Volume 10, Issue 7, July– 2025 International Journal of Innovative Science and Research Technology
ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/25jul1564

Comparative Analysis of CNN and LSTM Neural Networks for Sentiment Classification on the Sentiment140 Dataset

Yiwen Tang[1]

[1]Eastlake High School Sammamish, United States

Publication Date: 2025/08/01

Abstract: Text sentiment analysis is of great help in mental health diagnosis. It can identify problems in early stages and actively intervene to prevent them from becoming serious. This study explores the application of deep learning techniques for sentiment analysis aimed at assessing mental health through text. In this paper, I use PyTorch to create a convolutional neural network (CNN) and a long short-term memory network (LSTM) and train these two neural networks based on the processed Sentiment140 dataset. Test Accuracy, Recall, F1 score, Total loss, and Training time to evaluate their performance. With a Test Accuracy of 87.42% as opposed to 81.25% for CNN, the results demonstrate that the LSTM model performs better than CNN across all evaluation metrics. Finally, I develop a web interface that enables users to enter text and receive sentiment analysis result based on trained LSTM model. This research can help improve mental health diagnosis and monitoring.

Keywords: *Sentiment Analysis, CNN, LSTM, Mental Health, PyTorch.*

How to Cite: Yiwen Tang (2025), Comparative Analysis of CNN and LSTM Neural Networks for Sentiment Classification on the Sentiment 140 Dataset. *International Journal of Innovative Science and Research Technology*, 10(7), 2602-2606. https://doi.org/10.38124/ijisrt/25jul1564

I. INTRODUCTION

In recent years, mental health has developed into a critical concern, as there is a growing awareness of its impact on overall well-being. At the same time, the increasing popularity of online platforms has given people easy ways to communicate digitally about their feelings and psychological states. Researchers can now efficiently identify emotional tone and deduce mental health conditions from online interactions by analyzing vast amounts of text data thanks to advancements in deep learning and natural language processing (NLP). Early detection and prompt intervention are greatly enhanced by this combination of artificial intelligence and mental health monitoring [1][2].

In this work, two neural network configurations—Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks—are trained and compared in order to create a sentiment analysis system that assesses a person's mental health. Trained models used the Sentiment140 dataset, which comprises labeled social media text, and the input is labeled as positive or negative. The goal is to identify the better model, based on accuracy, recall, F1 score, training time, and loss, and the preferred model is deployed in an operational website interface that conducts sentiment analysis.

This project expands the development of advanced technologies for mental health screening and analysis. The results could be useful for future applications in online support tools, therapy chatbots, and mental health platforms that support mental health conditions. The following are my primary contributions to this project:

- I preprocessed a sentiment dataset (e.g., Sentiment140) to ensure quality input for model training, including text normalization, tokenization, vocabulary building, and transforming the data into numerical formats suitable for neural network input. The dataset was also split into training, test, and validation sets to achieve accurate performance evaluations of the model.
- I trained two deep learning models, one Convolutional Neural Network (CNN) and one Long Short-Term Memory (LSTM) network, both of which were developed and trained with PyTorch. They have been trained with similar settings and tested with parameters such as accuracy, recall, F1 measure, loss function, as well as training time.
- I created a website interface to demonstrate the practical use of the learned sentiment classification model. Users can enter free text on the website and will be provided with immediate response of positive or negative sentiment. In its backend, it loads and deploys the top-performing model of that training step, making the research interactive.

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/25jul1564

II. RELATED WORKS

A well-known area of natural language processing, sentiment analysis finds applications in everything from public opinion extraction to product review classification. Previous methods used conventional machine learning algorithms like Naive Bayes, Support Vector Machines (SVM), and decision trees, as well as rule-based systems. More potent models, such as Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks, have become more well-known with the rise of deep learning because of their capacity to recognize intricate patterns in sequential data.

By treating sentences as collections of word embeddings, CNN, which was first created for image recognition, has been successfully applied to text classification tasks. CNNs are appropriate for sentiment analysis tasks because they are especially good at spotting local patterns like important phrases or emotionally charged word combinations.

LSTM networks, a specialized form of Recurrent Neural Networks (RNNs), excel in capturing temporal dependencies and contextual information in sequential data. Their ability to retain and influence long-range dependencies enables them to model subtle emotional nuances in language effectively.

Hybrid CNN–LSTM architectures have been used in recent research to predict mental health and analyze sentiment. For example, Islam et al. [3] used a CNN–LSTM model to detect depression on Twitter data with an accuracy of about 94.3%. When Wang et al. [4] compared CNN and LSTM models over training epochs in 2023, they discovered that LSTM consistently outperformed CNN in capturing sentiment information by a small margin. Rahman et al. [5] used a CNN–LSTM model on Twitter data from Bangladesh and achieved about 90% accuracy in detecting depression in a regional setting. In a similar vein, Zhang et al. [6] suggested a CNN–LSTM attention-based framework with SHAP-based interpretability to more accurately identify suicidal thoughts. Additionally, Chen and Liu [7] investigated an ensemble transformer–LSTM model for multiclass mental health prediction, encompassing disorders like PTSD, anxiety, and depression, and they reported strong performance in all categories.

In conclusion, direct comparisons of CNN and LSTM architectures under identical experimental setups are still comparatively understudied, even though numerous studies have shown the efficacy of CNN, LSTM, and hybrid models in sentiment analysis and mental health prediction. This paper fills this gap by building CNN and LSTM models in PyTorch with the same training parameters and a single preprocessing pipeline. Both models are trained on the Sentiment140 dataset and evaluated on multiple performance metrics. The following sections will describe the structure and principles of CNN and LSTM models in detail, laying the groundwork for a fair and comprehensive performance comparison.

III. METHODS

➢ *Data and Processing*

The dataset used in this study, Sentiment140, consists of labeled text samples indicating either positive or negative sentiment. All text inputs passed through a standard preprocessing workflow to provide data uniformity. The workflow involved transforming all characters to lowercase, elimination of punctuation and special characters, and tokenization of all sentences at the word level. To facilitate batch processing, all tokenized sequences were either padded or truncated to a predetermined length of 50 tokens. A vocabulary mapping was built that can map tokens to numerical indices, with the use of padding tokens that are specified to provide consistency in the sequence length.

To allow for supervised learning as well as model validation, the preprocessed data was divided into training and test subsets based on an 8:2 ratio, using 80% for training and 20% for testing. The subdivision ensured that the model has adequate data for the learning of patterns while maintaining a representative set for the evaluation of the model's performances. The processing flow is shown in Fig 1.

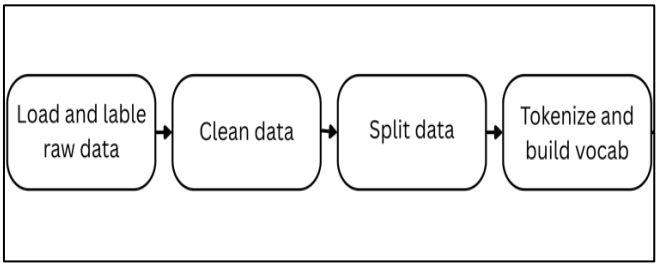

Fig 1 Preprocessing Data.

➢ *Models*

Two types of deep learning models were implemented using PyTorch: a CNN and an LSTM model. The CNN model began with an embedding layer, following that one-dimensional convolutional layer extracted local n-gram features in the text. The features passed through ReLU activation functions and max pooling functions; thus, the model can extract the strongest features in the sequence. The final output layer, a fully connected softmax classifier, produced the predictions of sentiment classes. CNN structure is shown as Fig 2.

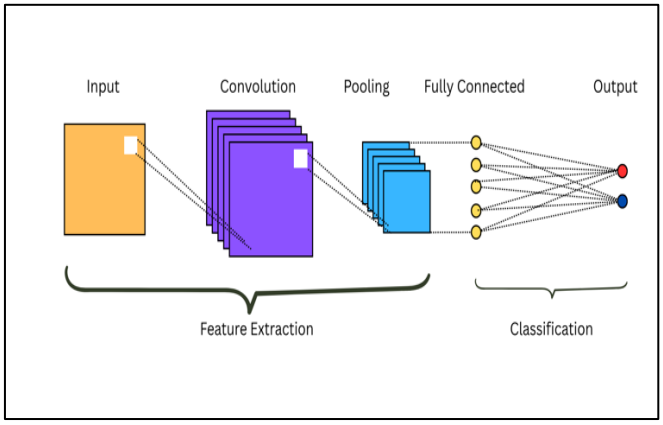

Fig 2 CNN Structure.

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/25jul1564

In contrast, the LSTM model was designed to capture long range dependencies and contextual relationships within the text. The model architecture included an embedding layer followed by one or more LSTM layers, which processed the sequential data and retained relevant information across time steps. The final hidden states were passed through a fully connected layer and softmax function to produce binary classification outputs. LSTM structure is shown as Fig 2.

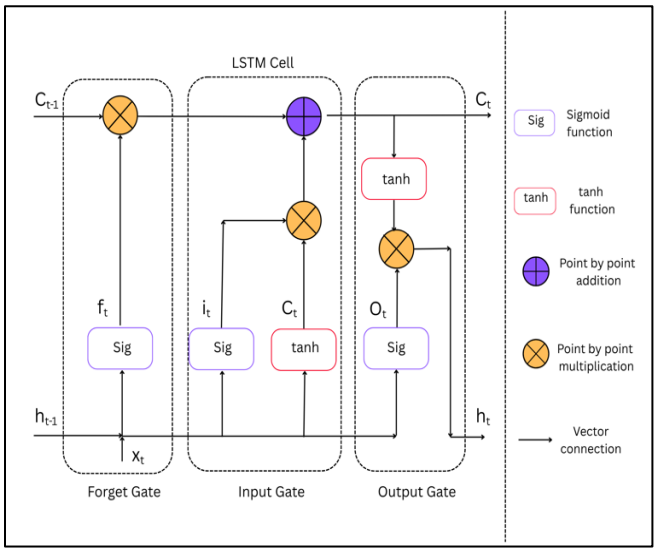

Fig 3 LSTM Structure.

➢ *Training Configurations*

Both the CNN and LSTM models were trained using a fixed input sequence length of 50 tokens and a batch size of 32. With an initial learning rate of 0.0005, the Adam optimizer was chosen due to its effectiveness and adaptive learning rate characteristics. The CrossEntropyLoss, a loss function that works well for multi-class classification tasks, was employed during training. Training epochs differed slightly according to the model's development. The LSTM would initially be trained over five epochs, whereas the CNN would typically be trained over three to five epochs. The LSTM training structure was further optimized based on experimental results because the LSTM model took a long time to train and showed little improvement in accuracy.

Due to hardware constraints, the entire training process was carried out in a CPU environment. The models recorded performance metrics for every epoch, allowing for consistent monitoring and comparison throughout training cycles.

➢ *Evaluating Metrics*

The evaluation of the effectiveness of the trained model used several performance metrics.

Table 1 Prediction Outcomes

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

Based on Table 1, 4 metrics of results Recall, F1 Score, Total Loss and Test Accuracy are defined as below.

- Recall quantifies the model's ability to identify all actual positive cases. Its formula is shown as equation (1).

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

The percentage of texts with negative or disturbing sentiment that the model correctly identifies is measured in sentiment detection. In mental health settings, high recall is crucial because overlooking a negative sentiment (false negative) could lead to incorrect diagnoses of people who might need assistance.

- F1 Score is the harmonic mean of precision and recall as equations (2)(3).

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

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

It balances the duality between the model's ability to identify positive cases (recall) and its accuracy in labeling only true positives (precision). When working with imbalanced datasets or when both kinds of errors have intolerable repercussions, a model with a high F1 Score is both sensitive and precise, meaning it has a high recall and a low false positive rate.

- Test Accuracy measures the proportion of correct predictions (both positive and negative) made by the model on the test dataset, shown in equation (4).

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

A high accuracy shows the model's ability to perform reliably on new data.

- Total Loss is the sum of prediction errors over all samples in a training epoch, shown in equation (5).

$$Total\ Loss\ (Epoch) = \sum_{i=1}^{N} Loss_i \quad (5)$$

In this formula, *N = number of batches* and $Loss_i$ *= value returned for batch i.*

It is calculated by penalizing inaccurate predictions with a loss function like CrossEntropyLoss. Effective learning and enhanced model performance are indicated by a declining total loss over epochs. It shows how well the model has done at reducing classification errors throughout training.

When comparing the CNN and LSTM models, these metrics offer a thorough picture of how well the models perform in sentiment classification.

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/25jul1564

IV. RESULTS

This report presents a comprehensive evaluation of the Convolutional Neural Network (CNN) model used for sentiment detection in text, focusing on five metrics: Recall, F1 Score, Total Loss, Test Accuracy, and Training Time per Epoch. Each of these metrics provides unique insights into the model's effectiveness.

Table 2 Training Results of the CNN-Based Sentiment Assessment Model

Epoch	Recall	F1 Score	Total Loss	Test Accuracy	Train Time
1	0.8393	0.8060	15562.75	79.75%	3 hours 19 minutes
2	0.8016	0.8075	14151.33	80.84%	5 hours 30 minutes
3	0.7874	0.8056	13604.37	80.95%	5 hours 32 minutes
4	0.7843	0.8051	13184.25	80.97%	4 hours 09 minutes
5	0.8047	0.8114	12790.82	81.25%	3 hours 59 minutes

The CNN model showed steady performance improvement over five training epochs. The recall value started at 0.8393 and showed fluctuations before improving to 0.8047 by the final epoch. Its F1 Score, a harmonic mean of precision and recall, remained relatively stable, peaking at 0.8114 by the final epoch. The total loss consistently decreased from 15,562.75 in Epoch 1 to 12,790.82 by Epoch 5, suggesting that the model effectively minimized classification errors over time.

The test accuracy gradually improved from 79.75% to 81.25%, indicating a moderate gain in the model's generalization capabilities. However, training time for each epoch varied, with the fastest taking 3 hours and 19 minutes and the longest exceeding 5.5 hours, showing that CNN models are moderately fast but can still require substantial training time depending on system resources.

Table 3 Training Results of the LSTM Model Based on Sentiment140 Dataset

Epoch	Recall	F1 Score	Total Loss	Test Accuracy	Train Time
1	0.7900	0.7949	16222.67	79.63%	4 hours 55 minutes
2	0.8257	0.8307	13954.69	83.19%	4 hours 19 minutes
3	0.8422	0.8467	12885.04	84.76%	3 hours 11 minutes
4	0.8556	0.8603	11923.76	86.11%	3 hours 06 minutes
5	0.8690	0.8735	10976.75	87.42%	3 hours 06 minutes

In almost every evaluation metric, the LSTM model performed better than the CNN model. The recall increased significantly from 0.7900 in Epoch 1 to 0.8690 by Epoch 5, indicating strong improvements in identifying all true positive cases. Reliability increased as the F1 Score increased from 0.7949 to 0.8735 in a similar upward trend.

Compared to the CNN, the overall loss dropped more abruptly, from 16,222.67 in Epoch 1 to just 10,976.75 in Epoch 5. This implies that because of its ability to model long-range dependencies in sequential text, the LSTM was able to learn more meaningful representations from the data.

Above all, the test accuracy increased significantly from 79.63% to 87.42%, surpassing the CNN by over 6%. Despite its higher accuracy and performance, the LSTM's training time was comparable to CNN, ranging from 3 to 5 hours per epoch, showing it was optimized efficiently in later epochs.

Using Flask, a basic web interface was created to show the trained LSTM model's usefulness. When users enter free-form text, like journal entries or diary reflections, the website classifies the sentiment as either positive or negative. Users can anonymously submit daily thoughts and receive instant sentiment feedback through this interface, which mimics a real-world mental health screening tool. The lstm_model.pth file and a saved vocabulary (vocab.pkl) are loaded by the back end to preprocess the text and perform PyTorch inference.

- *Here are Some Results of the Sentiment Analysis Web Interface:*

Fig 4 Sentiment Analysis Web Interface-Positive Result

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/25jul1564

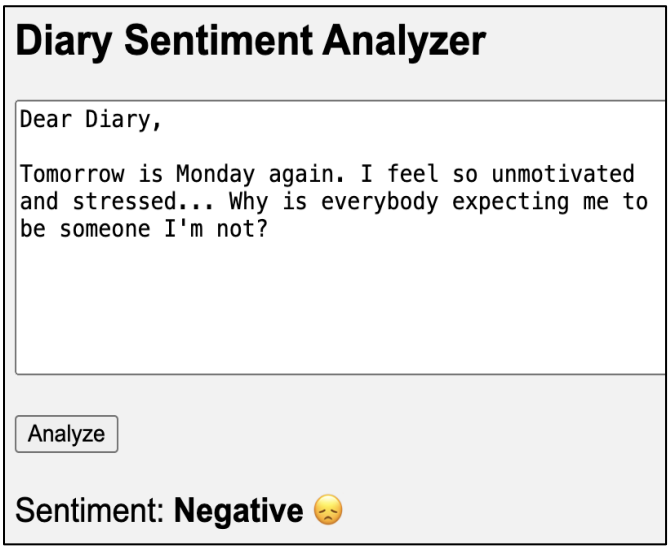

Fig 4 Sentiment Analysis Web Interface-Negative Result

Teenagers or young adults looking for private, non-clinical emotional feedback will find the website's intuitive interface and emphasis on accessibility and ease of use ideal.

V. CONCLUSION

This study presents a comparison of two deep learning models for sentiment classification: CNN and LSTM. Both models were trained using the same preprocessing and training parameters on the Sentiment140 dataset to ensure a fair comparison. By outperforming all key metrics, including a significantly higher test accuracy (87.42%) and lower total loss, the LSTM model proved that it could capture long-term dependencies in text. The CNN model did worse in terms of accuracy and generalization, despite being faster at times. These findings imply that deep learning models, especially LSTM architectures, can aid in the creation of sophisticated instruments for mental health monitoring and analysis. By incorporating understandable artificial intelligence elements and managing more complex emotional categories than binary sentiment, future research can build upon this foundation.

REFERENCES

[1]. Kumar, A., & Sharma, R. (2023). Deep learning approaches for depression detection from social media texts: A comprehensive survey. Journal of Biomedical Informatics, 136, Article 104275.
[2]. Zhang, T., Lin, Y., & Yu, S. (2023). Emotion fusion for mental health classification on social media. IEEE Transactions on Affective Computing. Advance online publication.
[3]. Islam, M. R., Kabir, M. A., & Ahmed, M. M. (2022). Depression detection from social media using hybrid CNN–LSTM model. Journal of Affective Computing, 13(2), 130–139.
[4]. Wang, Y., Chen, Z., & Sun, Q. (2023). Comparative study of CNN and LSTM for sentiment classification. Proceedings of the 2023 International Conference on Artificial Intelligence Applications, 88–94.
[5]. Rahman, T., Hossain, M. N., & Akter, S. (2024). Mental health detection from Bangla tweets using deep learning models. Asian Journal of Computer Science and Information Technology, 15(1), 22–29.
[6]. Zhang, L., Li, H., & Zhou, X. (2023). Attention-based CNN–LSTM with SHAP interpretability for suicidal ideation detection. IEEE Access, 11, 27432–27445.
[7]. Chen, R., & Liu, Y. (2022). An ensemble transformer–LSTM approach for multiclass mental health prediction. Expert Systems with Applications, 198, 116842.