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

(RESEARCH ARTICLE)

Check for updates

LSTM-based deep learning framework for sentiment classification of Flipkart product Reviews

Pilli Lalith Sriharsha * and Suneel Kumar Duvvuri

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

Global Journal of Engineering and Technology Advances, 2026, 27(01), 158-177

Publication history: Received on 17 March 2026; revised on 23 April 2026; accepted on 25 April 2026

Article DOI: https://doi.org/10.30574/gjeta.2026.27.1.0099

Abstract

The primary objective of this study is to develop a deep learning-based model for multi-class sentiment analysis of Flipkart product reviews, categorizing them into positive, negative, and neutral classes. The dataset comprises 21,904 product reviews, and data preprocessing was performed using Pandas and NumPy, including handling missing values and duplicate removal. Text preprocessing techniques such as tokenization, stop word removal, and stemming were applied using NLTK. Exploratory Data Analysis (EDA) was conducted using Matplotlib and Seaborn, while Word Cloud visualization was used to identify frequently occurring terms. The textual data was transformed into numerical sequences using the TensorFlow Keras Tokenizer and standardized through sequence padding. A deep learning model based on Long Short-Term Memory (LSTM) was implemented using TensorFlow/Keras, incorporating an Embedding layer, LSTM layer, Dropout layer, and Dense output layer with Softmax activation. The model was trained using the Adam optimizer with Early Stopping for optimization. The proposed model achieved a high-test accuracy of 96.94%, and evaluation using Scikit-learn metrics demonstrated strong performance with high precision, recall, and F1-score across all sentiment classes. The confusion matrix further confirms the model's effectiveness in accurately classifying positive, negative, and neutral reviews. The study demonstrates that LSTM-based deep learning models, when combined with effective preprocessing and feature engineering techniques, can achieve highly accurate multi-class sentiment classification, proving robust and reliable for analyzing real-world product review data.

Keywords: Sentiment Analysis; LSTM; TensorFlow/Keras; NLTK; Pandas; NumPy; Matplotlib; Seaborn; Word Cloud; Scikit-learn; Flipkart Reviews; Deep Learning

1. Introduction

1.1. Background of the Study

The rapid growth of digital technologies and internet usage has transformed e-commerce, allowing customers to easily explore and purchase products online. A key feature of these platforms is user-generated product reviews, which provide valuable insights into customer experiences, opinions, and satisfaction. Unlike traditional advertisements, these reviews reflect genuine user feedback and offer reliable information on product quality, usability, pricing, and service efficiency. However, the exponential growth in review volume makes manual analysis impractical, creating a need for automated sentiment analysis techniques [1] [16].

1.2. Overview of Sentiment Analysis

Sentiment analysis, also known as opinion mining, is a Natural Language Processing technique used to identify and classify opinions expressed in textual data into categories such as positive, negative, and neutral. In this study, sentiment analysis is applied to Flipkart product reviews to automatically determine customer opinions and extract

* Corresponding author: Pilli Lalith Sriharsha.

meaningful insights. From a computational perspective, sentiment analysis is treated as a classification problem where contextual relationships and linguistic structures play a crucial role. The transformation of raw textual input into numerical representations through tokenization and encoding enables models to learn patterns effectively. Deep learning models such as LSTM further enhance this process by capturing sequential dependencies and improving contextual understanding[3] [4].

1.3. Importance of Sentiment Analysis in E-Commerce

E-commerce platforms generate large volumes of textual customer reviews that offer valuable insights into user experiences and product performance. Manual analysis is inefficient, making automated sentiment analysis essential for extracting meaningful information[18] [32].

Table 1 Sentiment Classification Categories

Sentiment	Definition	Review
Positive	Expresses satisfaction or favourable opinion	"Loved the product quality!"
Negative	Expresses dissatisfaction or unfavourable opinion	"The item was broken on arrival."
Neutral	Neither strongly positive nor negative	"It works as expected."

The table 1 neatly classifies customer reviews into three sentiment types-positive, negative, and neutral. Each category is defined with a clear example, making sentiment analysis straightforward and easy to interpret.

1.4. Challenges in Sentiment Analysis

Sentiment analysis is challenging due to linguistic variability, informal expressions, and contextual dependencies in user-generated content. Additional issues such as noisy data, imbalanced classes, mixed sentiments, and implicit expressions further complicate accurate classification, requiring advanced models to capture deeper context[22].

Table 2 Common Challenges in Sentiment Analysis

Challenge	Description	Example
Noisy Data	Informal text, abbreviations, errors	"Gr8 product!!"
Context Sensitivity	Meaning depends on context	"Not bad"
Imbalanced Classes	Unequal sentiment distribution	More positive reviews
Mixed Sentiments	Multiple opinions in one review	"Good but late delivery"

The table 2 outlines key challenges in sentiment analysis, such as noisy data, context sensitivity, and class imbalance. It illustrates each issue with practical examples from the dataset, making the problems clear and relatable.

1.5. Evolution of Sentiment Analysis Techniques

Sentiment analysis has progressed from rule-based methods, which relied on predefined lexicons and lacked contextual understanding, to machine learning approaches that learned patterns but required manual feature engineering. Deep learning models, particularly LSTM networks, further improved performance by effectively capturing sequential dependencies and contextual relationships in text[20].

Table 3 Comparative Overview of Sentiment Analysis Approaches, Their Key Features, and Advantages

Approach	Features	Advantages
Rule-Based	Lexicons, keyword matching	Simple, interpretable
Machine Learning	Naïve Bayes, SVM	Learns from data
Deep Learning	LSTM, RNN	Captures sequence & context
Approach	Features	Advantages

Table 3 compares sentiment analysis approaches. It highlights how rule-based methods rely on lexicons for simplicity, machine learning models learn patterns from data, and deep learning techniques capture sequential context for richer understanding.

1.6. Significance of LSTM in Text Classification

The complexity of textual data requires models capable of capturing sequential dependencies and contextual relationships, which traditional machine learning approaches often fail to handle effectively as they treat text as independent features. Long Short-Term Memory (LSTM) networks address this limitation by incorporating memory cells and gating mechanisms that enable the retention and selective updating of information over time. This capability allows LSTM models to effectively learn both short-term and long-term dependencies in textual data, making them highly suitable for sentiment classification tasks.

1.7. Motivation of the Study

The rapid growth of e-commerce platforms has resulted in large volumes of unstructured textual data, necessitating automated systems for sentiment analysis. Manual analysis is inefficient and inconsistent, motivating the development of scalable deep learning-based solutions. This study aims to improve classification accuracy and address challenges related to contextual understanding and sequential data processing using LSTM models.

1.8. Problem Statement

The increasing volume of customer review data presents challenges in extracting meaningful insights due to unstructured text, contextual ambiguity, mixed sentiments. Traditional methods fail to capture sequential dependencies, leading to reduced classification performance. Therefore, there is a need for an advanced model capable of effectively processing sequential data and improving sentiment classification accuracy.

1.9. Objectives of the Study

The primary objective of this study is to develop an effective framework for analyzing customer review data and classifying sentiments accurately. The study focuses on leveraging advanced computational techniques to process unstructured textual information and extract meaningful insights that can support data-driven decision-making.

- To design a structured pipeline for sentiment analysis
- To convert textual data into numerical representations
- To implement an LSTM-based classification model
- To evaluate model performance using standard metrics
- To improve accuracy and robustness

1.10. Scope of the Study

This study focuses on sentiment classification of Flipkart reviews using textual data. It includes preprocessing, feature extraction, model development, and evaluation within a controlled experimental setup. The scope is limited to text-based analysis and does not include multimodal data or real-time deployment.

1.11. Organization of the Thesis

This research work is organized into five chapters covering introduction, literature review, methodology, results, and conclusion. Each chapter focuses on a specific aspect of the study, ensuring a structured and logical flow of information.

2. Literature Review

2.1. Introduction to Literature Review

Sentiment analysis has become important in NLP due to the growth of user-generated text, especially in e-commerce where reviews reflect customer experiences and perceptions, requiring advanced methods to handle linguistic variability and context. Over time, approaches evolved from traditional models with limited contextual understanding to deep learning techniques, and this chapter reviews existing methods, datasets, performance, and limitations while identifying research gaps supporting the use of LSTM-based models.

2.2. Early Machine Learning Approaches

Early sentiment analysis relied on traditional machine learning methods like Naïve Bayes and SVM, which treat text as independent features and use statistical patterns for classification. Studies such as Pang et al. (2002) and Turney (2002) achieved moderate accuracy but were limited in handling context, ambiguity, and complex linguistic structures[7] [17]

2.3. Representation Learning and Feature Engineering

With advancements in sentiment analysis, research shifted toward improving feature representation to enhance classification performance. These approaches focused on transforming textual data into numerical representations while preserving semantic relationships between words. Maas et al. (2011) introduced distributed representations that improved sentiment classification accuracy to around 88%. Mikolov et al. (2013) further advanced this area through Word2Vec, enabling better semantic understanding. However, these embeddings are static and fail to capture context-dependent meanings, limiting their effectiveness in complex sentence structures[5] [11].

2.4. Deep Learning-Based Approaches

Deep learning improved sentiment analysis by enabling automatic feature extraction and capturing complex textual patterns, with CNNs performing well in sentence classification. While RNNs introduced sequential processing, LSTM models overcame their limitations using memory and gating mechanisms, allowing better handling of long-term dependencies and outperforming traditional approaches[9] [35].

2.5. Transformer-Based Advancements

Transformer-based models use attention mechanisms to capture deep contextual relationships and process text bidirectionally, improving understanding of complex sentences. Models like BERT achieve high accuracy, but their high computational cost limits practical use compared to LSTM-based systems[13] [27].

2.6. Optimization and Training Techniques

Optimization techniques improve model performance by minimizing loss and ensuring stable convergence during training. The Adam optimizer uses adaptive learning rates and combines momentum with gradient-based methods, enabling faster and more efficient training[15] [10]

2.7. Recent E-Commerce Sentiment Studies

Recent studies apply deep learning to e-commerce sentiment analysis, achieving high accuracy on large-scale review datasets. However, challenges such as data imbalance and noise remain, highlighting the need for more efficient and scalable models[24] [33].

2.8. Author-wise Comparison of Existing Studies

Table 4 Comparative Analysis Table

Author / Reference	Year	Method Used	Dataset	Accuracy	Key Contribution	Limitation
Bo Pang et al [2]	2002	Naive Bayes, SVM	Movie Reviews	82.9%	Introduced ML for sentiment classification	No context understanding
Peter Turney [19]	2002	PMI (Unsupervised)	Product Reviews	74%	No labelled data required	Poor handling of complex sentences
Andrew L. Maas et al [21]	2011	Word Embeddings	IMDB Reviews	88%	Improved feature representation	Static embeddings
Tomas Mikolov et al [8]	2013	Word2Vec	Large Corpus	Not reported	Semantic word vectors	No contextual awareness
Yoon Kim [12]	2014	CNN	Sentence Classification	87–90%	Automatic feature extraction	Poor long-term dependency

Ilya Sutskever et al [26]	2014	RNN	Sequential Text	80–85%	Sequence modelling	Vanishing gradient problem
Sepp Hochreiter & Jürgen Schmidhuber [14]	1997	LSTM	Sequential Data	Not reported	Handles long-term dependencies	High computation
Duyu Tang et al [28]	2015	LSTM	Review Data	85–88%	Better context understanding	Sensitive to noise
Jacob Devlin et al [23]	2019	BERT	Large Text Corpus	92–94%	Bidirectional context understanding	High computational cost
Chi Sun et al [30]	2019	Fine-tuned BERT	Benchmark Datasets	94–96%	Improved performance via fine-tuning	Requires heavy resources
Diederik P. Kingma & Jimmy Ba [31]	2015	Adam Optimizer	Not reported	Not reported	Efficient training of DL models	Sensitive to parameters
Zhang et al [6]	2018	DL Survey	E-commerce Reviews	~90%	Overview of DL methods	Data imbalance issue
Proposed Work	2026	LSTM-Based Deep Learning Framework	Flipkart Reviews Dataset	~96%	Enhances sentiment classification using LSTM	Computational complexity

Table 4 presents an author-wise comparison of sentiment analysis techniques, including methodologies, datasets, accuracy, and limitations[34][29] .

2.9. Comparative Analysis (Aligned with Your Pipeline)

The comparative analysis of existing approaches shows that traditional machine learning models are simple and efficient but fail to capture contextual and sequential relationships. Representation learning improves semantic understanding but lacks contextual awareness in dynamic text environments. Deep learning models such as CNN and RNN enhance feature extraction but have limitations in capturing long-term dependencies. LSTM models overcome these issues by effectively modelling sequential data, while transformer models provide high accuracy at the cost of computational complexity. Therefore, LSTM offers a balanced solution for sentiment classification in this study[25].

2.10. Research Gaps

Based on the literature review, several research gaps are identified in existing sentiment analysis approaches. Traditional models fail to capture contextual and sequential relationships, while representation learning techniques lack dynamic context understanding. Deep learning models improve performance but still face challenges such as data noise, imbalance, and computational complexity. Additionally, transformer-based models require high computational resources, limiting their practical applicability. These limitations highlight the need for an efficient model that balances performance and complexity, motivating the use of LSTM for sentiment classification in this study.

2.11. Summary of the Chapter

This chapter presented a comprehensive review of sentiment analysis techniques, covering traditional machine learning methods, representation learning approaches, deep learning models, and transformer-based architectures. The analysis highlighted their strengths, limitations, and applicability in real-world scenarios. The comparative evaluation and identification of research gaps provide a strong foundation for the proposed methodology. The findings justify the selection of an LSTM-based approach as an effective solution for handling sequential dependencies and improving sentiment classification performance.

3. Methodology

3.1. Introduction

This study adopts a structured methodology to transform raw textual review data into a format suitable for multi-class sentiment classification using deep learning techniques. Customer reviews are inherently unstructured and exhibit variations in language, writing style, and expression patterns, making direct analysis challenging. Therefore, a systematic pipeline is required to ensure consistency, improve data quality, and enable accurate sentiment prediction.

The proposed framework integrates dataset inspection, preprocessing, feature engineering, exploratory analysis, text representation, sequence preparation, and LSTM-based modelling. Each stage contributes to refining input data while preserving contextual relationships, enabling the model to effectively learn patterns and perform accurate sentiment classification [36][37][38][39].

3.2. System Workflow and Architecture

The system is designed as a sequential workflow integrating data processing and model development. Each stage transforms raw textual input into structured numerical representations suitable for deep learning models.

Algorithm: LSTM-Based Sentiment Classification Framework
Input: Review dataset with sentiment labels
Output: Predicted sentiment class
Step 1- Load and inspect the dataset.
Step 2 - Handle missing values and analyse duplicate entries.
Step 3 - Perform text cleaning and normalization.
Step 4 - Extract structural text features.
Step 5 - Conduct exploratory data analysis for pattern identification.
Step 6 - Generate sentiment-based word representations.
Step 7 - Encode sentiment labels numerically.
Step 8 - Convert text into sequences using tokenization.
Step 9 - Apply sequence padding for uniform input length.
Step 10 - Construct the LSTM model architecture.
Step 11 - Train the model with optimization and regularization.
Step 12 - Evaluate performance using standard metrics and confusion matrix.
Step 13 - Deploy the model for prediction on unseen data.

3.3. Dataset Description

The dataset contains 21,904 Flipkart product reviews with attributes such as product name, review text, summary, and sentiment label. The review text serves as the primary input feature, while the sentiment label acts as the target variable for classification. Initial inspection confirms that the dataset is largely complete, with minimal missing values. This ensures reliability and suitability for further preprocessing and modelling tasks.

Table 5 Dataset Overview

Attribute	Description
Total Record	21,904
Input Feature	Review Text
Target Feature	Sentiment
Classes	Positive, Negative, Neutral

Table 5 presents the dataset attributes, showing a total of 21,904 records with review text as the input feature and sentiment as the target. It categorizes sentiments into three classes-positive, negative, and neutral-providing the foundation for model training and evaluation.

3.4. Data Preprocessing

Data preprocessing ensures that the dataset is clean and consistent. Missing values, duplicate entries, and irregular text patterns are addressed to improve model performance. The preprocessing pipeline standardizes text and removes noise, ensuring that the dataset is suitable for feature extraction and sequence modelling.

Mathematical Representation of Text Cleaning

Let raw text be T, then cleaned text T' is:

$$T' = f(T) \quad 1$$

where f represents normalization, tokenization, and filtering operations.

3.4.1. Handling Missing Values

Missing values are handled by replacing null entries with empty strings instead of removing records, ensuring dataset completeness. This approach preserves information and prevents disruptions in subsequent preprocessing steps.

3.4.2. Duplicate Review Analysis

Duplicate entries are identified to understand repetition patterns in the dataset. Instead of removing them, their presence is analysed to maintain dataset characteristics. This step ensures better understanding of data distribution and prevents unintended bias during model training.

3.4.3. Text Cleaning and Normalization

Text normalization reduces inconsistencies in casing, punctuation, and formatting, improving data quality. The process ensures uniform representation of textual data while preserving semantic meaning.

3.5. Feature Engineering and Exploratory Data Analysis

Feature engineering extracts structural features such as word count, character count, and sentence count to analyse textual patterns. EDA reveals that the dataset has a balanced sentiment distribution, supporting unbiased model training.

Table 6 Text Feature Representation

Feature	Description
Character Count	Length of review
Word Count	Number of tokens
Sentence Count	Number of sentences

Table 6 outlines the structural features of the dataset, including character count, word count, and sentence count. These attributes provide insights into review length and complexity, supporting deeper text-based sentiment analysis.

3.6. Frequent Word Analysis

Frequent word analysis identifies commonly occurring terms within each sentiment class, helping in understanding user expression patterns.

This analysis enhances interpretability and supports feature extraction for sentiment classification.

3.7. Text Representation and Sequence Preparation

Text data is transformed into numerical sequences for deep learning models. Tokenization maps words to numerical indices, while padding ensures uniform sequence length.

Tokenization Formula

$$S = \{w_1, w_2, \ldots, w_n\} \rightarrow \{x_1, x_2, \ldots, x_n\} \quad 2$$

where $w_i$are words and $x_i$are numerical indices.

Padding Formula

$$X' = \text{pad}(X, L) \quad 3$$

where $L$is fixed sequence length.

3.8. Proposed LSTM Model

The Long Short-Term Memory (LSTM) model, a type of Recurrent Neural Network (RNN), is designed to capture sequential dependencies in textual data. It effectively learns long-term contextual relationships using memory cells and gating mechanisms, making it suitable for sentiment classification tasks.

The proposed model includes an Embedding layer, LSTM layer, Dropout layer, and Dense output layer with Softmax activation. The Embedding layer converts text into vector form, the LSTM captures contextual dependencies, Dropout reduces overfitting, and the Dense layer performs final sentiment classification.

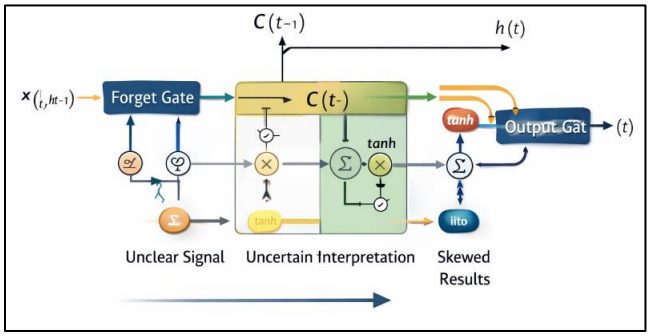

Figure 1 LSTM cell architecture illustrating input, forget, and output gate interactions

The fig 1 depicts the structure of an LSTM cell, showing how information flows through forget, input, and output gates. It highlights how sequential dependencies are managed to retain context in deep learning models.

The LSTM model captures sequential dependencies in textual data, enabling effective sentiment classification.

LSTM Mathematical Representation

$$h_t = \sigma(W_h x_t + U_h h_{t-1} + b_h) \quad 4$$

where $h_t$is hidden state, $x_t$is input, and $\sigma$is activation function.

3.9. Performance Metrics Interpretation

Model performance is evaluated using accuracy, precision, recall, and F1-score.

Accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad 5$$

Precision

$$Precision = \frac{TP}{TP + FP} \quad 6$$

Recall

$$Recall = \frac{TP}{TP + FN} \quad 7$$

F1-Score

3.10. $\boldsymbol{F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}}$ **8**

Confusion Matrix Analysis

The confusion matrix evaluates classification performance by comparing predicted and actual labels.

Confusion Matrix Representation

$$CM = \begin{bmatrix} TP & FP \\ FN & TN \end{bmatrix} \quad 9$$

3.11. Prediction System

The prediction system processes new input reviews and classifies them into sentiment categories. This demonstrates the practical applicability of the model and its ability to generalize to unseen data.

3.12. Summary of Methodology

This chapter presented a structured methodology for sentiment classification, integrating preprocessing, feature engineering, and LSTM-based modelling. The proposed approach ensures accurate and scalable sentiment classification, forming the foundation for the results presented in the next chapter.

4. Results and Discussion

4.1. Introduction

This chapter presents the results obtained from the implementation of the proposed sentiment classification system. The outcomes are derived after applying the complete processing pipeline, including data preparation, feature extraction, sequence transformation, and LSTM-based model training. The objective is to evaluate how effectively the model classifies customer reviews into multiple sentiment categories based on learned patterns. The analysis includes both dataset-level observations and model-level performance evaluation. Statistical summaries and graphical representations are used to understand sentiment distribution and dataset characteristics. Further, performance metrics, training curves, and confusion matrices provide insights into model behavior and prediction accuracy, validating the effectiveness of the system.

4.2. Dataset Description

Table 7 Overview of dataset attributes, including record count, feature composition, input and target variables, and sentiment class distribution.

Attribute	Description
Total Records	21,904
Total Features	5
Input Feature	Review Text
Target Feature	Sentiment
Sentiment Classes	Positive, Negative, Neutral

The dataset contains 21,904 customer reviews from an e-commerce platform, with review text as the input and sentiment labels (positive, negative, neutral) as the target for classification. Initial analysis shows the dataset is mostly complete, with only a few missing values in the summary column. Additional attributes provide context but are not used for modelling, ensuring a reliable dataset for effective sentiment classification.

The above Table 7 outlines the fundamental characteristics of the dataset utilized in this study. The corpus comprises 21,904 records distributed across the five features, with the Review Text and serving as the primary input variable and Sentiment designated as the target label.

4.3. Exploratory Data Analysis

Exploratory Data Analysis (EDA) is conducted to understand dataset characteristics and identify patterns influencing sentiment classification. It includes analysis of sentiment distribution, review length variations, and relationships between textual features such as character count, word count, and sentence count. These analyses provide insights into how users express opinions across different sentiment categories. Both statistical summaries and graphical representations are used to interpret dataset structure and support model evaluation.

4.3.1. Statistical Summary of Features

Statistical analysis of textual features provides insights into review structure and composition. Features such as character count, word count, and sentence count help in understanding variation in review length and textual patterns. The results indicate that most reviews are short and concise, with minimal variability in sentence structure. This consistency is beneficial for sentiment analysis tasks, as it simplifies pattern recognition for the model.

Table 8 Descriptive Statistics of Text Units

Statistic	**num_characters**	**num_words**	**num_sentences**
Count	21904	21904	21904
Mean	11.982834	2.010683	1.000730
Standard Deviation	6.011409	1.012804	0.028658
Minimum	3	1	1
25th Percentile	8	1	1
Median (50%)	11	2	1
75th Percentile	16	2	1
Maximum	62	13	3

The above table 8 displays about the dataset comprises 21,904 text units, each analysed for character count, word count, and sentence count. On average, a text unit contains approximately 12 characters, 2 words, and 1 sentence, indicating that the corpus is dominated by short expressions.

4.3.2. Sentiment Distribution Analysis

The distribution of sentiment classes provides an overview of how reviews are categorized within the dataset. Understanding this distribution ensures that the dataset is balanced and suitable for training the model. The dataset shows nearly equal proportions of positive, negative, and neutral reviews, which helps prevent bias during model training and improves classification performance.

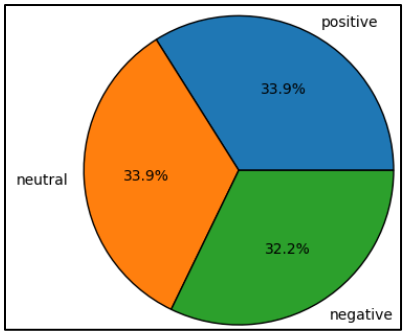

Figure 2 Sentiment distribution in dataset with nearly equal proportions of positive, neutral & negative reviews

Fig 2 shows the sentiment distribution across the dataset, with positive and neutral reviews each accounting for 33.9% and negative reviews comprising 32.2%. This balanced distribution ensures fair representation of all sentiment categories for model training and evaluation.

4.3.3. Review Length Analysis

The analysis of review length based on characters and words provides insights into how users express their opinions. Shorter reviews tend to reflect quick feedback, while longer reviews may include detailed explanations. The results indicate that most reviews are short, with variations across sentiment categories. Positive reviews are often brief, whereas negative reviews tend to be slightly longer, reflecting more detailed dissatisfaction.

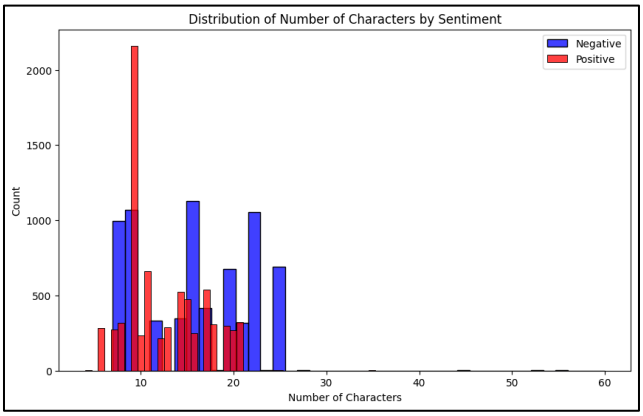

Figure 3 Distribution of character counts across reviews by sentiment category

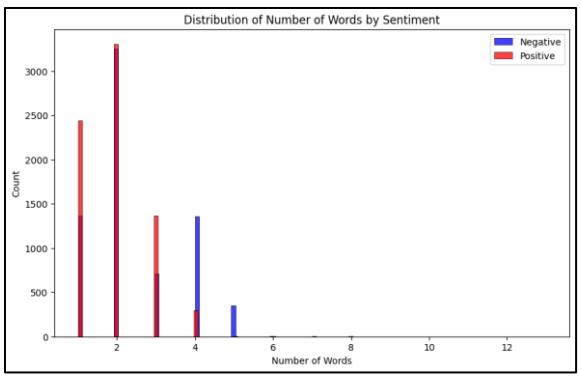

Figure 4 Distribution of word counts across reviews by sentiment category

Fig 3 illustrates the distribution of review character counts across sentiments. Positive reviews tend to cluster around shorter lengths, while negative reviews show broader variation with multiple peaks, highlighting differences in expression patterns.

Figure 4 shows the distribution of word counts across sentiments. Positive reviews are concentrated in shorter texts, while negative reviews appear more frequently in slightly longer texts, highlighting differences in sentiment expression length.

4.3.4. Feature Relationship Analysis

Feature relationship analysis examines how textual attributes interact with each other. This helps in identifying dependencies between features such as character count, word count, and sentence count. The analysis shows strong correlations between character and word counts, while sentence counts remain relatively stable. These relationships provide deeper insights into how textual features vary with sentiment.

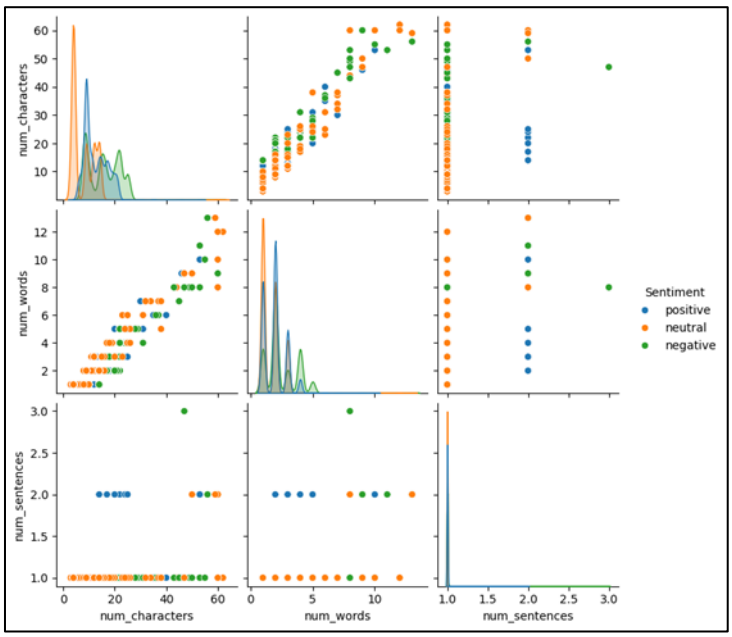

Figure 5 Pair plot showing relationships among text features by sentiment.

Figure 5 presents the pair plot analysis of text features characters, words, and sentences across sentiments. It highlights how these structural attributes vary by sentiment category, revealing correlations and distributional differences that support deeper understanding of review patterns.

4.4. Text Preprocessing and Feature Engineering

The preprocessing stage ensures that textual data is converted into a structured format suitable for analysis. Steps such as handling missing values, duplicate analysis, normalization, and tokenization improve data quality. These preprocessing steps reduce noise and enhance the quality of input data, enabling the model to learn meaningful patterns effectively.

Table 9 Impact of Preprocessing Steps on Text Quality

Preprocessing Step	Operation Performed	Effect on Text Data
Missing Value Handling	Null values replaced with empty strings	Ensures dataset completeness
Duplicate Analysis	Identified repeated reviews	Maintains data distribution awareness
Lowercase Conversion	Converted all text to lowercase	Reduces vocabulary variation
Tokenization	Split text into meaningful tokens	Enables structured processing
Stop word Removal	Removed common words	Focuses on meaningful terms
Text Transformation	Cleaned and filtered text	Improves input quality for model

The table 9 outlines the key preprocessing steps applied to the dataset. It highlights operations such as handling missing values, removing duplicates, converting text to lowercase, tokenization, stop word removal, and text transformation, all of which collectively enhance text quality and prepare the data for effective sentiment analysis.

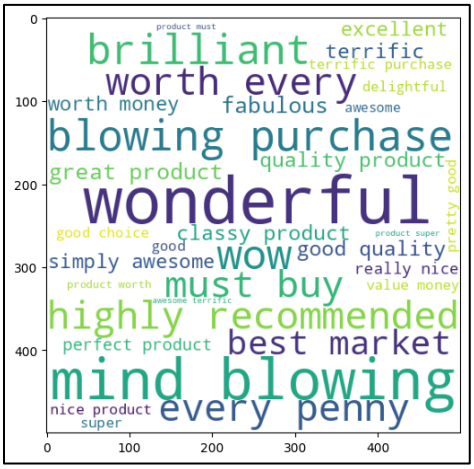

Figure 6 Word cloud of frequently used positive expressions in product reviews.

Figure 7 Word cloud of frequently used negative expressions in product reviews.

Figure 8 Word cloud of frequently used neutral expressions in product reviews.

Fig6 displays the word cloud of positive sentiments, emphasizing frequently used expressions such as “wonderful,” “highly recommended,” and “must buy.” The visualization highlights the dominant language patterns in favorable reviews, showcasing the strength of positive consumer perception.

Fig 7 presents the word cloud of negative sentiments, emphasizing frequent expressions such as “waste money,” “terrible product,” and “utterly disappointed.” This visualization highlights the dominant language patterns of dissatisfaction, reflecting consumer frustration and poor product experiences.

Fig 8 presents the word cloud of neutral sentiments, with terms like “nice,” “fair,” and “decent” appearing most prominently. This visualization captures the balanced tone of moderate reviews, reflecting consumer perceptions that are neither strongly positive nor negative.

4.5. Text Representation and Model Configuration

The processed text is converted into numerical sequences to enable compatibility with the LSTM model. Tokenization converts words into sequences, and padding ensures uniform input length across all samples. The dataset is divided into training and testing subsets, and sentiment labels are encoded numerically. These steps ensure efficient processing and improved model performance.

4.6. Model Performance Evaluation

The performance of the model is evaluated using metrics such as accuracy, precision, recall, and F1-score. The model achieved a high accuracy of 96.94%, indicating strong predictive capability. The results show balanced performance across all sentiment classes, confirming that the model effectively handles multi-class classification tasks.

Table 10 Precision, recall, and F1-scores for sentiment classification across negative, neutral, and positive classes

Class	Precision	Recall	F1
Negative	0.9941	0.9532	0.9732
Neutral	0.9286	0.9906	0.9586
Positive	0.9910	0.9637	0.9772

Table 10 reports the classification performance across sentiment classes. Negative, neutral, and positive categories all achieve high precision, recall, and F1 scores, demonstrating the robustness and reliability of the model in sentiment prediction.

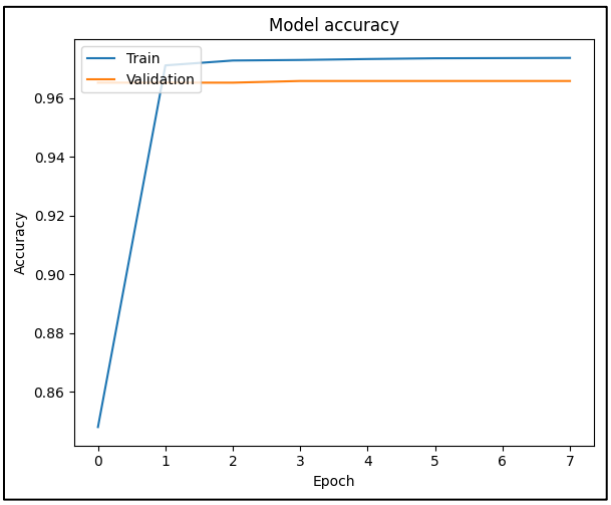

Figure 9 Training and validation accuracy trends across epochs

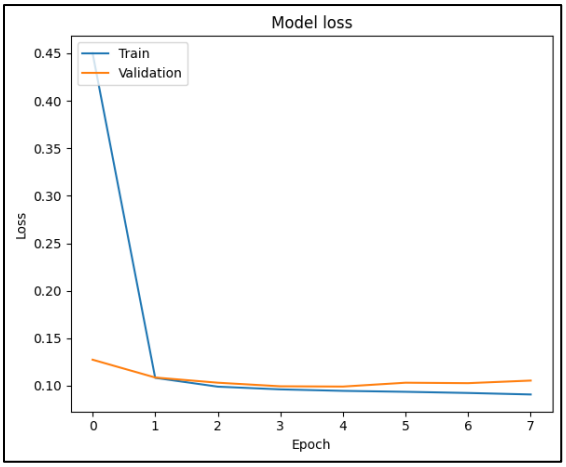

Figure 10 Training and validation loss trends across epochs.

Fig 9 illustrates the model accuracy across epochs, showing rapid improvement in training accuracy that stabilizes near 0.97, while validation accuracy remains consistently high around 0.96. This indicates strong generalization with minimal overfitting.

Figure 10 illustrates the model loss across epochs, showing a sharp decline in training loss that stabilizes near 0.10, while validation loss remains consistently low around 0.13. This demonstrates effective learning and strong generalization performance.

4.7. Confusion Matrix Analysis

The confusion matrix provides a detailed evaluation of classification performance by comparing actual and predicted values. Diagonal values represent correct predictions, while off-diagonal values indicate misclassifications. The results show strong diagonal dominance, indicating high accuracy and effective classification across all sentiment categories.

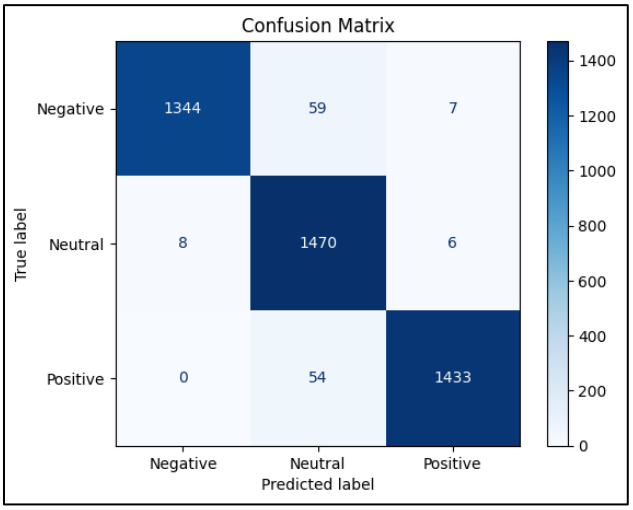

Figure 11 Confusion matrix illustrating sentiment classification performance across negative, neutral, and positive categories

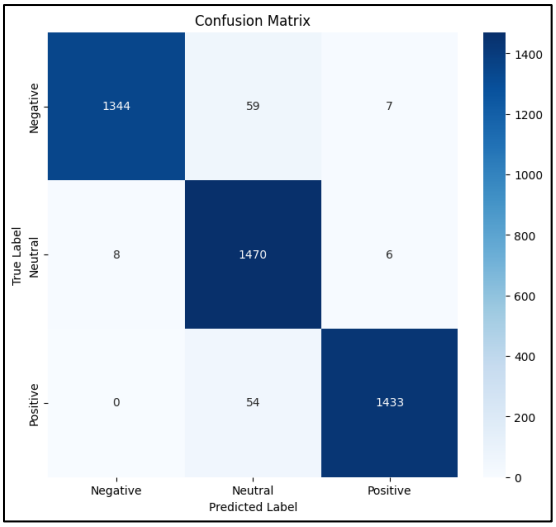

Figure 12 Confusion matrix depicting sentiment classification outcomes across negative, neutral & positive categories

Fig 11 shows the confusion matrix for sentiment classification, where most samples are correctly predicted across negative, neutral, and positive classes. The strong diagonal dominance highlights the model's high accuracy and reliable performance in distinguishing sentiments.

Figure 12 presents the confusion matrix for sentiment classification, clearly showing strong diagonal values where predictions align with true labels. The minimal off-diagonal entries highlight the model's high accuracy and effective distinction among negative, neutral, and positive classes.

4.8. Prediction System Results

The prediction system demonstrates the model's ability to classify new input reviews. Input text undergoes preprocessing and transformation before being passed to the trained model. The model outputs sentiment labels along with confidence scores, confirming its ability to generalize and perform effectively on unseen data.

5. Discussion of Results

The results indicate that the LSTM model effectively captures sentiment patterns from textual data. High accuracy and balanced performance across classes demonstrate model robustness. Preprocessing, feature extraction, and sequence modelling contribute significantly to improved performance, ensuring reliable sentiment classification.

Table 11 Key performance observations and interpretations of sentiment classification model

Observation	Interpretation
High Accuracy (96.94%)	Model effectively captures sentiment patterns
Balanced Precision & Recall	Model performs consistently across classes
Strong F1-Score	Good balance between precision and recall
Stable Training Curves	Model avoids overfitting
Confusion Matrix Diagonal Dominance	Most predictions are correct

Table 11 summarizes key evaluation observations, highlighting the model's high accuracy, balanced precision and recall, and strong F1-score. The stable training curves and diagonal dominance in the confusion matrix further confirm effective learning and reliable sentiment classification.

5.1. Conclusion of Results and Discussion

This chapter presented a detailed evaluation of the sentiment classification system. The model achieved high accuracy, confirming its effectiveness in handling multi-class sentiment analysis tasks. The results validate capability of the LSTM model in processing sequential text data and provide a strong foundation for further improvements in future work.

Table 12 Performance metrics summarizing training, validation, test, and final model accuracy.

Performance Metric	Value	Description
Training Accuracy	97.3% (approx.)	Accuracy achieved during training phase
Validation Accuracy	96.5% (approx.)	Performance on validation data
Test Accuracy	96.94%	Performance on unseen test dataset
Final Model Accuracy	96.94%	Overall achieved accuracy of the proposed model

The table 12 presents performance metrics of the proposed sentiment analysis model, showing training, validation, and test accuracies. It highlights the model's strong generalization ability with a final accuracy of 96.94%.

6. Conclusion and Future Work

This study presented a deep learning-based approach for sentiment classification using an LSTM model applied to Flipkart product reviews. The methodology involved multiple stages, including data preprocessing, feature engineering, exploratory data analysis, text representation, and sequence modeling. Each stage contributed to transforming raw textual data into a structured format suitable for classification.

The results demonstrate that the LSTM model effectively captures contextual relationships within text and achieves strong performance across evaluation metrics such as accuracy, precision, recall, and F1-score. The model shows consistent classification performance across different sentiment classes, confirming its ability to generalize well to unseen data and its suitability for real-world sentiment analysis applications.

6.1. Key Findings

The analysis revealed that preprocessing and feature engineering play a crucial role in improving model performance. Handling missing values, analyzing duplicate entries, and extracting structural features such as word count significantly contributed to enhancing the quality of input data and model accuracy.

Additionally, the use of LSTM architecture enabled the model to capture sequential dependencies in textual data more effectively than traditional approaches. The balanced dataset further supported unbiased training, leading to consistent performance across positive, negative, and neutral sentiment classes.

6.2. Limitations of the Study

Despite achieving strong classification performance, certain limitations are associated with the dataset and methodology used in this study. The model is trained on a specific dataset of product reviews, which may not fully represent the diversity of language patterns across different domains or platforms. Variations in writing styles and vocabulary can affect the model's ability to generalize to new datasets.

Another limitation arises from the complexity of natural language, particularly in handling ambiguous or mixed sentiments. Reviews that contain both positive and negative expressions, sarcasm, or implicit meanings may not always be accurately classified. Additionally, the model relies primarily on textual features and does not incorporate external contextual information, which may limit its overall understanding.

6.3. Future Work

Future work can focus on enhancing the model by incorporating advanced deep learning architectures such as BiLSTM, GRU, or transformer-based models like BERT. These models have the potential to capture more complex contextual relationships and improve classification accuracy, especially for ambiguous and context-dependent text.

Further improvements can include expanding the dataset to cover multiple domains and languages, integrating multimodal data such as images or ratings, and applying hyperparameter optimization techniques. Additionally, incorporating attention mechanisms and real-time sentiment analysis systems can enhance both performance and practical applicability of the model.

This research successfully demonstrated the application of deep learning techniques for sentiment classification of customer reviews. The integration of preprocessing, feature engineering, and LSTM-based modeling resulted in a robust and effective classification system.

Overall, the study provides a strong foundation for future research in sentiment analysis and highlights the potential of deep learning approaches in handling complex textual data. The findings contribute to both academic research and practical applications in areas such as customer feedback analysis and recommendation systems

Compliance with ethical standards

Acknowledgments

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

Disclosure of conflict of interest

The author declares that there are no competing interests.

References

[1] B. Liu, “Sentiment Analysis and Opinion Mining,” Morgan & Claypool Publishers, 2012.

[2] Y. Goldberg, “A Primer on Neural Network Models for Natural Language Processing,” Oct. 2015, [Online]. Available: http://arxiv.org/abs/1510.00726

[3] A. Go, R. Bhayani, and L. Huang, “Twitter Sentiment Classification using Distant Supervision.” [Online]. Available: http://tinyurl.com/cvvg9a

[4] M. Hu and B. Liu, “Mining and Summarizing Customer Reviews,” in Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD ’04), Seattle, Washington, USA: ACM, 2004, pp. 168–177. doi: 10.1145/1014052.1014073.

[5] B. Pang, ¡£¢, L. Lee, ¡¤¢, and ¡£¥, “Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales,” 2005. [Online]. Available: http://www.sover.net/

[6] S. Angelidis and M. Lapata, “Multiple Instance Learning Networks for Fine-Grained Sentiment Analysis.” [Online]. Available: https://github.com/stangelid/milnet-sent

[7] J. Y. Lee and F. Dernoncourt, “Sequential Short-Text Classification with Recurrent and Convolutional Neural Networks.”

[8] B. Pang and L. Lee, “A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts.” [Online]. Available: www.cs.cornell.edu/people/pabo/movie-

[9] V. Hatzivassiloglou and K. R. McKeown, “Predicting the Semantic Orientation of Adjectives,” in Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL), Association for Computational Linguistics, 1997, pp. 174–181.

[10] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, “Hierarchical Attention Networks for Document Classification.”

[11] J. Pennington, R. Socher, and C. D. Manning, “GloVe: Global Vectors for Word Representation.” [Online]. Available: http://nlp.

[12] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, “Natural Language Processing (almost) from Scratch,” Mar. 2011, [Online]. Available: http://arxiv.org/abs/1103.0398

[13] Q. V. Le and T. Mikolov, “Distributed Representations of Sentences and Documents,” May 2014, [Online]. Available: http://arxiv.org/abs/1405.4053

[14] O. Irsoy and C. Cardie, "Opinion Mining with Deep Recurrent Neural Networks," in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar: Association for Computational Linguistics, 2014, pp. 720–728.

[15] A. Vaswani et al., "Attention Is All You Need," Aug. 2023, [Online]. Available: http://arxiv.org/abs/1706.03762

[16] M. E. Peters, M. Neumann, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations." [Online]. Available: http://allennlp.org/elmo

[17] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information." [Online]. Available: http://www.isthe.com/chongo/tech/comp/fnv

[18] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent Trends in Deep Learning Based Natural Language Processing," Nov. 2018, [Online]. Available: http://arxiv.org/abs/1708.02709

[19] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A Survey of Transformers," Jun. 2021, [Online]. Available: http://arxiv.org/abs/2106.04554

[20] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of Tricks for Efficient Text Classification," Aug. 2016, [Online]. Available: http://arxiv.org/abs/1607.01759

[21] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment Classification using Machine Learning Techniques." [Online]. Available: http://reviews.imdb.com/Reviews/

[22] P. D. Turney, "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews." [Online]. Available: http://www.google.com

[23] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning Word Vectors for Sentiment Analysis."

[24] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," Sep. 2013, [Online]. Available: http://arxiv.org/abs/1301.3781

[25] Y. Kim, "Convolutional Neural Networks for Sentence Classification," Sep. 2014, [Online]. Available: http://arxiv.org/abs/1408.5882

[26] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," Dec. 2014, [Online]. Available: http://arxiv.org/abs/1409.3215

[27] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A Search Space Odyssey," IEEE Trans. Neural Netw. Learn. Syst., vol. 28, no. 10, pp. 2222–2232, Oct. 2017, doi: 10.1109/TNNLS.2016.2582924.

[28] P. Liu, S. Joty, and H. Meng, "Fine-grained Opinion Mining with Recurrent Neural Networks and Word Embeddings," Association for Computational Linguistics, 2015. [Online]. Available: https://github.com/ppfliu/opinion-target

[29] J. Devlin, M.-W. Chang, K. Lee, K. T. Google, and A. I. Language, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." [Online]. Available: https://github.com/tensorflow/tensor2tensor

[30] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to Fine-Tune BERT for Text Classification?," Feb. 2020, [Online]. Available: http://arxiv.org/abs/1905.05583

[31] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," Jan. 2017, [Online]. Available: http://arxiv.org/abs/1412.6980

[32] L. Zhang, S. Wang, and B. Liu, "Deep Learning for Sentiment Analysis: A Survey," Wiley Interdiscip. Rev. Data Min. Knowl. Discov., vol. 8, no. 4, p. e1253, 2018, doi: 10.1002/widm.1253.

[33] R. C. Staudemeyer and E. R. Morris, "Understanding LSTM -- a tutorial into Long Short-Term Memory Recurrent Neural Networks," Sep. 2019, [Online]. Available: http://arxiv.org/abs/1909.09586

[34] N. Srivastava, G. Hinton, A. Krizhevsky, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," 2014.

[35] P. Zhou, Z. Qi, S. Zheng, J. Xu, H. Bao, and B. Xu, "Text Classification Improved by Integrating Bidirectional LSTM with Two-dimensional Max Pooling."

[36] S. Veera Venkata Syam Prakash, P. Durga Prasad, and S. Kumar Duvvuri, "YouTube comment sentiment analysis using hybrid BI-LSTM and GloVE embeddings", doi: 10.33545/27076571.2026.v7.i4a.289.

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

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

[39] Kuppili Nikhita, Dondapati Sasi Prasanna, and Suneel Kumar Duvvuri, "Milk Quality Prediction using Machine Learning and Deep Learning Techniques," International Journal of Scientific Research in Computer Science, Engineering and Information Technology, vol. 12, no. 2, pp. 434–446, Apr. 2026, doi: 10.32628/CSEIT26121370.