Journal of Information Systems and Informatics

Vol. 7, No. 2, June 2025 e-ISSN: **2656-4882** p-ISSN: **2656-5935**

DOI: 10.51519/journalisi.v7i2.1140 Published By DRPM-UBD

Comparison of RNN and LSTM Classifiers for Sentiment Analysis of Airline Tweets

Rogaia Yousif[1], Noon Fahmi[2], Sarmed Awad[3], Al-Baraa Ali[4]

[1,2,3,4]Computer Science, University of Gezira, Wad Madani, Sudan
Email: [1]rogiayusif@gmail.com, [2]noonfahmi41@gmail.com, [3]sarmedawad80@gmail.com, [4]mr.albra2@gmail.com

Abstract

This study examines the application of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) models for sentiment analysis of airline-related tweets, focusing on customer feedback directed at U.S. airlines on the X platform (formerly Twitter). The objective was to utilize these deep learning models to identify sentiment trends within text data and compare their performance in terms of computation time. The analysis was conducted on a 14,640-imbalanced dataset of classified tweets from February 2015 as positive, negative, or neutral. Both models were trained under identical conditions using Term Frequency-Inverse Document Frequency (TF-IDF) and Word2Vec for feature extraction. LSTM achieved 74% accuracy with AUC scores of 0.84, 0.90, and 0.89. RNN achieved 72% accuracy with AUC scores of 0.78, 0.87, and 0.85. In terms of time efficiency, RNN outperformed LSTM, requiring 57.16 seconds for training and 0.52 seconds for testing, compared to LSTM's 82.40 and 0.82 seconds. Time performance was also evaluated per sentiment class, and RNN consistently outperformed LSTM. These results highlight the trade-off between accuracy and computational cost. Limitations include dataset imbalance and LSTM's slower computation due to its internal gate mechanisms. Future work could prioritize integrating hybrid models and may use data imbalance techniques to improve sentiment classification.

Keywords: RNN, LSTM, Sentiment analysis, Time-consuming, Airlines tweets

1. INTRODUCTION

Sentiment analysis or opinion mining analyzes people's emotions, opinions, and attitudes as expressed in writing [1]. Nowadays, Individual users are increasingly interested in views about products and services available on the internet, which largely affects their choices. Beyond individual users, consumer sentiment analysis is crucial for companies to know how their products and services are perceived. Due to the massive amounts of data and viewpoints created, shared, and exchanged daily on the internet and various media [2], sentiment analysis has become essential for creating opinion mining systems. Sentiment analysis aims is to identify views expressed in a text and discern positive and negative opinions, which are crucial for the decisions and strategies of decision-makers [3].

Sentiment analysis recognition from text is an essential task in Natural Language Processing (NLP), which goes beyond standard word processor functions, which view text simply as a sequence of characters. NLP is one of the artificial intelligence subfields focused on the computational understanding, analysis, and extraction of meaning from human language intelligently and beneficially [4]. Developers can use NLP techniques to organize and structure their information for various tasks, including sentiment analysis, automatic text summarization, named entity recognition, speech recognition, translation, topic segmentation, and relationship extraction [5]. NLP takes into account the language's hierarchical nature, multiple words create a phrase, various phrases form a sentence, and ultimately, sentences express concepts. NLP is viewed as a challenging issue in the realm of computer science. The human language is seldom exact or clearly articulated. Understanding human language means understanding not only the words themselves, but also the concepts and connections that constitute meaning. While language is among the easiest subjects for humans to learn, its inherent ambiguity renders natural language processing a complex challenge for computers to conquer [6].

In the age of digital interaction, customer feedback on social media has become an essential information source. Analyzing these large volumes of textual data presents significant challenges due to their sequential nature and contextual complexity [7]. X is a social networking platform where users can share posts. On this site, people can express thoughts or feelings on various topics, fields, or themes [8]. It contains datasets of user opinions about airlines, which can be divided into negative, neutral, and positive. The amount of relevant information on X is greater than on other social media platforms, and the response time on X is significantly quicker. Marketers use sentiment analysis to gain insights about products and understand market trends [9].

Customer satisfaction assesses how consumers perceive products and services [10]. Many studies have shown that the quality and customer contentment are crucial factors in assessing the performance of the business. To maintain the organization's competitiveness, companies must carefully consider the needs and desires of their customers regarding the products or services they offer [11]. Furthermore, they should effectively manage their clients to ensure they are pleased to engage in business with them. Studies show that the airline industry has faced challenges in delivering exceptional services and fulfilling the needs of various consumer groups [12].

Deep learning has transformed natural language processing by allowing models to learn contextual relationships and complex patterns in text data [13]. This advancement has made it particularly effective for sentiment analysis tasks, especially when dealing with unstructured data from social media platforms [14]. This research offers a significant theoretical contribution by expanding the use of deep learning models, specifically RNN and LSTM. It focuses on utilizing artificial intelligence applications—particularly sentiment analysis and its related techniques—to analyze social media data from the X (Twitter) platform. The aim is to extract meaningful insights from unstructured text, enabling a deeper understanding of public opinion and communication trends across the airline industry.

In recent years, most studies in sentiment analysis field have examined several machine learning and deep learning techniques to enhance classification performance. Traditional machine learning models such as Support Vector Machine and Logistic Regression, when combined with Bag-of-Words and TF-IDF, achieved accuracy rates around 77% on real-world, imbalanced datasets like the Twitter US Airline Sentiment corpus [15]. However, these approaches were limited in capturing the contextual meaning of the text.

To address these limitations, researchers have turned to deep learning methods. RNN and LSTM models were applied across different datasets, achieving an accuracy of 88.47% in binary sentiment classification [9]. Similarly, another study applied LSTM and Gated Recurrent Unit (GRU) models to hotel reviews and reported high precision, recall, and F1-score (0.94, 0.98, and 0.96, respectively) for positive sentiments, though the models struggled with negative and neutral classes due to data imbalance [16]-[17]. Further advancements include the integration of LSTM into RNN networks to form an LSTM-RNN model, which was tested on SST, STT', and IMDB datasets. The model achieved accuracy rates of 52.5%, 84.3%, and 88.9%, respectively, and demonstrated good time efficiency, completing classification in 2264 seconds. However, other models like CNN and DRNN outperformed it in AUC metrics [18]-[19]. In another study focusing on Weibo comment texts, an LSTM-based model designed for three-class sentiment classification (positive, neutral, negative) reached 98.31% accuracy and a 98.28% F1-score, highlighting the model's strong capability in capturing contextual sentiment. However, it also faced challenges in terms of processing speed and computational cost [20].

These studies together show the growing effectiveness of deep learning in sentiment analysis, while also pointing to key limitations such as class imbalance, computational demands, and the need for efficient time performance. Despite the significant contributions in sentiment analysis utilizing deep learning techniques, particularly RNN and LSTM models, most previous studies have focused primarily on improving classification accuracy, with limited attention given to performance in terms of training and testing time. However, processing time is a critical factor, especially in time-sensitive applications or when working with large-scale data. While some studies have briefly addressed time performance using combined models such as LSTM-RNN, they did not conduct a separate comparison between RNN and LSTM as independent models [16].

Despite the increasing use of deep learning in sentiment analysis, there is a lack of direct comparative studies between RNN and LSTM models trained and evaluated under identical conditions, particularly in the context of airline-related tweets. Most previous research has either evaluated these models independently or focused on accuracy without addressing computational cost or time efficiency. This study addresses this gap by directly comparing both models on a real-world imbalanced dataset using consistent experimental settings.

This study utilizes a dataset of tweets concerning major US airlines collected in early 2015, offering structured sentiment labels that make it suitable for evaluating sentiment analysis models on real-world, short-text data. While the dataset reflects opinions from a past period, it remains a reliable benchmark for model testing and comparison. The paper contributes to the field by comparatively analyzing two widely adopted deep learning models—RNN and LSTM—under identical experimental conditions using TF-IDF and Word2Vec for feature extraction, emphasizing not only classification accuracy but also computational efficiency—a factor critical for time-sensitive sentiment analysis applications. This dual-focus approach addresses a gap in existing literature, where time performance is often overlooked. This gap analysis can build on this foundation by increasing dataset size, addressing data imbalance issues, and further refining the models for enhanced performance. This study aims to fill that gap by evaluating and comparing the performance of RNN and LSTM individually, considering not only accuracy but also training and testing time. By conducting experiments under the same conditions, this study highlights the trade-offs in computational efficiency between the two models in sentiment analysis tasks.

The primary objective of this study is to compare Long Short-Term Memory (LSTM) and Recurrent Neural Network (RNN) models in sentiment classification across positive, negative, and neutral classes based on customer reviews on X (formerly Twitter), with a particular focus on the feedback at U.S. airlines to analyze the computational time. Additionally, the study underscores the potential of these models to enhance strategic decision-making processes for airlines [21], providing a practical approach to develop data-driven strategies that boost service quality, customer satisfaction, and competitiveness in the airlines.

2. METHODS

Deep learning models like RNN and LSTM have shown promising capabilities in addressing these challenges by capturing temporal patterns and contextual dependencies in language. The paper aims to analyze customer reviews of popular

airlines on the X platform, following the principles used in natural language processing tasks. This includes all steps of preprocessing, training the model, and testing it on the collected tweets. The methodology is illustrated in
Figure **1**.

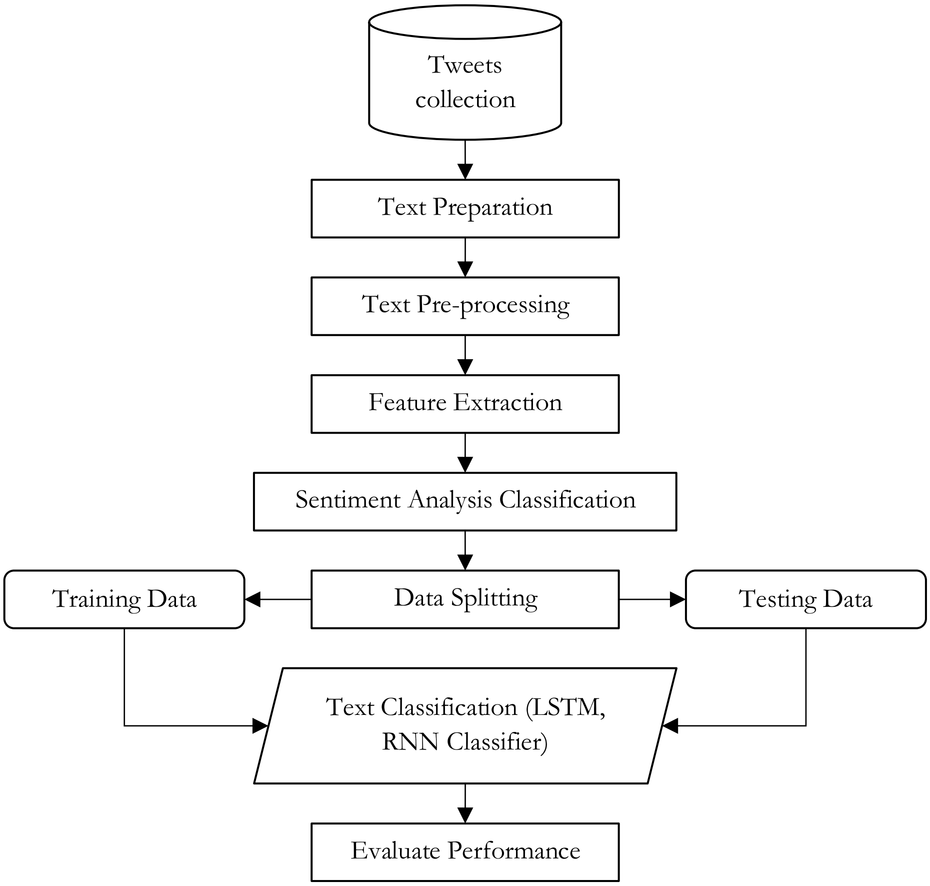

Figure 1. Overview of Study Methodology

Airline companies operate in a highly competitive environment where customer satisfaction directly influences brand reputation and business performance. With the social media platforms widespread use like X (Twitter), customers now openly express their opinions and experiences about airline services in real time. These tweets often include compliments, complaints, and suggestions, making them valuable sources of feedback for airlines.

2.1. Tweets collection

The collection of datasets is the foundation of any sentiment analysis research. This initial stage involves gathering tweets. X is a social media platform where users can share tweets, limited to 280 characters. For this analysis, we utilized a dataset obtained from Kaggle that comprises tweets related to airlines for sentiment analysis. This dataset contains 14,640 tweets collected in February 2015 and focuses on six U.S. airlines. Each tweet includes sentiment labels that indicate whether the sentiment is positive, negative, or neutral. However, the distribution of these sentiments is not balanced across the dataset, as shown in Figure 2, which illustrates the proportions of positive, neutral, and negative tweets. Nevertheless, the dataset remains a valuable resource for understanding and analyzing public sentiment toward airlines.

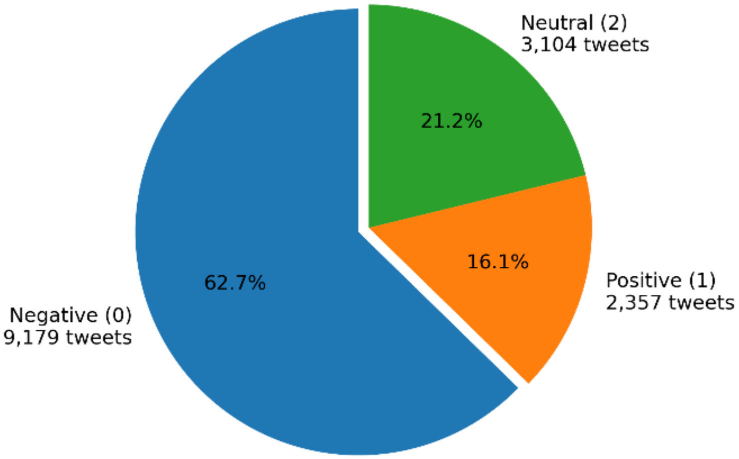

Figure 2. Sentiment Labels Distribution in the Dataset

2.2. Text Preparation

This process involves transforming raw data into an analysis-ready format, focusing on the text column, which contains the actual content of the tweets. This column is key for sentiment analysis, while the airline_sentiment column provides labels positive, negative, or neutral serving as the target variable for model training and evaluation. This approach ensures accurate sentiment classification.

Although the dataset exhibited class imbalance, with negative tweets forming the majority, no balancing techniques such as oversampling, undersampling, or class weighting were applied. Instead, the models were trained using the original data distribution. This decision aimed to evaluate both models under natural, real-world conditions, allowing for the assessment of their baseline performance when exposed to imbalanced data distributions.

2.3. Text Pre-processing

The preprocessing phase is essential in sentiment analysis to ensure clean, consistent, and suitable text data for deep learning models. Tweets can include various symbols (e.g., #, @), numbers, punctuation, and stop words—common words like "she," "the," "is," and "that" that do not carry any sentiment. Removing this noise is crucial for extracting valuable features.

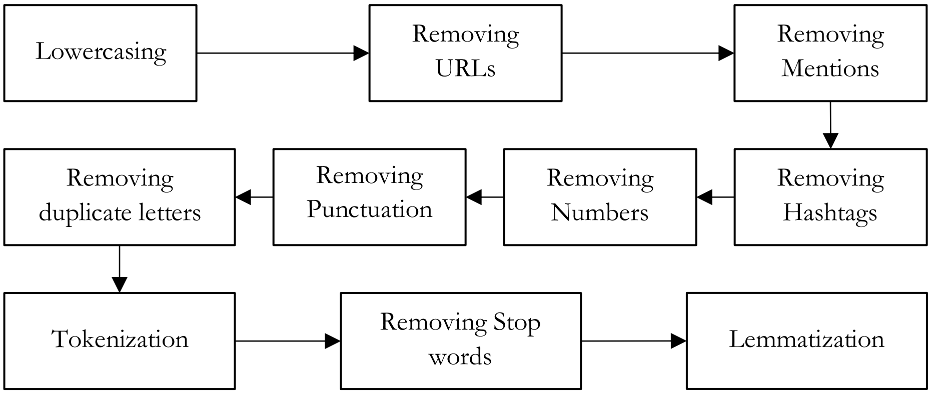

Figure 3. Workflow of Text Pre-processing Steps

The preprocessing steps that were used as follows:

1) Data Cleaning: This process begins with lowercasing the text to ensure uniformity. URLs are removed since they do not contribute to sentiment analysis (e.g., https://example.com). Mentions (e.g., @username) are also excluded, as they are often irrelevant to sentiment classification. Similarly, hashtags are usually discarded, along with numbers and punctuation marks. Redundant words, such as repeated characters (e.g., "sooooo good"), are standardized (e.g., "so good").
2) Tokenization: After cleaning, the text is tokenized; it is split into individual words or tokens. For example, the sentence "I love flying" is tokenized into ["I", "love", "flying"].
3) Stop Word Removal: Commonly used words, known as stop words are eliminated.
4) Lemmatization: In this step, words are reduced to their base form (e.g., "better" is transformed into "good"). Lemmatization helps maintain semantic accuracy.

To better understand the distribution of terms in the dataset, word clouds were generated before and after preprocessing. The initial word cloud (Figure 4) displayed a high frequency of common and repetitive words that lacked analytical value for the sentiment classification task. After preprocessing (Figure 5), the updated word cloud highlighted more relevant and meaningful terms, providing a clearer representation of the core content within the text data. This transformation demonstrates the effectiveness of the preprocessing steps in enhancing data quality for further analysis.

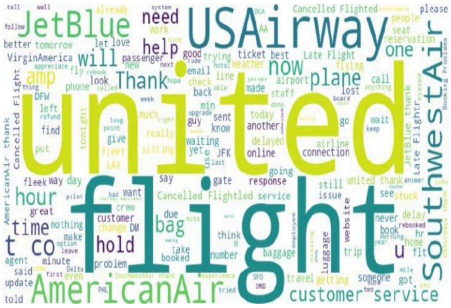

Figure 4. Before Pre-processing

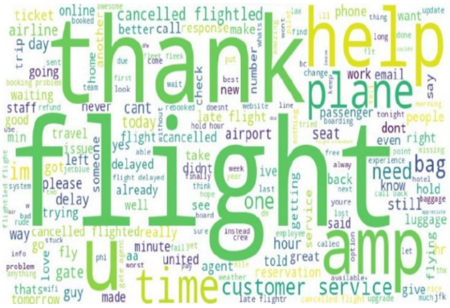

Figure 5. After Pre-processing

Additionally, Table 1 presents examples of actual tweets before and after cleaning. This demonstrates how the raw text is transformed into a structured and model-ready format.

Table 1. Examples of Raw Tweets and Their Cleaned Versions

No.	Actual Tweets	Tweets after clean
1.	@VirginAmerica What @dhepburn said.	said
2.	@VirginAmerica I didn't today… Must mean I need to take another trip!	didnt today must mean need take another trip
3.	@VlrginAmerica really aggressive to blast obnoxious "entertainment" in your guests' faces &amp; they have little recourse	really aggressive blast obnoxious entertainment guest face amp little recourse
4.	@VirginAmerica seriously would pay $30 a flight for seats that didn't have this playing.\n it's really the only bad thing about flying VA	seriously would pay flight seat didnt playing really bad thing flying va
5.	@VirginAmerica Really missed a prime opportunity for Men Without Hats parody, there. https://t.co/mWpG7grEZP	really missed prime opportunity men without hat parody

By following these preprocessing steps, the text data is transformed into a structured, consistent, and noise-free format suitable for effective feature extraction and sentiment classification.

2.4. Feature Extraction

In sentiment analysis, feature extraction refers to transforming raw textual data into numerical representations that could be used by deep learning models. Two commonly used methods for feature extraction are TF-IDF and Word2Vec.

2.4.1. Term Frequency-Inverse Document Frequency (TF-IDF)

A statistical metric that evaluates the significance of a term in a document concerning the entire dataset by combining term frequency (TF), which indicates how a term occurs frequently in a document, and inverse document frequency (IDF), which reflects the significance of common words that frequently appear in many documents while focusing on rare words [22]. First, the term frequency for each word in a document is calculated, followed by the computation of inverse document frequency, which assigns log-based weights to each term; the TF and IDF scores are then multiplied for every word, leading to a vector representation that forms a matrix where each row corresponds to a document and each column indicates a word's TF-IDF score.

2.4.2. Word2Vec

A machine learning technique that transforms words into dense vector representations (word embeddings) in a manner that reflects their semantic meaning; thus, words with comparable meanings have similar vector representations [23]. There are two main types of Word2Vec models: the Continuous Bag of Words (CBOW), which is designed to predict a word based on the context around it. Conversely, the model predicts context from a specific target word called a skip-gram. This methodology primarily employs the Skip-Gram model [24].

By using TF-IDF and Word2Vec together, the model can recognize semantically similar words (e.g., “good” and “great”) and differentiate between important and less important words in the context of a tweet. This combined approach is particularly beneficial for datasets with short texts, such as tweets, as it captures both local (document-specific) and global (corpus-wide) information. Moreover, padding is a crucial technique used in NLP to ensure that all sequences, such as sentences or documents, in a dataset have the same length. This uniformity is necessary because most deep learning models require inputs of consistent size to work efficiently.

2.5. Sentiment Analysis Classification

Each tweet in the dataset undergoes an evaluation of its polarity score. A sentence can either state a fact (objective) or express an opinion (subjective). By classifying sentences as either objective or subjective, we can eliminate objective statements, which may enhance the performance of the sentiment classification task, categorizing sentiments as positive, negative, or neutral. This process aligns with the broader goal of sentiment analysis, which quantifies the emotional tone within textual data. One of the methods used to calculate sentiment scores is the Absolute Proportional Difference. This approach considers the counts of positive (P), negative (N), and neutral (or other) (O) words within a text. The sentiment score is calculated by the following formula:

$$Sentiment\ Score=\frac{P\text{-}N}{P+N+O} \tag{1}$$

2.6. Data Splitting

The dataset was separated into two subsets using a standard data splitting technique to ensure reliable model evaluation. For training, 80% of the data was allocated, allowing the models to identify patterns within the data. The remaining 20% was used for testing the models to assess their generalization ability. During training, the performance was continuously monitored with validation data to track model behavior and avoid overfitting.

2.7. Text Classification

RNN and LSTM are popular text classification deep learning models. Both models are designed to handle sequential data by learning the order and context of words. In this study, both RNN and LSTM models are applied to classify text data and compare their performance and time efficiency.

2.7.1. Recurrent Neural Network (RNN)

RNNs are widely used for modeling sequential data in deep learning. This paper implements RNNs to explore their capacity and evaluate their performance. By applying preprocessing steps, the input data was appropriately prepared for the recurrent layers. The architecture of the RNN model consisted of two SimpleRNN layers, illustrated in Figure 6, each followed by dropout layers to reduce overfitting. These recurrent layers were connected to fully connected dense layers using ReLU activation, concluding with a softmax layer for multi-class classification. While RNNs are relatively easy to implement and train, but because of the vanishing gradient problem, they have limitations in learning long-term dependencies. This issue became apparent during model evaluation, where accuracy decreased on longer sequences and imbalanced classes.

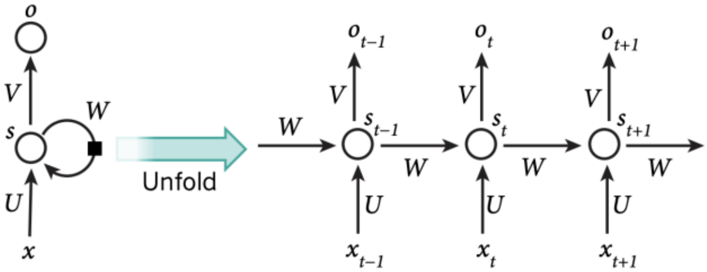

Figure 6. RNN Model Architecture [25]

2.7.2. Long Short-Term Memory (LSTM)

To address the shortcomings of standard RNNs, LSTM networks were adopted as a more advanced alternative. LSTMs are a specialized form of RNNs designed to retain information over long sequences using gated memory cells. These gates, which include input, forget, and output, solve the vanishing gradient problem and manage the information flow, making LSTMs suitable for datasets with long-range temporal dependencies.

The LSTM model architecture was enhanced by replacing SimpleRNN layers with LSTM layers, as illustrated in Figure 7, which were again followed by dropout layers and connected to dense classification layers.

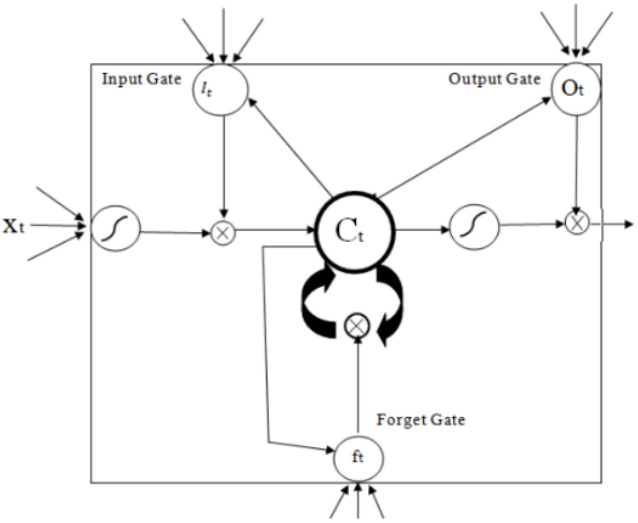

Figure 7. LSTM Model Architecture [26]

Both LSTM and RNN models were trained under identical settings. The training utilized categorical cross-entropy as the loss function and the Adam optimizer with a learning rate of 0.001. Each model was trained for 20 epochs with a batch size of 32. The architectures included two hidden layers with 128 and 64 units, respectively, interleaved with dropout layers (rate = 0.2) to reduce overfitting. A dense layer with 32 units and ReLU activation preceded the output layer. Feature extraction was performed using both TF-IDF and Word2Vec to enhance the input representation. Early stopping was applied during training to improve efficiency and prevent overfitting. These hyperparameters were selected based on initial experiments and practices commonly adopted in sentiment analysis studies.

2.8. Evaluate Performance

The final step of this study involves evaluating the trained RNN and LSTM classifiers on the test data allocated during the data splitting step. The models predict the sentiment for each tweet, and several performance metrics are computed to assess their effectiveness. One key technique used for evaluation and summarization is the confusion matrix. By considering the True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN), we can analyze how well the classification model performs. The model performance is analyzed through the confusion matrix to determine several metrics: accuracy, precision, recall, F1-score, and the area under the ROC curve (AUC). Additionally, evaluates the processing time required for data handling.

Accuracy measures the percentage of samples that have been classified correctly. Accuracy is calculated using Equation 2.

$$Accuracy=\frac{\Sigma_{i=1}^{N} TP_i}{\Sigma_{i=1}^{N} (TP_i+FP_i+FN_i)} \tag{2}$$

Precision shows the percentage of true positive predictions for each sentiment class relative to all positive predictions. Often referred to as positive predictive value, precision is calculated using Equation 3.

$$Precision_i=\frac{TP_i}{TP_i+FP_i} \tag{3}$$

Recall evaluates the model's ability to identify actual positive samples. Recall, also known as sensitivity, is calculated to show the completeness of a classifier using Equation 4.

$$Recall_i=\frac{TP_i}{TP_i+FN_i} \tag{4}$$

The F1 score provides a measure of the test's accuracy and is used to calculate both precision and recall as shown in Equation 5.

$$F1_i = 2 \times \frac{Precision_i \times Recall_i}{Precision_i + Recall_i} \quad (5)$$

The area under the ROC curve (AUC) is an important measure that represents the ability of the model to differentiate between sentiment classes; a higher AUC means greater distinguishing capability. This evaluation step offers a clear understanding of the effectiveness of the model. Furthermore, the processing time measured in this study refers to the duration from when the model is trained on the training data until it is evaluated and produces predictions. This approach ensures a comprehensive evaluation of the classifiers' performance.

3. RESULTS AND DISCUSSION

The experimental results of the RNN and LSTM models are presented in this section, especially emphasizing a comparison between training and testing times. In addition to time performance, the evaluation also includes confusion matrices, ROC curves, AUC scores, and class-wise evaluation metrics to highlight the differences between the two models. It is worth noting that all experiments for both models were conducted under identical conditions and experimental settings to ensure a fair and objective comparison. The same dataset, preprocessing steps, feature extraction methods, number of epochs, number of layers and units, programming libraries and versions, and a consistent T4 GPU computing environment were used. Therefore, the observed differences in computational time reflect the inherent characteristics of each model without the influence of any external factor.

3.1. Time Efficiency Comparison between RNN and LSTM Models

Since computational efficiency is a crucial factor in sentiment analysis models, this study intends to compare the training and testing times of RNN and LSTM models. The experimental results clearly demonstrate that the RNN model consistently outperforms the LSTM model regarding speed during the training and testing phases. Specifically, RNN required 57.16 seconds for training and 0.52 seconds for testing, while LSTM required 82.40 and 0.82 seconds. Table 2 includes an overview of the total training and testing times, as illustrated in Figure 8.

Table 2. Training and Testing Time Comparison

Model	Training Time (s)	Testing Time (s)
RNN	57.16	0.52
LSTM	82.40	0.82

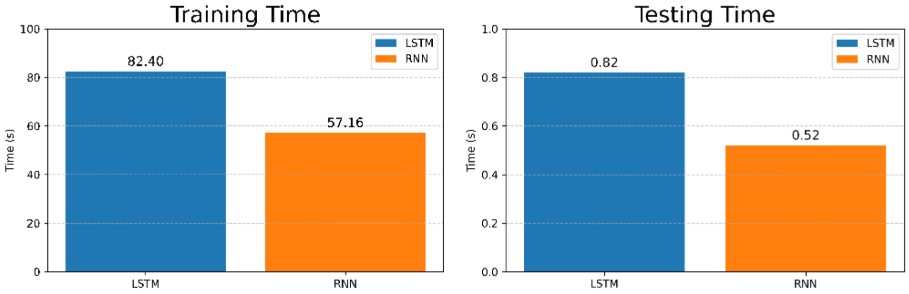

Figure 8. Training and Testing Time Comparison

Overall, RNN outperforms LSTM in terms of speed across all sentiment classes. For example, when evaluating the testing time for the positive class, the RNN required 0.1309 seconds, significantly faster than LSTM's 0.1489 seconds. This trend is evident across the negative and neutral classes as well. Detailed testing times per sentiment class are presented in Table 3 and also illustrated in Figure 9.

Table 3. Testing Time per Sentiment Class

Class	Testing Time(s)	
	RNN	**LSTM**
Positive	.1309	.1489
Negative	.1713	.1949
Neutral	.5084	.5785

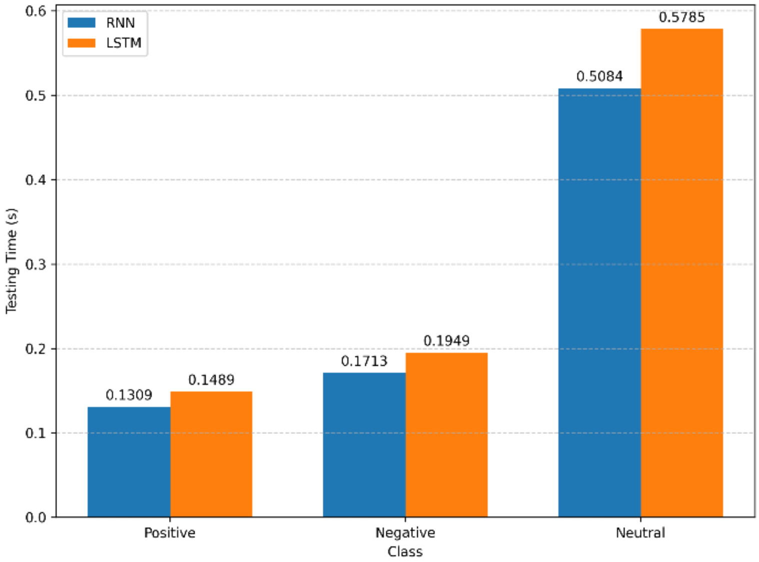

Figure 9. Testing Time per Sentiment Class

3.2. ROC Curve and AUC Analysis for RNN Model

The Receiver Operating Characteristic (ROC) Curve and Area Under the Curve (AUC) curve of the RNN model, as illustrated in Figure 10, illustrate the ability of the model to differentiate between sentiment classes. The model achieved an AUC of 0.78 for the negative class, 0.87 for the positive class, and 0.85 for the neutral class. Overall, the ROC-AUC curve reflects the strength of the model in handling the classification task, especially for the dominant classes. The macro-average AUC across the three classes was 0.83, indicating the RNN's overall discriminative capability.

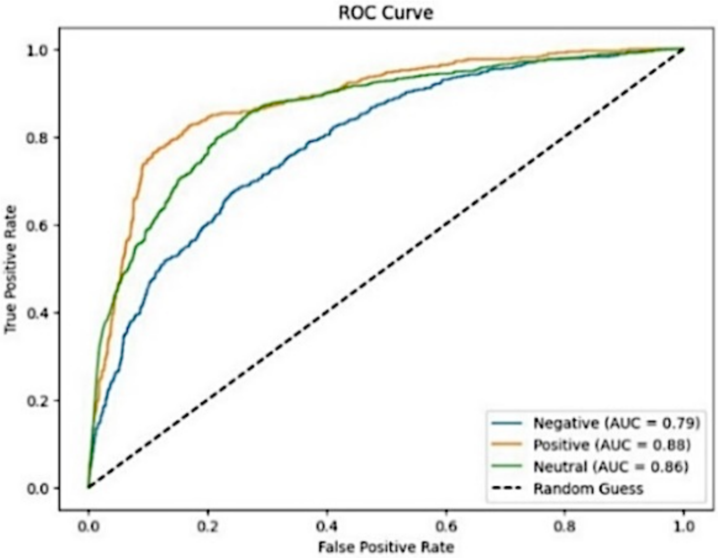

Figure 10. ROC Curve and AUC Scores for RNN Model

3.3. ROC Curve and AUC Analysis for LSTM Model

The Area Under the Curve (AUC) scores for the LSTM model demonstrate its ability to differentiate between the three sentiment classes effectively. The AUC for the negative class is 0.80. The positive class achieved the highest AUC of 0.88. Meanwhile, the neutral class has an AUC of 0.85. The macro-average AUC across these three classes is 0.84, highlighting the LSTM's general discriminative capability.

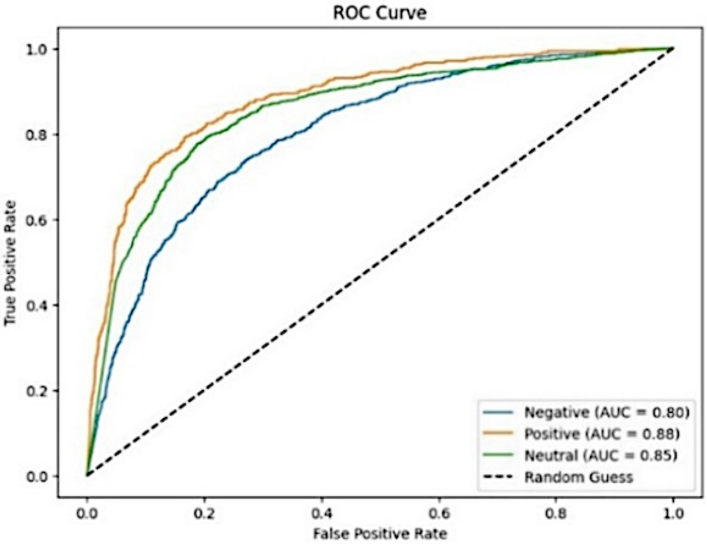

Figure 11. ROC Curve and AUC Scores for LSTM Model

3.4. Confusion Matrix Analysis

A comparison between the RNN and LSTM models shows that both models are able to classify sentiments with a reasonable level of accuracy. However, the LSTM model generally performs better than RNN, especially in identifying positive and neutral sentiments. Both models demonstrate lower performance on negative sentiment classification, as shown in the confusion matrices in Figure 12. Misclassifications are more prevalent in the negative class compared to the positive and neutral categories.

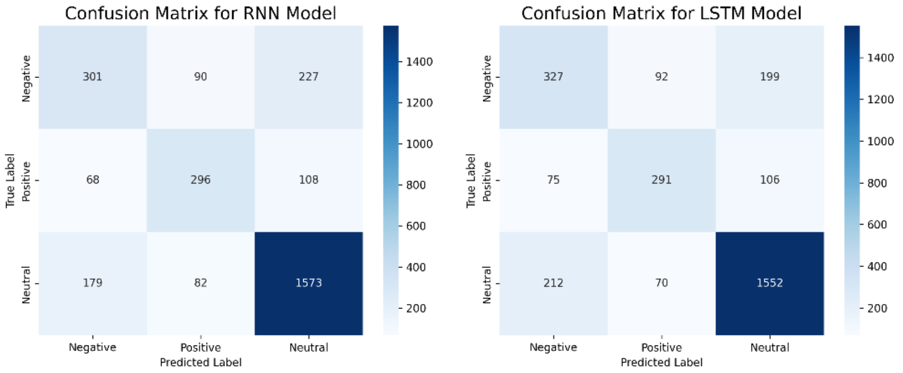

Figure 12. Confusion Matrices

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

3.5. Evaluation Metrics and Comparative Results

The performance of both RNN and LSTM models is evaluated through several metrics, including Precision, Recall, F1-score, Accuracy, and AUC. These metrics are reported for each sentiment class. This comparative assessment highlights the advantages and drawbacks of each model. Table 4 outlines the detailed results.

Table 4. Evaluation Metrics Across Sentiment Classes

Metric	Class	RNN	LSTM
Precision	Negative	.421	.520
	Positive	.649	.662
	Neutral	.829	.825
Recall	Negative	.478	.530
	Positive	.621	.597
	Neutral	.859	.845
F1-Score	Negative	.504	.525
	Positive	.635	.628
	Neutral	.844	.835
Accuracy		72%	74%
AUC		.83	.84

3.6. Discussion

The comparative analysis of time efficiency between Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) models underscores how architectural complexity directly influences processing performance. Due to its relatively simple design, the RNN model consistently demonstrates faster execution times across both training and testing phases. In contrast, the LSTM model, with its gated structure designed to better capture long-term dependencies, exhibits slower performance owing to its more computationally intensive operations. It's also essential to mention that individual training times for each sentiment class were not explicitly calculated. This is because the training process involves simultaneous updates to shared parameters across all classes, making class-wise time measurement impractical. However, the performance benefit of GPU acceleration was clearly evident for both models, significantly reducing runtime, with RNN retaining its edge in speed under both CPU and GPU environments.

When examining model effectiveness through Receiver Operating Characteristic (ROC) curves and Area Under the Curve (AUC) metrics, the RNN model exhibits a competent ability to distinguish between certain sentiment categories. Nonetheless, performance varies notably across different classes. This variability likely stems from imbalanced class distributions in the dataset, which hinder the model's generalization ability—especially for underrepresented sentiment classes. For example, sentiments such as "neutral" or "slightly negative," which appeared less frequently in the dataset, were not as reliably predicted by the RNN.

On the other hand, the LSTM model delivers a more consistent and robust performance across most sentiment categories. Its AUC scores reflect a high capability to differentiate between dominant sentiments such as "positive" and "negative." Still, the model faces challenges when dealing with sentiments that are inherently ambiguous or overlapping in context. This is particularly true for categories with fewer training instances or those expressing nuanced emotional tones. Despite its architectural advantages, the LSTM's effectiveness can still be compromised by the limitations of the training data.

The confusion matrix analysis further reinforces these findings. Both models struggle with accurately identifying negative sentiment instances, even though such instances are relatively common in the dataset. This difficulty likely arises from the semantic complexity and contextual variability inherent in negative expressions. Rather than being solely a result of class imbalance, the misclassification of negative sentiments seems rooted in the linguistic subtleties and overlaps between categories, which the models especially the RNN struggle to differentiate effectively. Additionally, the ability of both models to learn distinct contextual patterns associated with negative tones appears limited, which contributes to reduced classification precision in this class.

A more comprehensive evaluation using performance metrics such as precision, recall, F1-score, accuracy, and AUC reveals that both models are competent at sentiment classification, albeit with differing strengths. The RNN model's lightweight structure makes it a strong candidate for applications requiring rapid inference, while the LSTM model offers better accuracy and class-wise discrimination, making it suitable for use cases that prioritize classification quality over execution speed. The disparity in performance across sentiment classes also highlights the importance of choosing the appropriate model architecture depending on the specific needs and constraints of the deployment environment.

In practical terms, these findings hold considerable implications for real-world applications like airline customer feedback analysis. RNN's faster inference time makes it an ideal choice for scenarios where real-time processing is critical, such as live customer sentiment tracking. Meanwhile, the LSTM model's higher classification fidelity supports deeper sentiment interpretation and more nuanced insights. Both models, therefore, contribute meaningfully to the extraction of valuable sentiment data, which can guide strategic decision-making and enhance customer experience management.

4. CONCLUSION

This study compared LSTM and RNN models for sentiment classification using labeled tweets related to U.S. airlines collected in February 2015. Both models were trained under identical conditions using TF-IDF and Word2Vec for feature extraction. The results explicate that RNN has faster training and testing times due to its simpler architecture, which enables less computation time. In contrast, LSTM achieved higher accuracy and AUC scores across positive, negative, and neutral sentiment classes. These results demonstrate a trade-off between accuracy and computation time, which is relevant when considering large-scale or time-sensitive applications. The results also highlighted that both models struggled to accurately classify negative sentiments, despite their higher frequency in the dataset. This suggests that further efforts are needed to address the semantic overlap between negative and neutral expressions, as well as the effects of class imbalance. In future studies, based on the results of the sentiment analysis presented in this paper, researchers should focus on enhancing the adoption of deep learning models for analyzing tweets from airlines. They could prioritize integrating hybrid models that leverage the strengths of different architectures to improve overall performance, or they could explore using a bidirectional Long Short-Term Memory (biLSTM) model. Additionally, addressing data imbalance through balancing techniques such as oversampling or undersampling may enhance sentiment classification. It is also important to ensure that the dataset is balanced across the different sentiment classes.

REFERENCES

[1] R. Alroobaea, "Sentiment analysis on amazon product reviews using the recurrent neural network (rnn)," *International Journal of Advanced Computer Science and Applications,* vol. 13, no. 4, 2022.

[2] M. Wankhade, A. C. S. Rao, and C. Kulkarni, "A survey on sentiment analysis methods, applications, and challenges," *Artificial Intelligence Review,* vol. 55, no. 7, pp. 5731-5780, 2022.

[3] P. Cen, K. Zhang, and D. Zheng, "Sentiment analysis using deep learning approach," *J. Artif. Intell,* vol. 2, no. 1, pp. 17-27, 2020.

[4] P. M. Mah, I. Skalna, and J. Muzam, "Natural language processing and artificial intelligence for enterprise management in the era of industry 4.0," *Applied Sciences,* vol. 12, no. 18, pp. 9207, 2022.

[5] S. B. B. Priyadarshini, A. B. Bagjadab, and B. K. Mishra, "A brief overview of natural language processing and artificial intelligence," *Natural language processing in artificial intelligence,* pp. 211-224, 2020.

[6] B. S. Anupama, D. B. Rakshith, K. M. Rahul, and M. Navaneeth, "Real time Twitter sentiment analysis using natural language processing," *Int J Eng Res,* vol. 9, pp. 1107-1112, 2020.

[7] J. S. Datt, K. Bhasin, and P. Jamwal, "SENTIMENT ANALYSIS USING CUSTOMER FEEDBACK," *International Journal of Trendy Research in Engineering and Technology,* vol. 7, no. 4, 2023.

[8] O. Adwan, M. Al-Tawil, A. Huneiti, R. Shahin, A. A. Zayed, and R. Al-Dibsi, "Twitter sentiment analysis approaches: A survey," *International Journal of Emerging Technologies in Learning (iJET),* vol. 15, no. 15, pp. 79-93, 2020.

[9] P. C. Shilpa, R. Shereen, S. Jacob, and P. Vinod, "Sentiment analysis using deep learning," in *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, 2021: IEEE, pp. 930-937.

[10] A. Alqurafi and T. Alsanoosy, "Measuring Customers' Satisfaction Using Sentiment Analysis: Model and Tool," *J. Comput. Sci,* vol. 20, pp. 419-430, 2024.

[11] S. Banerjee, A. Pandit, T. O. Olubiyi, and R. P. Kumar, "Sentiment Analysis in Customer Relationship Management," in *Demystifying Emotion AI, Robotics AI, and Sentiment Analysis in Customer Relationship Management*: IGI Global Scientific Publishing, 2025, pp. 161-178.

[12] S. L. Idris and M. Mohamad, "A Study on Sentiment Analysis on Airline Quality Services: A Conceptual Paper," *Information Management and Business Review,* vol. 15, no. 4, pp. 564-576, 2023.

[13] N. S. Suryawanshi, "Sentiment analysis with machine learning and deep learning: A survey of techniques and applications," in *International Journal of Science and Research Archive*, vol. 12: GSC Online Press, 2024, pp. 005-015.

[14] N. C. Dang, M. N. Moreno-García, and F. De la Prieta, "Sentiment analysis based on deep learning: A comparative study," *Electronics,* vol. 9, no. 3, pp. 483, 2020.

[15] M. T. H. K. Tusar and M. T. Islam, "A comparative study of sentiment analysis using NLP and different machine learning techniques on US airline Twitter data," in *2021 International Conference on Electronics, Communications and Information Technology (ICECIT)*, 2021: IEEE, pp. 1-4.

[16] Y. A. Singgalen, "Sentiment Analysis and Trend Mapping of Hotel Reviews Using LSTM and GRU," *Journal of Information Systems and Informatics,* vol. 6, no. 4, pp. 2814-2836, 2024.

[17] K. L. Tan, C. P. Lee, K. S. M. Anbananthen, and K. M. Lim, "RoBERTa-LSTM: a hybrid model for sentiment analysis with transformer and recurrent neural network," *IEEE Access,* vol. 10, pp. 21517-21525, 2022.

[18] D. Wang, "Sentiment Analysis of English Text based on LSTM-RNN," *International Journal of Educational Innovation and Science,* 2022.

[19] M. Beseiso, "Word and character information aware neural model for emotional analysis," *Recent Patents on Computer Science,* vol. 12, no. 2, pp. 142-147, 2019.

[20] Y. Qixuan, "Three-Class Text Sentiment Analysis Based on LSTM," in *arXiv e-prints*, ed, 2024, p. arXiv:2412.17347.

[21] Z. Jin, Y. Yang, and Y. Liu, "Stock closing price prediction based on sentiment analysis and LSTM," *Neural Computing and Applications,* vol. 32, pp. 9713-9729, 2020.

[22] H. H. Hussein and A. Lakizadeh, "A Systematic Assessment of Sentiment Analysis Models on Iraqi Dialect-based Texts," *Systems and Soft Computing,* pp. 200203, 2025.

[23] L. Damayanti and K. M. Lhaksmana, "Sentiment analysis of the 2024 Indonesia presidential election on twitter," *Sinkron: jurnal dan penelitian teknik informatika,* vol. 8, no. 2, pp. 938-946, 2024.

[24] M. Idris, A. Rifai, and K. D. Tania, "Sentiment Analysis of Tokopedia App Reviews using Machine Learning and Word Embeddings," *Sinkron: jurnal dan penelitian teknik informatika,* vol. 9, no. 1, pp. 210-219, 2025.

[25] P. C. Huy, N. Q. Minh, N. D. Tien, and T. T. Q. Anh, "Short-term electricity load forecasting based on temporal fusion transformer model," *Ieee Access,* vol. 10, pp. 106296-106304, 2022.

[26] P. Pan and Y. Chen, "Automatic subject classification of public messages in e-government affairs," *Data and Information Management,* vol. 5, no. 3, pp. 336-347, 2021.