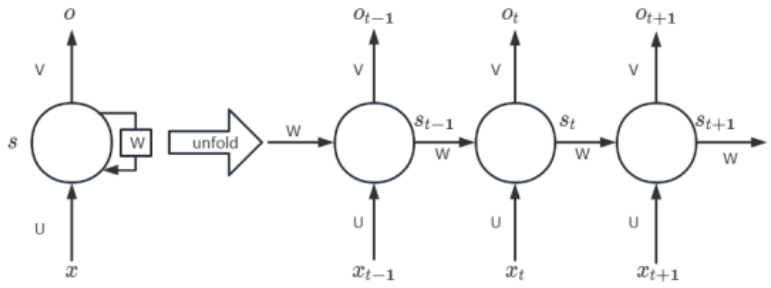

Figure 2: Architecture of the Recurrent Neural Network (RNN) Model

In contrast, the LSTM model is specifically designed to capture long-range dependencies and contextual relationships within text. The model architecture comprises an embedding layer followed by one or more LSTM layers that process sequential data while selectively preserving relevant information across time steps. The final hidden state is passed through a fully connected layer and a softmax function to generate binary classification output. The LSTM architecture is illustrated in Figure 3.

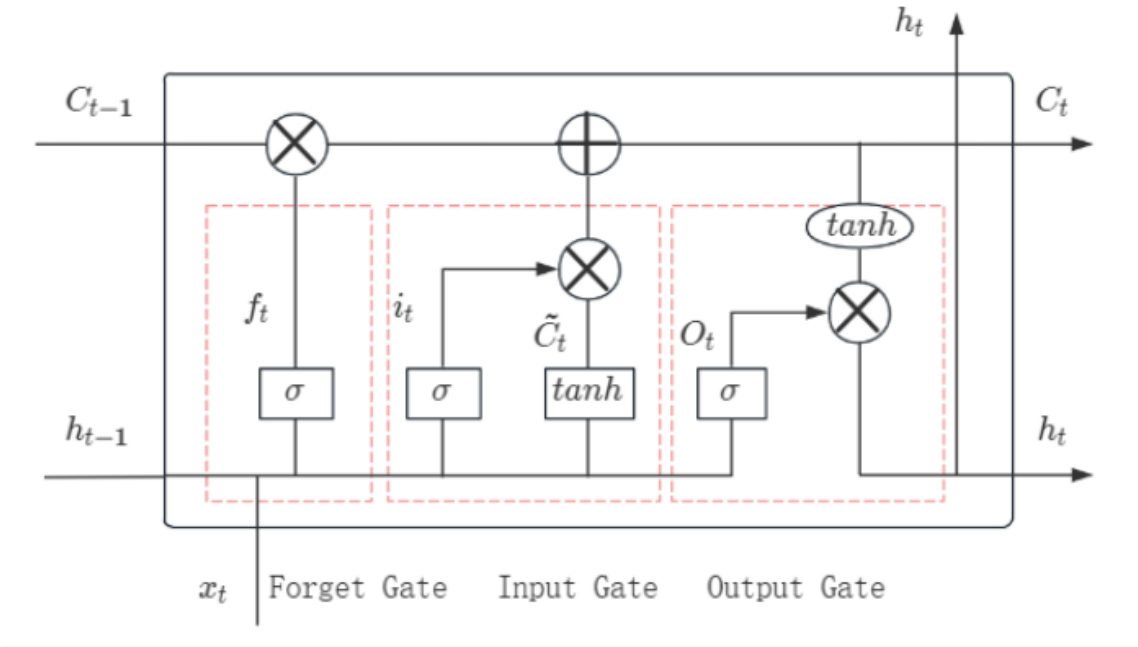

Figure 3: Architecture of the Long Short-Term Memory (LSTM) Network

3.3 Training Configuration

The LSTM model in this study was trained with a fixed input sequence length of 600 tokens and a batch size of 128. An initial learning rate of 0.0005 was set, and the RMSprop optimizer was selected for model optimization. The CrossEntropyLoss function, which performs well for classification tasks, was employed as the loss function during training. An early stopping strategy was implemented with a maximum of 10 training epochs. The actual training process terminated after 6 epochs when no further improvement in validation loss was observed. Additionally, mixed-precision training was enabled to accelerate computation, and key metrics—including training time, accuracy, recall, and F1-score for each epoch—were automatically logged to facilitate subsequent performance analysis and comparison.

The entire training process was conducted in a GPU environment, which significantly reduced the required training time compared to using a CPU. Performance metrics for each epoch were recorded, enabling continuous monitoring and comparison throughout the training cycles.

3.4 Evaluation Metrics

The effectiveness of the trained models was assessed using multiple performance indicators.

Table 1: Model Prediction Results

	Predicted positive	**Predicted negative**
Actual positive	True Positive (TP)	False Negative (FN)
Actual negative	False Positive (FP)	True Negative (TN)

According to Table 1, the four key evaluation metrics—Recall, F1-Score, Loss, and Test Accuracy—are defined as follows:

- Recall quantifies the model's ability to identify all actual positive instances, as expressed in

Equation (1):

$$Recall = \frac{TP}{TP + FN} \quad (1)$$

In the context of sentiment analysis for product reviews, Recall is employed to measure the model's capability to correctly identify negative reviews. Maintaining a high Recall rate is particularly crucial in e-commerce applications, as failing to detect negative reviews (false negatives) can prevent businesses from promptly identifying genuine product deficiencies. This, in turn, may adversely impact customer satisfaction, damage brand reputation, and cause companies to miss valuable opportunities for product and service improvement.

The F1-Score is defined as the harmonic mean of Precision and Recall, as illustrated in Equations (2) and (3):

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3)$$

It achieves a balance between the model's ability to identify positive instances (Recall) and its accuracy in labeling only the true positives (Precision). When dealing with imbalanced datasets or when the costs associated with both types of errors are high, a model with a high F1-score is both sensitive and precise, indicating it maintains high recall while keeping the false positive rate low.

- Test Accuracy measures the proportion of correct predictions (including both positive and negative instances) made by the model on the test dataset, as shown in Equation (4):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

A high accuracy indicates the model's capability to perform reliably on new, unseen data.

- Loss: This metric is computed by employing a loss function, such as CrossEntropyLoss, which penalizes inaccurate predictions, as formulated in Equation (5):

$$L = -\frac{1}{N} \sum_{i=1}^{N} [y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)] \quad (5)$$

y_i represents the true label (0 or 1) for the i-th sample， $\hat{y}_i$ denotes the predicted probability of the i-th sample belonging to the positive class， is the batch size.

The loss function effectively quantifies the divergence between the predicted probability distribution and the true distribution, and guides the optimization of model parameters through the backpropagation algorithm during training.

When comparing RNN and LSTM models, these metrics collectively provide a comprehensive assessment of their performance in sentiment classification tasks.

IV. RESULTS

This study presents a comprehensive evaluation of the Recurrent Neural Network (RNN) and the Long Short-Term Memory (LSTM) model for text sentiment detection, focusing on five key metrics: Recall, F1-score, Loss, Test Accuracy, and training time per epoch. Each of these metrics provides distinct insights into model effectiveness. The model was trained using an early stopping strategy with a maximum of 10 epochs. The actual training process terminated after 6 epochs when no further improvement in validation loss was observed.

Table 2: RNN Model Training Results on the Weibo_Senti_100k Dataset

Epoch	Recall	F1 Score	Loss	Test Accuracy	Train Time
1	0.821	0.819	0.383	0.826	32.24s
2	0.890	0.888	0.276	0.896	32.77s
3	0.887	0.886	0.296	0.891	32.34s
4	0.908	0.907	0.236	0.912	32.44s
5	0.908	0.907	0.227	0.908	32.50s
6	0.883	0.881	0.290	0.889	33.45s

The RNN model demonstrated overall performance improvement over the six training epochs, albeit with a slight decline in the final epoch. The Recall rate started at 0.821, fluctuated during training, rose to 0.908 by the fifth epoch, but decreased to 0.883 in the sixth epoch. The F1-score, representing the harmonic mean of precision and recall, showed a general increasing trend, reaching 0.907 in the fifth epoch before declining to 0.881 in the sixth epoch. The Loss value consistently decreased from 0.383 in the first epoch to 0.227 in the fifth epoch, indicating effective reduction in classification errors over time, though it increased again to 0.290 in the final epoch.

The test accuracy progressively improved from 0.826 to 0.912, then decreased to 0.889, suggesting enhanced generalization capability of the model. The training time per epoch remained largely consistent throughout the process.

Table 3: LSTM Model Training Results on the Weibo_Senti_100k Dataset

Epoch	Recall	F1 Score	Loss	Test Accuracy	Train Time
1	0.905	0.904	0.239	0.906	45.79s
2	0.910	0.909	0.220	0.910	46.12s
3	0.913	0.912	0.213	0.913	45.95s
4	0.916	0.915	0.210	0.916	45.97s
5	0.916	0.915	0.214	0.917	45.93s
6	0.915	0.914	0.220	0.915	46.17s

The LSTM model outperformed the RNN model across nearly all evaluation metrics. The Recall rate improved from 0.905 in the first training epoch to 0.915 by the sixth epoch, indicating enhanced capability in identifying true positive reviews. A similar upward trend was observed in reliability, with the F1-score increasing from 0.904 to 0.914.

Compared to the RNN, the LSTM exhibited consistently lower Loss values, decreasing from 0.239 in the first epoch to 0.220 in the sixth. This suggests that the LSTM's ability to model long-term dependencies in sequential text enables it to learn more meaningful representations from the data.

In summary, the test accuracy of the LSTM was generally higher than that of the RNN, rising from 0.906 to 0.915. Although the LSTM achieved superior accuracy and performance, this came at the cost of increased computational time, with each training epoch requiring approximately 12 seconds longer than the RNN. This demonstrates the LSTM's trade-off between performance gains and training efficiency.

V. CONCLUSION

This study conducted a systematic comparison of the performance between Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks in sentiment classification tasks. To ensure a fair comparison, both models were trained on the Weibo_Senti_100k dataset using identical preprocessing pipelines and training parameters. Experimental results demonstrate that the LSTM model significantly outperformed the RNN model across all key evaluation metrics, achieving a test accuracy of 91.5% while maintaining lower loss values. This robustly validates LSTM's superiority in capturing long-range dependencies within text sequences. Although the RNN model exhibited advantages in training efficiency, it demonstrated inferior performance in classification accuracy and generalization capability.

This research confirms the effectiveness of deep learning models, particularly the LSTM architecture, in product review sentiment analysis, providing both a theoretical foundation and practical reference for building efficient consumer sentiment analysis systems. Future research can be extended in the following directions: incorporating explainable AI techniques to enhance model transparency, and expanding from the current binary sentiment classification to more fine-grained sentiment dimensions (such as multi-level rating or aspect-based sentiment analysis), thereby better addressing the needs of real-world business applications.

VI. ACKNOWLEDGEMENTS

This research was supported by the Sichuan Science and Technology Program, Soft Science Project (No.2022JDR0076) and Sichuan Province Philosophy and Social Science Research Project (No. SC23TJ006). We also would like to thank the sponsors of Meteorological Information and Signal Processing Key Laboratory of Sichuan Higher Education Institutes of Chengdu University of Information Technology, and the fund of the Scientific and Technological Activities for Overseas Students of Sichuan Province (2022).

REFERENCES

[1]. Benzirar A, Hamidi M, Bouami F M. 2025. Building a speech emotion recognition system using RNN, GRU and LSTM. *International Journal of Speech Technology*, 28(3): 1-15. DOI:10.1007/S10772-025-10214-Z.

[2]. Yu X, Yang Y. 2025. Real-Time Chord Perception Using LSTM-Based RNN with Monitor Mechanism for Music Education and Analysis. *Journal of Circuits, Systems and Computers*. DOI:10.1142/S0218126625504675.

[3]. Kumar S, Pandey S, Singh R, et al. 2025. Leveraging neural networks (RNN LSTM) in enhancing energy efficiency and network lifetime in WSNs. *International Journal of Information Technology*. DOI:10.1007/S41870-025-02675-X.

[4]. Samrraie A A L, Abdalla M A, Alrawashdeh B A K, et al. 2025. Deep Learning Models Based on CNN, RNN, and LSTM for Rainfall Forecasting: Jordan as a Case Study. *Mathematical Modelling of Engineering Problems*, 12(7). DOI:10.18280/MMEP.120724.

[5]. Zelios V, Mastorocostas P, Kandilogiannakis G, et al. 2025. Short-Term Electric Load Forecasting Using Deep Learning: A Case Study in Greece with RNN, LSTM, and GRU Networks. *Electronics*, 14(14): 2820. DOI:10.3390/ELECTRONICS14142820.

[6]. Atlas G L, Arockiam D, Muthusamy A, et al. 2025. A modernized approach to sentiment analysis of product reviews using BiGRU and RNN based LSTM deep learning models. *Scientific Reports*, 15(1): 16642. DOI:10.1038/S41598-025-01104-0.