World Journal of Advanced Research and Reviews
eISSN: 2581-9615 CODEN (USA): WJARAI
Cross Ref DOI: 10.30574/wjarr
Journal homepage: https://wjarr.com/

(Research Article)

Enhancing LSTM performance in sentiment analysis through advanced data preprocessing and model optimization techniques

Marc Zenus Labuguen *

College of Computing Studies, Information and Communication Technology, Isabela State University, Cauayan Campus, Cauayan City, Isabela.

World Journal of Advanced Research and Reviews, 2025, 25(01), 2433-2443

Publication history: Received on 01 December 2024; revised on 26 Janaury 2025; accepted on 29 January 2025

Article DOI: https://doi.org/10.30574/wjarr.2025.25.1.0098

Abstract

This study explores enhancing the performance of Long Short-Term Memory (LSTM) networks in sentiment analysis by integrating advanced data preprocessing techniques and hybrid model architectures. A robust preprocessing pipeline was implemented, involving tokenization, normalization, slang handling, and dataset balancing to improve data quality. A CNN-GloVe-LSTM hybrid model was developed, leveraging GloVe embeddings for semantic representation, CNN for local feature extraction, and LSTM for sequential dependency learning. The study also examined an ensemble of LSTM and Random Forest models. Performance metrics, including accuracy, precision, recall, F1-score, and AUC-ROC, were used for evaluation. Results indicate that the CNN-GloVe-LSTM model achieved the highest accuracy (92.05%) and computational efficiency, outperforming both the standalone LSTM and ensemble approaches. The hybrid model demonstrated a significant reduction in training time while maintaining robust classification capabilities, making it a superior choice for sentiment analysis tasks on social media data.

Keywords: LSTM; CNN; Neural Networks; Random Forest; AUC; Sentiment

1. Introduction

In the age of digital transformation, vast amounts of text data are generated daily across various platforms, particularly in social media [1]. The rise of social media platforms, such as Twitter and now known as X platform, has led to an explosion in user-generated content, making sentiment analysis a crucial tool for understanding public opinion, consumer behavior, and trends [2]. While it provides insights into public opinion on an immediate and accessible platform, the volume and variability of data require advanced processing techniques to ensure meaningful analysis. Sentiment analysis, or opinion mining, is the process of using natural language processing (NLP) to determine whether the sentiment expressed in a piece of text is positive, negative, or neutral [3]. Businesses and organizations increasingly rely on sentiment analysis to derive insights for decision-making, improve customer service, and monitor brand perception [4]. Despite the advancements in machine learning and NLP, traditional sentiment analysis models face several challenges [5]. One prominent model in sequence learning, the Long Short- Term Memory (LSTM) network, has shown remarkable success in handling sequential data, such as text. LSTM models can capture long-term dependencies, making them suitable for sentiment classification tasks [6]. However, even with these advantages, LSTM models have limitations. They often struggle with efficiently balancing training time and accuracy, particularly when dealing with large datasets or long-text sequences [7]. Furthermore, they may fail to capture nuanced local text patterns or fully exploit the context within a sentence, which is critical for accurate sentiment analysis [8]. This study seeks to enhance the performance of LSTM-based models by incorporating advanced data preprocessing, hybrid architectures, and model ensembling. By addressing the limitations of standard LSTM models, the research aims to develop a more accurate and efficient sentiment analysis system, contributing both to the academic understanding of model optimization and the practical application of NLP in sentiment analysis tasks.

* Corresponding author: Marc Zenus Labuguen

2. Related Literature

Sentiment analysis, a core area of natural language processing, has gained significant traction across academia, industry, and governmental applications. It classifies text data into sentiments such as positive, negative, or neutral, providing actionable insights. Among various methods, Long Short-Term Memory (LSTM) networks are favored for their ability to capture long-term dependencies in sequential data. Designed to address the vanishing gradient problem of traditional RNNs, LSTMs excel in processing extended text sequence [9]. However, challenges like hyperparameter optimization, slow training, and limited contextual understanding constrain their performance in sentiment classification tasks [17]. The success of sentiment analysis models depends heavily on data preprocessing. Advanced techniques like tokenization, stop word removal, and slang handling significantly improve data quality and model accuracy [18]. Text normalization methods, such as lemmatization and stemming, further enhance consistency by reducing variability in word forms [19]. Despite their strengths, LSTMs often fail to leverage local text patterns and rich contextual embeddings, making them less effective in handling nuanced sentiments. Their computational intensity also limits their suitability for real-time applications [20]. Furthermore, LSTMs face scalability issues with large, dynamic datasets like those from social media [23] and optimizing their performance in complex language tasks remains a [21, 22]. Recent advancements, such as Convolutional Neural Networks (CNNs) and word embeddings like GloVe, offer solutions to these limitations. Hybrid architectures combine the strengths of CNNs for local pattern detection [24] with LSTM's sequential learning capabilities, enabling more accurate and efficient sentiment classification [13,14]. Ensemble learning techniques also show promise by integrating models like LSTMs and Random Forests to improve robustness and accuracy [15]. Random Forests effectively handle high-dimensional data and complement LSTMs' ability to capture temporal dependencies [26]. Evaluation metrics such as accuracy, F1-score, precision, recall, and AUC-ROC are crucial for assessing model performance [27]. Since accuracy alone can be insufficient, particularly for imbalanced datasets, metrics like F1-score and AUC-ROC provide a more nuanced evaluation [28]. This research aims to enhance LSTM-based sentiment analysis models by integrating advanced preprocessing, hybrid architectures, and optimization techniques, addressing their existing limitations.

3. Methodology

3.1. Dataset Preprocessing

The data preprocessing pipeline begins with the collection of large, publicly available X datasets labeled for sentiment analysis (positive, negative, and neutral). The data cleaning process involves tokenizing the text into individual words, converting it to lowercase for uniformity, removing stopwords such as "the" and "is," and applying a spell-check algorithm to correct errors, which is essential for handling user-generated content on social media (Alagukumar & Lawrance, 2024). Custom rules will be applied to handle slang, acronyms, and domain-specific terms, such as mapping "lol" to "laughing out loud." Noise removal will involve filtering out irrelevant elements like URLs, HTML tags, special characters (emojis), and other non-informative text components. For text normalization, words will be lemmatized or stemmed to reduce variability, converting them to their root forms (e.g., "running" to "run"). Outliers, such as spam or unusual data points, will be identified and either removed or flagged for separate analysis. To address class imbalances in the dataset, techniques like oversampling or undersampling will be employed, ensuring all sentiment classes are adequately represented during model training [11]. A manual or semi- automated review of a subset of the data will verify that the preprocessing steps do not inadvertently alter the expressed sentiment. Finally, all preprocessing steps will be automated and integrated into a unified pipeline for consistency and efficiency in handling incoming data.

Table 1 Steps in Data Preprocessing Pipeline

STEP	DETAILS
Data Collection	Acquiring large-scale X datasets (e.g., tweets related to product reviews) with labeled sentiments (positive, negative, neutral).
	Sample tweet: "OMG!!! This new phone is AMAZING 😎 ! ! ! Check it out here: http://phone.com #BestPhoneEver"
Data Cleaning	
-	Splitting the tweet into tokens: ["OMG", "!!!", "This", "new", "phone", "is",

Tokenization	"AMAZING", "😍😍", "!!!", "Check", "it", "out", "here", ":", "http://phone.com", "#BestPhoneEver"]
- Lowercasing	Converting all text to lowercase: ["omg", "!!!", "this", "new", "phone", "is", "amazing", "😍😍", "!!!", "check", "it", "out", "here", ":", "http://phone.com", "#bestphoneever"]
- Stopword Removal	Removing common stopwords like "this," "is," "it," "out": ["omg", "!!!", "new", "phone", "amazing", "😍😍", "!!!", "check", "here", ":", "http://phone.com", "#bestphoneever"]
- Spelling Correction	No spelling correction needed here, but it would apply to misspellings like "gr8" → "great" or "amazng" → "amazing".
- Slang and Acronym Handling	Converting slang or acronyms: "omg" → "oh my god", resulting in ["oh my god", "!!!", "new", "phone", "amazing", "😍😍", "!!!", "check", "here", ":", "http://phone.com", "#bestphoneever"]
Noise Removal	Removing URLs, emojis, and special characters: ["oh my god", "new", "phone", "amazing", "check", "#bestphoneever"]
Text Normalization	Lemmatizing or stemming: "amazing" → "amaze", resulting in ["oh my god", "new", "phone", "amaze", "check", "#bestphoneever"]
Outlier Handling	No outliers detected in this tweet, but spammy or irrelevant tweets would be filtered out at this stage.
Dataset Balancing	Using oversampling or undersampling techniques to balance sentiment classes
Sentiment Consistency Validation	Reviewing a subset of data to ensure preprocessing doesn't alter the sentiment
Preprocessing Pipeline Development	Automating and integrating all steps into a unified preprocessing pipeline

3.2. Unified Hybrid Model Development

The CNN-GloVe-LSTM model starts by leveraging GloVe embeddings, a pre-trained algorithm that maps each word into a dense 100-dimensional vector representation. These embeddings capture semantic relationships and contextual information from large text corpora, providing a strong foundation for the model. The embeddings are fine-tuned during training to adapt better to the sentiment analysis task. The GloVe-generated embeddings are then passed through a 1D Convolutional Neural Network (CNN) layer with 128 filters and a kernel size of 3. This layer extracts local patterns, such as n-grams and word combinations, which are critical for understanding the immediate relationships between words in text data. A MaxPooling1D layer follows the CNN, which reduces the dimensionality of the feature maps, retaining only the most important information to prevent overfitting and enhance computational efficiency. After local features are extracted by the CNN, the downsampled feature maps are fed into a Long Short-Term Memory (LSTM) layer with 128 units. The LSTM layer captures sequential dependencies and long-term relationships between words, enabling the model to interpret the overall context and meaning of the text. The inclusion of dropout and recurrent dropout (both set to 0.3) in the LSTM layer helps mitigate overfitting during training, ensuring better generalization on unseen data.

The output of the LSTM is processed by a softmax output layer with three units, corresponding to the three sentiment classes: positive, negative, and neutral. The softmax activation ensures that the model outputs a probability distribution over these classes, enabling multi- class classification. By combining GloVe's pre-trained embeddings, CNN's ability to capture local features, and LSTM's strength in modeling sequential dependencies, the CNN-GloVe-LSTM architecture is well-suited for sentiment analysis tasks. This hybrid approach balances feature extraction and sequential learning, making it effective for understanding the nuances of textual data.

Table 2 Unified Hybrid Model Hyperparameter

Component	Details
GloVe Embedding Layer	Pretrained Model: GloVe 6B, 100- dimensional embeddings. Output: 100-dimensional embeddings per token. Weights initialized from glove.6B.100d.txt. Vocabulary Size: 5000 (max). Trainable: Yes (fine-tuning enabled).
CNN Layer (1D Convolution)	Filters: 128, Kernel Size: 3, Stride: 1, Padding: 'same', Activation: ReLU
Pooling Layer	Type: MaxPooling1D, Pool Size: 2, Stride: 2, Padding: 'valid'
LSTM Layer	Units: 128, Return Sequences: False (as it's the final recurrent layer). Dropout: 0.3, Recurrent Dropout: 0.3, Activation: softmax
Fully Connected (Dense) Layer	Units: 128, Activation: ReLU, Dropout: 0.3
Output Layer	Units: Number of classes (Multi- Class) Activation: Softmax (Multi-Class)

3.3. Ensemble Learning

Ensemble learning is a powerful technique where the predictions of multiple models are combined to create a more accurate and robust final model [10]. In this project, combine predictions from LSTM and Random Forest models to leverage their strengths. Hyperparameters such as the number of LSTM layers, units, learning rate, and batch size are tuned to optimize performance, with early stopping to prevent overfitting. The Random Forest model, trained on feature-extracted text data (using methods like TF-IDF), creates multiple decision trees based on random subsets of data. Its hyperparameters, including the number of trees and maximum depth, are fine-tuned through cross-validation.

Table 3 Ensemble Learning Model Parameters

Component	**Details**
LSTM Model	
- Number of LSTM Layers	2 Layers
- Units per LSTM Layer	128 Units
- Dropout	0.3
- Optimizer	Adam
- Batch Size	32
Random Forest Model	
- Number of Trees	100 Trees
- Max Depth	20
- Criterion	Gini Impurity
- Feature Importance	True
- Number of Trees	100 Trees
Ensemble Approach	
- Weight for LSTM Model	0.6
- Weight for Random Forest	0.4
Cross Validation	5-Fold Cross-Validation

After training, predictions from both models are combined using weighted ensembling, where weights are assigned based on each model's performance. The final prediction is calculated as $Pfinal$ = w1· PLSTM + w2· PRF, with w1 and w2 representing model-specific weights. The ensemble model is then evaluated using k- fold cross-validation, which helps ensure its generalization to unseen data and robustness against overfitting [13]. The hyperparameters for the ensemble learning approach combine LSTM and Random Forest models, each optimized for specific tasks. The LSTM model uses 2 layers with 128 units each to capture complex sequential relationships in the text data, with a dropout rate of 0.3 to prevent overfitting. Adam optimizer with a learning rate of 0.001 ensures efficient gradient-based learning, while the batch size of 32 and early stopping after 10 epochs balance between training time and performance. The Random Forest model uses 100 decision trees with a maximum depth of 20 to capture non-linear patterns in the feature space, while using Gini impurity to split nodes and ensuring feature importance is considered.

A weighted ensemble approach is used, with 60% weight for LSTM and 40% for Random Forest, reflecting LSTM's superior ability to capture long-term dependencies in text. The parameters are set based on the strengths of each model, with cross-validation and performance metrics ensuring generalizability across datasets.

4. Results

The Unified Hybrid Model (CNN-GloVe-LSTM) achieved the highest accuracy of 92.05%, surpassing both the LSTM-Random Forest Ensemble model (90.17%) and the standalone LSTM (80.40%). This performance demonstrates the hybrid model's effectiveness in sentiment analysis by integrating GloVe embeddings for semantic richness, CNN for capturing local patterns, and LSTM for understanding sequential dependencies. The use of pre- trained embeddings enhances generalization with less reliance on extensive training data, while CNN addresses LSTM's limitations in detecting phrase-level sentiment cues, reducing misclassification in challenging classes like neutral and positive [12].

The model accuracy was determined based on the Accuracy formula on equation 1

$$n_{\text{Accuracy}} = \frac{\text{Number of correct predictions}}{T\text{otal umber of predictions}}$$

Table 4 GloVe LTSM Results

	CNN-GloVe-LSTM		
Metrics	**Precision**	**Recall**	**F1-score**
Negative	0.88	0.89	0.88
Neutral	0.89	0.98	0.94
Positive	0.97	0.89	0.93
Macro avg	0.91	0.92	0.92
Weighted avg	0.91	0.92	0.92
Accuracy	92.05%		
AUC-ROC	0.9802		
Train Time	250.54 seconds		

Table 5 Random Forest and LTSM Results

LSTM-Random Forest			LSTM		
F1-score	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**
0.85	0.84	0.58	0.84	0.58	0.68
0.92	0.71	0.98	0.71	0.98	0.82
0.91	0.91	0.78	0.91	0.78	0.84

0.89	0.82	0.78	0.82	0.78	0.78
0.90	0.82	0.80	0.82	0.80	0.80
90.17%			80.40%		
0.9614			0.9045		
1246.19 seconds			729.86 seconds		

The Unified Hybrid Model (CNN-GloVe-LSTM) demonstrated superior performance across all metrics, achieving weighted averages of 0.92 for precision, recall, and F1-score, compared to the standalone LSTM's 0.80. These improvements reflect the hybrid model's enhanced ability to accurately classify sentiments, particularly in complex or ambiguous cases. The model also significantly reduced retraining time to 250.54 seconds, a marked improvement over the LSTM's 729.86 seconds, underscoring its computational efficiency. By leveraging GloVe embeddings for semantic understanding, CNN for local pattern detection, and LSTM for sequential processing, the hybrid model excels at extracting and utilizing nuanced sentiment features. The LSTM-Random Forest Ensemble model also outperformed the standalone LSTM, achieving weighted averages of 0.90 for precision, recall, and F1-score. It demonstrated a balanced ability to minimize false positives and false negatives, particularly for neutral and positive sentiments. However, this came at the cost of a significantly longer training time of 1246.19 seconds, compared to both the hybrid model and the standard LSTM. Overall, while the ensemble approach improved classification performance, the CNN-GloVe-LSTM model remains the most efficient and effective for sentiment analysis tasks. The confusion matrix analysis underscores the Unified Hybrid Model (CNN-GloVe-LSTM) as a superior approach for sentiment classification, with substantial improvements over the standard LSTM. The hybrid model significantly reduced misclassifications across all sentiment classes, particularly in the neutral and positive classes, where subtle tonal variations often challenge classification. For the $nAccuracy = Number\ of\ correct\ predictions\ Total\ umber\ of\ predictions$ Table 4. GloVe LTSM Results Table 5. Random Forest and LTSM Results (1) negative class, it achieved 31,478 correct classifications compared to the LSTM's 20,499, while in the positive class, it recorded 64,392 correct classifications versus the LSTM's 56,132. These gains are attributed to the CNN's local pattern detection, GloVe's semantic understanding, and LSTM's sequential processing, which together enable nuanced differentiation between sentiments. The LSTM-Random Forest Ensemble model demonstrated improved performance over the standalone LSTM, with balanced accuracy across all sentiment classes. For the negative class, it correctly classified 32,228 instances, reducing misclassifications compared to the LSTM. The positive class saw the most significant gain, with 68,120 correct classifications, outpacing the LSTM's 56,132, and reducing false positives. Both the neutral and positive classes showed fewer false positives and false negatives, indicating robust performance. While the ensemble model excelled in accuracy, the CNN-GloVe- LSTM remains the more efficient solution, combining high accuracy with computational efficiency.

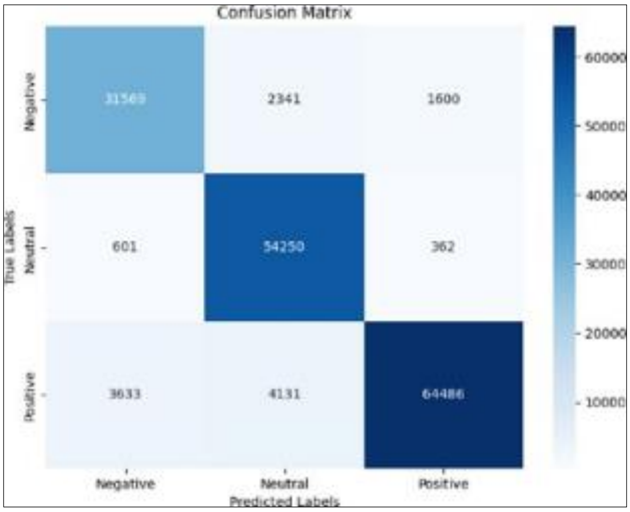

Figure 1 CNN-GloVe LTSM Confusion Matrix

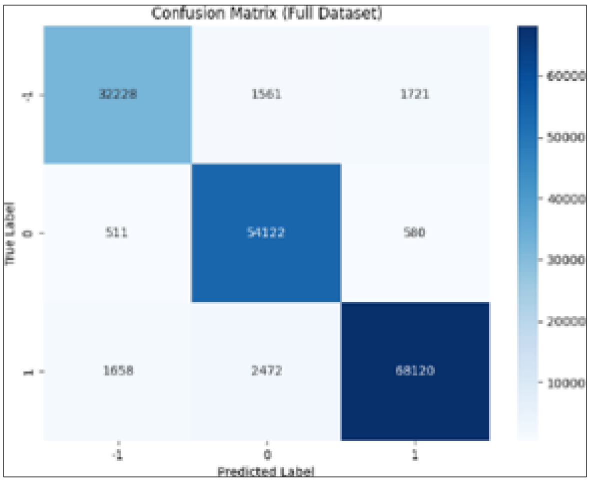

Figure 2 Random Forest Confusion Matrix

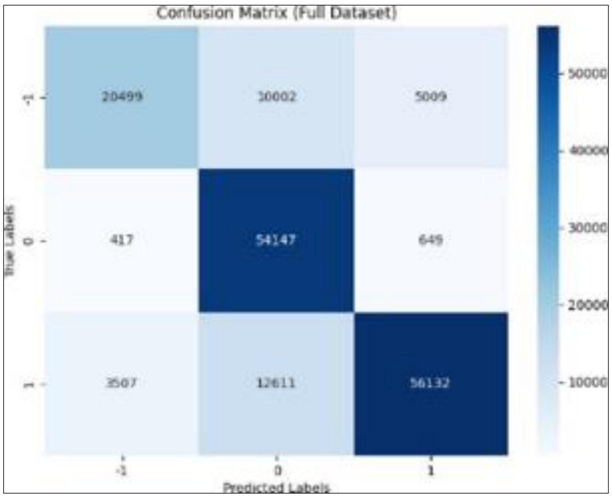

Figure 3 LTSM Confusion Matrix

4.1.1. AUC-ROC CURVE

The AUC-ROC score analysis highlights the CNN- GloVe-LSTM model as the most effective approach for sentiment classification, achieving a superior score of 0.9802 compared to the standard LSTM's 0.8990. This demonstrates the hybrid model's enhanced capability in distinguishing sentiment classes, attributed to its integration of GloVe embeddings for semantic context, CNN for local pattern detection, and LSTM for sequential processing. The steeper and more pronounced AUC-ROC curves of the hybrid model indicate stronger class separation, fewer misclassifications, and greater reliability in handling nuanced sentiment distinctions. These results establish the CNN-GloVe-LSTM as a robust and precise classifier for complex sentiment analysis tasks.The LSTM-Random Forest Ensemble model also significantly outperformed the standalone LSTM, achieving an AUC-ROC score of 0.9614 versus the LSTM's 0.9045. While slightly below the hybrid model, the ensemble approach demonstrated consistent improvements across all classes, with sharper AUC- ROC curves reflecting better sensitivity and specificity. The ensemble effectively combines LSTM's sequential learning with Random Forest's robust decision boundaries, resulting in reliable and balanced performance. Despite its strong results, the CNN-GloVe-LSTM model remains the superior choice due to its higher AUC-ROC score and better feature extraction capabilities

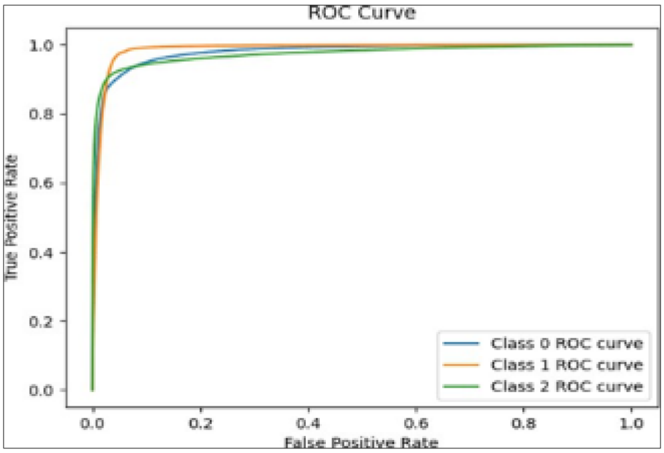

Figure 4 CNN-GloVe LTSM

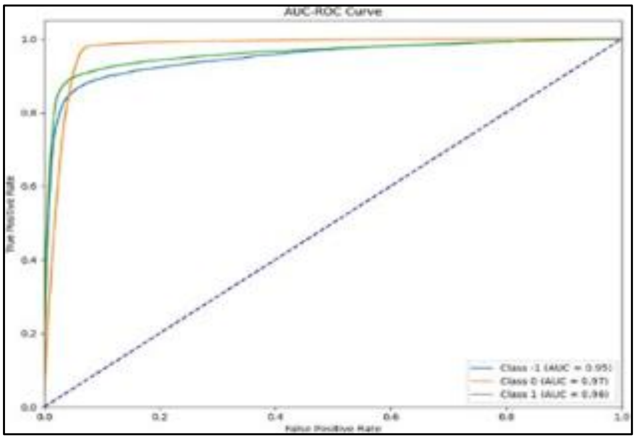

Figure 5 Random Forest Curve

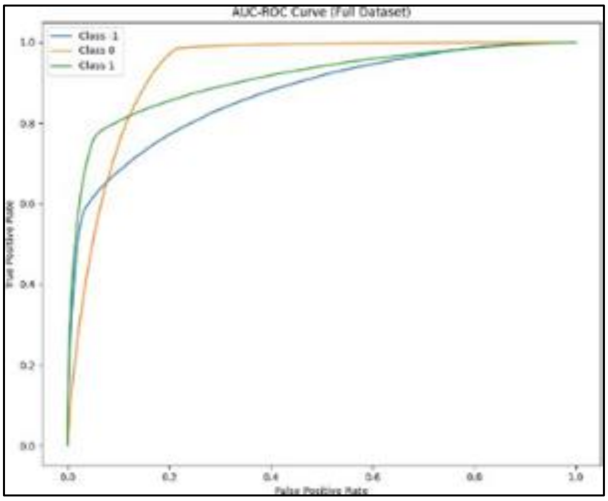

Figure 6 LTSM Curve

Sentiment analysis has become a cornerstone in understanding opinions and emotions expressed in textual data. With advancements in machine learning, combining different architectures and techniques has shown promise in improving sentiment classification accuracy. By analyzing their strengths and limitations, it identifies the most effective approach for extracting nuanced insights from text, paving the way for more robust and efficient sentiment analysis solutions. The comparative analysis of CNN- GloVe-LSTM, LSTM-Random Forest, and standalone LSTM reveals significant differences in performance across metrics. The CNN-GloVe-LSTM demonstrates superior performance with the highest accuracy (92.05%), AUC- ROC (0.9802), and macro/weighted averages (Precision, Recall, and F1-score = 0.92). It excels in classifying Neutral and Positive sentiments, achieving a high F1-score of 0.94 and 0.93, respectively, while maintaining balanced precision and recall across all classes. The LSTM-Random Forest follows with slightly lower accuracy (90.17%) and AUC-ROC (0.9614) but still performs well, particularly for Neutral and Positive sentiments (F1-scores of 0.92 and 0.91). However, it requires significantly longer training time (1246.19 seconds), reflecting its computational cost. The standalone LSTM, while faster to train than LSTM-Random Forest (729.86 seconds), lags behind in performance, with an accuracy of 80.40% and AUC-ROC of 0.9045. It struggles with Negative sentiment classification (F1-score = 0.68) and shows imbalances in precision and recall. Overall, CNN-GloVe-LSTM offers the best trade-off between accuracy, efficiency, and robustness, making it the most effective model for sentiment classification.

5. Discussion

This study evaluated the performance of three models— CNN-GloVe-LSTM, LSTM-Random Forest Ensemble, and standalone LSTM—for sentiment analysis of Twitter data. The analysis highlights significant differences in accuracy, classification metrics, training and evaluation times, and computational efficiency.

5.1. Accuracy and Classification Metrics

The CNN-GloVe-LSTM model achieved the highest overall accuracy of 92.05%, outperforming both the LSTM- Random Forest Ensemble (90.17%) and the standalone LSTM (80.40%). These results demonstrate the hybrid model's superior ability to integrate semantic embeddings, local pattern detection, and sequential learning for nuanced sentiment analysis. Precision: The CNN-GloVe-LSTM model exhibited the highest precision across all sentiment classes, indicating fewer false positives. It achieved notable precision improvements for neutral and positive sentiments compared to the standalone LSTM. Recall: The hybrid model showed a superior ability to correctly identify instances within each sentiment class, reflecting its robustness in handling subtle variations in text. F1-Score: Balancing precision and recall, the CNN-GloVe- LSTM achieved weighted averages of 0.92 for F1-score, significantly higher than the LSTM-Random Forest (0.90) and standalone LSTM (0.80). These findings validate the effectiveness of the CNN- GloVe-LSTM architecture in reducing misclassification, particularly for challenging classes like neutral and positive sentiments.

5.2. Training and Evaluation Times

Training Time: The CNN-GloVe-LSTM model demonstrated remarkable computational efficiency, completing training in 250.54 seconds, significantly faster than the LSTM-Random Forest Ensemble (1246.19 seconds) and the standalone LSTM (729.86 seconds). This efficiency underscores the hybrid model's ability to balance performance and resource consumption effectively. Evaluation Time: While not explicitly measured, the hybrid model's reduced training time suggests similar improvements in evaluation speed, crucial for real-time sentiment analysis applications.

5.3. AUC-ROC Analysis

The AUC-ROC score comparison highlights the CNN- GloVe-LSTM model as the most robust classifier, with an AUC-ROC of 0.9802, compared to the LSTM-Random Forest (0.9614) and standalone LSTM (0.9045). The hybrid model's steeper and more defined curves in the AUC-ROC plot indicate superior class separation and reduced misclassifications, particularly for overlapping sentiment categories. These results reflect the hybrid model’s ability to leverage GloVe embeddings for semantic richness and CNN-LSTM integration for feature extraction and sequence learning.

5.4. Practical Implications

The choice of a sentiment analysis model depends on the application’s specific requirements and constraints: CNN-GloVe-LSTM: Ideal for applications requiring high accuracy and nuanced sentiment classification with moderate computational resources. LSTM-Random Forest Ensemble: Suitable for scenarios requiring robust class separation but can tolerate higher computational costs. Standalone LSTM: Provides a baseline for fast training but struggles with accuracy and misclassification, especially for complex or ambiguous sentiments. The CNN-GloVe-LSTM model emerges as the most effective solution, balancing superior classification performance with computational efficiency. Its integration of complementary techniques - semantic embeddings, local feature extraction, and sequential processing - addresses the limitations of traditional LSTM architectures. While the LSTM-Random Forest Ensemble offers competitive performance, its higher computational demands make it less practical for resource-constrained scenarios. The standalone LSTM lacks the robustness and accuracy required for complex sentiment analysis tasks.

References

[1] Tyagi V, Kumar A, Das S. Sentiment Analysis on Twitter Data Using Deep Learning approach. 2020 2nd International Conference on Advances in Computing, Communication Control and Networking. 2020.

[2] Sosa PM. Twitter Sentiment Analysis using combined LSTM-CNN Models. UCSB. Available from: https://www.academia.edu/35947062/Twitter_Sentiment_Analysis_using_combined_LSTM_CNN_Models

[3] Rehman AU, Malik AK, Raza B, Ali W. A hybrid CNN-LSTM model for improving accuracy of movie reviews sentiment analysis. Multimedia Tools Appl. 2019;78(18):26597–613. https://doi.org/10.1007/s11042-019-07788-7

[4] Tam S, Said RB, Tanriover OO. A CONVBILSTM Deep Learning Model-Based Approach for Twitter Sentiment Classification. IEEE Access. 2021; 9:41283–93. https://doi.org/10.1109/access.2021.3064830

[5] Al-Selwi NSM, Hassan NMF, Abdulkadir NSJ, Muneer NA. LSTM inefficiency in Long-Term Dependencies Regression Problems. J Adv Res Appl Sci Eng Technol. 2023;30(3):16–31. https://doi.org/10.37934/araset.30.3.1631

[6] Yang B, Cardie C. Context-aware Learning for Sentence-level Sentiment Analysis with Posterior Regularization. Proc 52nd Annu Meet Assoc Comput Linguist. 2014;325–35. https://doi.org/10.3115/v1/p14-1031

[7] Wankhade M, Rao ACS, Kulkarni C. A survey on sentiment analysis methods, applications, and challenges. Artif Intell Rev. 2022;55(7):5731–80. https://doi.org/10.1007/s10462-022-10144-1

[8] Minaee S, Azimi E, Abdolrashidi A. Deep-Sentiment: sentiment analysis using ensemble of CNN and Bi-LSTM models. arXiv. 2019. https://doi.org/10.48550/arxiv.1904.04206

[9] Alagukumar S, Lawrance R. Impacts of various text preprocessing methods for topic modeling techniques. In: Futuristic Trends in Computing Technologies and Data Sciences. IIP Series; 2024. p. 162–70. https://doi.org/10.58532/v3bgct2p6ch3

[10] Rane N, Choudhary SP, Rane J. Ensemble deep learning and machine learning: applications, opportunities, challenges, and future directions. Stud Med Health Sci. 2024;1(2):18–41. https://doi.org/10.48185/smhs.v1i2.1225

[11] Chamidah N, Widiyanto D, Seta HB, Aziz A. The impact of oversampling and undersampling on Aspect-Based Sentiment Analysis of Indramayu tourism using logistic regression. Rev Intell Artif. 2024;38(3):795–804. https://doi.org/10.18280/ria.380306

[12] Negi M, Kaushik J, Dahiya A, Kaushik D. Sentiment Analysis with CNNs and LSTMs. SciOpen Preprints. 2022. https://doi.org/10.14293/s2199-1006.1.sor-.pp3potq.v1

[13] Dutschmann T, Kinzel L, Ter Laak A, Baumann K. Large-scale evaluation of k-fold cross-validation ensembles for uncertainty estimation. J Cheminform. 2023;15(1). https://doi.org/10.1186/s13321-023-00709-9

[14] Kang X, Han F, Fayjie A, Gong D. FocDepthFormer: Transformer with LSTM for Depth Estimation from Focus. arXiv. 2023. https://arxiv.org/abs/2310.11178

[15] Hegde SJ, Madhunandana HM, Mohana N. Sentiment analysis with LSTM Recurrent Neural Network Approach for movie reviews using deep learning. 2023 3rd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA). 2023. https://doi.org/10.1109/icimia60377.2023.10426266

[16] Nowak J, Taspinar A, Scherer R. LSTM Recurrent neural networks for short text and sentiment classification. Lect Notes Comput Sci. 2017;553–62. https://doi.org/10.1007/978-3-319-59060-8_50

[17] Zhou J, Huang JX, Chen Q, Hu QV, Wang T, He L. Deep Learning for Aspect-Level Sentiment Classification: Survey, vision, and challenges. IEEE Access. 2019; 7:78454–83. https://doi.org/10.1109/access.2019.2920075

[18] Nafea AA, Muayad MS, Majeed RR, Ali A, Bashaddadh OM, Khalaf MA, et al. A brief review on Preprocessing text in Arabic Language Dataset: Techniques and challenges. Deleted J. 2024;46–53. https://doi.org/10.58496/bjai/2024/007

[19] S A, Gnanasekaran P. Juncture of Text Preprocessing Techniques & Extracting Sentiment Analyzing of Micro-Blog Based on Machine Learning Algorithms. 2023 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES). 2023. https://doi.org/10.1109/icses60034.2023.10465450

[20] Barik K, Misra S, Ray AK, Bokolo A. LSTM-DGWO-Based Sentiment Analysis Framework for analyzing Online Customer Reviews. Comput Intell Neurosci. 2023; 2023:1–19. https://doi.org/10.1155/2023/6348831

[21] Weiss G, Goldberg Y, Yahav E. On the Practical Computational Power of Finite Precision RNNs for Language Recognition. arXiv. 2018. https://doi.org/10.48550/arxiv.1805.04908

[22] Tang D, Qin B, Feng X, Liu T. Effective LSTMs for Target-Dependent Sentiment Classification. arXiv. 2016. https://doi.org/10.48550/arxiv.1512.01100

[23] Ducci F, Kraus M, Feuerriegel S. Cascade-LSTM. KDD '20: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2020. https://doi.org/10.1145/3394486.3403317

[24] Zhu Q, Li X, Conesa A, Pereira C. GRAM-CNN: a deep learning approach with local context for named entity recognition in biomedical text. Bioinformatics. 2017;34(9):1547–55. https://doi.org/10.1093/bioinformatics/btx815

[25] Arsalan M. Transformers in Natural Language Processing: A Comprehensive review. Int J Res Appl Sci Eng Technol. 2024;12(5):5591–7. https://doi.org/10.22214/ijraset.2024.62863

[26] Capitaine L, Genuer R, Thiébaut R. Random forests for high-dimensional longitudinal data. Stat Methods Med Res. 2020;30(1):166–84. https://doi.org/10.1177/0962280220946080

[27] Naidu G, Zuva T, Sibanda EM. A review of evaluation metrics in Machine learning Algorithms. Lect Notes Netw Syst. 2023;15–25. https://doi.org/10.1007/978-3-031-35314-7_2

[28] Richardson E, Trevizani R, Greenbaum JA, Carter H, Nielsen M, Peters B. The receiver operating characteristic curve accurately assesses imbalanced datasets. Patterns. 2024;5(6):100994. https://doi.org/10.1016/j.patter.2024.100994