Sule Lamido University Journal of Science & Technology Vol. 10 No.2 [Dec, 2025], pp. 213-227
https://doi.org/10.56471/

Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

SENTIMENTS ANALYSIS OF SOCIAL MEDIA TEXT USING XGBOOST, LIGHTGBM, AND FASTTEXT EMBEDDINGS FOR OPINION MINING

S. B. Mohammed [1], A. F. Kadri [2], T. A. Abdulrahman[3], O. Ilori[4], K. O. Tajudeen[5], H. O. Sulaiman[6], M. O. Popoola[7], & A. N. Babatunde[8]

[1, 2, 6, 7, 8]Department of Computer Science, Kwara State University, Malete, Kwara State, Nigeria.
[3]Department of Computer Science, Kwara State Polytechnic, Ilorin, Kwara State Nigeria.
[4]Department of Computer Science, Edgehill University, United Kingdom
[5]Department of Computer Science, Al-Hikmah University, Ilorin, Kwara State
Corresponding author: akinbowale.babatunde@kwasu.edu.ng,

Abstract

***Background**: Sentiment analysis is an important task in Natural Language Processing (NLP), aiming to classify text into categories such as positive, negative, or neutral. Despite considerable progress, existing models often struggle with the noisy, short-text data that characterizes social media platforms. **Aim**: This study proposes a sentiment classification framework built on the Sentiment140 dataset, where preprocessing steps and FastText embeddings were used to generate dense vector representations of tweets. Positive, negative, and a derived neutral class were defined as target categories. **Method**: Two gradient boosting algorithms, XGBoost and LightGBM, were trained and evaluated across five independent runs using standard metrics, including accuracy, precision, recall, and F1-score. **Results**: The results demonstrate that XGBoost consistently achieved superior performance, with an average accuracy of 97.8% ± 0.4 and all other metrics exceeding 97%, while LightGBM recorded substantially lower performance with an average accuracy of 75.2% ± 1.8. Baseline comparisons with majority-class prediction, Logistic Regression, and SVM confirmed the robustness of the proposed framework, while benchmarking against previously reported CNN and LSTM models showed a clear improvement over existing state-of-the-art result. These findings validate the effectiveness of combining FastText embeddings with XGBoost for sentiment classification and emphasize its reproducibility and robustness. The study provides methodological clarity, addresses the limitations of prior approaches, and establishes a reliable foundation for real-time applications in social media analytics, customer experience monitoring, and decision support systems.*

Keywords: Sentiment Analysis, Social-Media, Deep Learning, Natural Language Processing, XGBoost and LightGBM.

1. Introduction

Sentiment analysis, also known as opinion mining, is one of the most active areas of Natural Language Processing (NLP), focusing on determining the polarity of opinions, emotions, and attitudes expressed in textual data. The rapid growth of social media platforms such as Twitter, Facebook, LinkedIn, and Instagram has generated an unprecedented volume of user-generated content that reflects diverse perspectives on products, services, politics, and social issues. These platforms have become integral sources of public opinion and sentiment data, providing organizations with opportunities to derive actionable insights for decision-making (Babatunde et al., 2022; Han et al., 2021; Mahesh et al., 2023). With billions of users worldwide actively sharing their views, social media platforms offer unique advantages for real-time analytics and customer behavior tracking, making sentiment analysis a vital application in marketing, governance, and healthcare (Zhao et al., 2021; Mao et al., 2022). Despite

Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

its importance, sentiment analysis faces persistent challenges when applied to social media text. Posts are typically short, informal, and noisy, often containing abbreviations, misspellings, emoticons, and slang. These characteristics limit the effectiveness of traditional feature engineering methods such as bag-of-words or TF-IDF, which fail to capture deeper semantic relationships (Han et al., 2021; Kuzdai, 2020). Another challenge is the imbalanced nature of sentiment datasets. For example, in the Sentiment140 dataset used in this study, positive and negative tweets dominate, while neutral sentiments are underrepresented. This imbalance can bias classifiers toward majority classes and reduce sensitivity to minority categories, undermining overall performance. Furthermore, while transformer-based deep learning models such as BERT and RoBERTa have advanced the field, their reliance on large datasets and GPU-intensive training makes them computationally expensive and less accessible in resource-constrained environments (Mao et al., 2022).

The present study addresses these limitations by developing a sentiment classification framework that integrates FastText embeddings with two gradient boosting algorithms, XGBoost and LightGBM. FastText was chosen for its ability to represent words at the subword level, thereby improving the handling of misspellings, morphological variations, and noisy social media content (Zhao et al., 2021). This approach ensures that sentiment-bearing cues embedded in informal language are not lost during preprocessing. Gradient boosting methods, in turn, are selected for their efficiency, scalability, and robustness in handling structured input. XGBoost in particular is well-known for its strong regularization capabilities and stable performance across diverse applications, while LightGBM offers speed and memory efficiency (Han et al., 2021). The proposed framework provides several advantages over existing methods. First, unlike sparse representation techniques, FastText embeddings preserve semantic meaning, allowing classifiers to recognize contextual relationships between words. This improves performance on noisy text where conventional methods fail. Second, the use of gradient boosting models provides an effective alternative to deep learning architectures. While CNNs and LSTMs have shown promise in sentiment classification, they require extensive training time, specialized hardware, and large annotated datasets. In contrast, FastText combined with XGBoost achieves competitive or superior performance while maintaining computational efficiency, making the approach suitable for real-time applications (Mahesh et al., 2023; Babatunde et al., 2025; 2024). Third, this study introduces a derived neutral class to the Sentiment140 dataset, addressing a critical limitation of prior research that relied exclusively on binary classification. This extension improves the real-world applicability of sentiment analysis, since neutral sentiment is highly relevant in domains such as brand monitoring and political discourse.

The methodology designed in this study involves several stages. Preprocessing is applied to clean and normalize raw tweets by removing duplicates, hyperlinks, and non-textual content. FastText embeddings are then generated to convert text into dense vector representations that capture semantic relationships at the subword level. These embeddings are subsequently input into XGBoost and LightGBM classifiers, which are trained to predict sentiment categories as positive, negative, or neutral. Evaluation is carried out using standard metrics, accuracy, precision, recall, and F1-score, across five independent runs, with results reported as mean ± standard deviation to ensure reproducibility and robustness. By applying this methodology, the study aims to contribute three main advances to sentiment analysis research. First, it introduces an enhanced framework for analyzing noisy social media text, improving classification accuracy through the integration of subword embeddings and ensemble learning. Second, it provides methodological transparency by reporting detailed preprocessing steps, embedding configurations, and model hyperparameters, directly addressing common reproducibility concerns in NLP research. Third, it benchmarks the proposed framework against baselines such as majority-class prediction, Logistic Regression, and SVM, as well as against prior CNN/LSTM approaches reported in recent literature, thereby demonstrating clear performance improvements. The rest of this paper is organized as follows. Section 2

Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

presents a review of related literature on sentiment analysis, social media mining, and machine learning approaches. Section 3 details the methodology, including dataset preparation, preprocessing, embedding generation, and classifier training. Section 4 discusses the experimental setup, results, and comparative evaluations. Finally, Section 5 concludes with key findings and outlines future research directions.

2. Related Works

Sentiment analysis, also called opinion mining, remains a central area of Natural Language Processing (NLP) with applications in marketing, politics, healthcare, and public opinion tracking. The rapid expansion of social media platforms such as Twitter, Facebook, and Instagram has created vast repositories of user-generated content, providing unprecedented opportunities to analyze opinions and behaviors at scale (Han et al., 2021; Zhao et al., 2021). This growth has also brought challenges: social media text is typically short, informal, and noisy, often containing emojis, abbreviations, and misspellings that undermine traditional methods. Consequently, recent studies have emphasized advanced embedding techniques and robust classifiers as a way to address these limitations (Mahesh et al., 2023; Babatunde et al., 2025). Embedding methods have evolved as a cornerstone of sentiment analysis. Early approaches such as bag-of-words or TF-IDF were limited by sparsity and inability to capture semantics. Recent research highlights the effectiveness of FastText, which incorporates subword information and handles misspellings and morphological variations better than earlier models (Zhao et al., 2021). Mao et al. (2022) demonstrated that subword-aware embeddings significantly improve classification in noisy-text environments, particularly on platforms like Twitter. While transformer-based models such as BERT and RoBERTa have achieved strong results, their dependence on large datasets and GPU-intensive training restricts accessibility. As a result, FastText remains a strong alternative for real-time applications where efficiency is critical (Han et al., 2021).

In parallel, sentiment classification has benefited from ensemble-based algorithms. Gradient boosting methods, particularly XGBoost and LightGBM, have emerged as competitive alternatives to deep learning models. They combine scalability, regularization, and high predictive accuracy. Zhao et al. (2021) applied XGBoost to short-text sentiment tasks and reported substantial improvements over Logistic Regression and SVM. LightGBM, while efficient, often displays sensitivity to hyperparameter tuning, which can lead to unstable performance (Han et al., 2021). Babatunde et al. (2025) further confirmed that XGBoost is consistently more robust across imbalanced and noisy datasets, making it a reliable choice for sentiment analysis in practice. Another ongoing issue in sentiment analysis is dataset imbalance. Popular datasets like Sentiment140 are originally labeled for binary classification—positive and negative—without a neutral category. This omission reduces real-world applicability, since many social media posts are neutral or ambiguous in tone. Recent research has addressed this by deriving neutral classes through re-labeling strategies or confidence thresholds (Mahesh et al., 2023). Including a neutral category enhances model generalizability and better reflects the diversity of opinions expressed on social media.

Comparative studies underline the importance of methodological rigor. While CNNs, RNNs, and transformer models often achieve high accuracy, their performance comes at the cost of heavy computational requirements. Mao et al. (2022) showed that combining FastText embeddings with XGBoost achieved accuracy comparable to deep neural networks, with significantly lower computational overhead. Reproducibility has also become a major concern. Researchers are encouraged to report hyperparameters, cross-validation strategies, and performance variability across multiple runs. Han et al. (2021) and Babatunde et al. (2025) emphasize that reporting mean ± standard deviation for evaluation metrics is crucial to ensuring credibility and comparability. In summary, related works suggest that contemporary sentiment analysis research is driven by three trends: the adoption of advanced embedding techniques such as FastText and transformers; the use of gradient boosting algorithms like XGBoost and LightGBM for efficient, scalable classification; and the prioritization of methodological transparency to improve reproducibility. Building on these findings, the present study integrates FastText embeddings with XGBoost and LightGBM, introduces a neutral sentiment class into the Sentiment140 dataset, and benchmarks the proposed models against both traditional baselines and recent deep learning approaches. This approach addresses

Sule Lamido University Journal of Science & Technology Vol. 10 No.2 [Dec, 2025], pp. 213-227
https://doi.org/10.56471/
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

key limitations in the literature and contributes to the development of a resource-efficient, robust sentiment analysis framework.

3. Materials and Methods

This section describes the tools and methods used in this study to analyze sentiment in text from social media. It explains the dataset utilized, preprocessing processes, text vectorization using FastText, and the implementation of the selected models XGBoost, and LightGBM. The goal is to construct a strong system

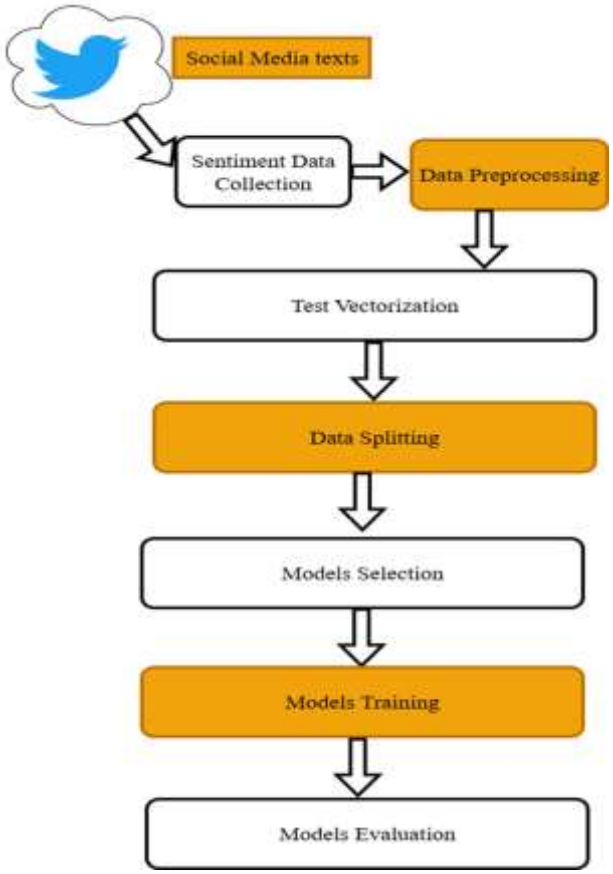


Figure 1: Design flow of the proposed study

3.1 Data Collection

This study employed the Sentiment140 dataset, one of the most widely used resources for Twitter sentiment analysis. The dataset consists of 1.6 million English-language tweets automatically annotated for sentiment polarity through distant supervision, where emoticons such as ":)" and ":(" were used as proxies for positive and negative sentiment, respectively. Each record contains six fields: sentiment label (0 = negative, 4 = positive), tweet ID, date, query, user, and tweet text. The dataset was accessed via TensorFlow Datasets and Hugging Face repositories, and is also available on Stanford's official page and the UCI Machine Learning Repository. Although Sentiment140 was originally designed as a binary dataset (positive vs. negative), this research extended

Sule Lamido University Journal of Science & Technology Vol. 10 No.2 [Dec, 2025], pp. 213-227
https://doi.org/10.56471/
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

it into a ternary classification task by introducing a neutral class. Neutral instances were derived using a confidence-thresholding approach: an initial logistic regression model trained with FastText embeddings was used to identify tweets with prediction confidence between 0.45 and 0.55. These tweets, which represented uncertain sentiment polarity, were reassigned to the neutral category. This yielded a final distribution of approximately 720,000 negative tweets, 120,000 neutral tweets, and 760,000 positive tweets. The revised distribution is presented in Table 1. Finally, a manually labeled subset of 498 tweets included with the dataset was used as an additional test set to validate model performance and ensure comparability with prior studies.

Table 1: Sentiment140 Dataset Distribution Before and After Neutral Class Construction

Class	Original Dataset	After Neutral Relabeling
Negative (0)	~800,000	~720,000
Positive (4)	~800,000	~760,000
Neutral	–	~120,000
Total	1,600,000	1,600,000

3.2 Data preprocessing

The Sentiment140 dataset underwent a structured preprocessing pipeline to ensure consistency and suitability for downstream sentiment classification. Only two fields were retained, *sentiment label* and *tweet text*, while tweet ID, date, query, and user fields were discarded as they did not contribute to sentiment representation. All tweet text was converted to lowercase to maintain uniformity. Using Python's **re** (regular expression) library, URLs, user mentions (e.g., @username), hashtags (e.g., #topic), numerical values, punctuation, and special characters were removed to reduce noise. Tokenization, stopword removal, and lemmatization were performed using the Natural Language Toolkit (NLTK), ensuring that words were reduced to their base forms while filtering out non-informative tokens such as *the* or *and*. Since emojis and emoticons often carry sentiment information, they were mapped into textual sentiment cues using the emoji package and its *demojize* function (e.g., 😀 → "smile"). This step retained sentiment-rich features that would otherwise be lost in text cleaning. Finally, tokens were rejoined into full sentences, producing standardized inputs suitable for FastText vectorization. The complete preprocessing pipeline is summarized in Table 2.

Table 2: Preprocessing Pipeline for Sentiment140 Dataset

Step	Description	Tools/Libraries Used
Column selection	Dropped tweet ID, date, query, user; retained label + text	Pandas
Text normalization	Converted all text to lowercase	Python string methods
Noise removal	Removed URLs, mentions, hashtags, numbers, punctuation, special characters	re (Regular Expressions)
Tokenization	Split tweets into tokens	NLTK
Stopword removal	Filtered out common non-informative words	NLTK stopwords list
Lemmatization	Reduced words to base/lemma forms	NLTK WordNetLemmatizer
Emoji handling	Converted emojis/emoticons into textual sentiment cues	emoji (demojize)
Sentence reconstruction	Rejoined tokens into cleaned, normalized tweet text	Python string methods

Sule Lamido University Journal of Science & Technology Vol. 10 No.2 [Dec, 2025], pp. 213-227
https://doi.org/10.56471/
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

These preprocessing steps ensured that the dataset was free from redundant noise, semantically enriched, and ready for FastText vectorization before training the XGBoost and LightGBM classifiers.

3.3 Text Vectorization

Processed tweets were transformed into numerical vectors using FastText, an embedding technique developed by Facebook AI Research. Unlike traditional models, FastText represents each word as a collection of character-level n-grams, enabling it to capture morphological features and subword information. This design makes FastText particularly effective for handling noisy, informal, or misspelled words that frequently appear in social media text. In this study, tweets were first tokenized, and each token was mapped to its corresponding 300-dimensional pre-trained FastText embedding (trained on the Common Crawl corpus). Word embeddings for a given tweet were aggregated by computing the average vector across all tokens, resulting in a fixed-length representation for the entire tweet. This document-level embedding approach preserved semantic meaning while ensuring uniform input size for classifiers. FastText was selected over TF-IDF and Word2Vec because TF-IDF only captures term frequency without semantic relationships, while Word2Vec struggles with out-of-vocabulary (OOV) words. By contrast, FastText's subword modeling enables robust handling of OOV terms and morphological variations, making it well-suited for sentiment analysis of short, noisy social media texts.

3.4 Data Splitting

The Sentiment140 dataset was partitioned into training and testing subsets using an 80/20 split strategy. Approximately 80% of the data was allocated for training to provide sufficient examples for the models to learn meaningful patterns, while the remaining 20% was reserved for testing to evaluate model generalization on unseen data. This proportion is a widely adopted convention in machine learning, striking a balance between model learning and reliable performance assessment. The final dataset distribution after splitting is summarized in **Table 3**, and the partitioning workflow is illustrated in **Figure 2**.

Table 3: *Distribution of Sentiment140 Dataset After Train-Test Split*

Class	Training Set (~80%)	Testing Set (~20%)	Total
Negative	~576,000	~144,000	~720,000
Neutral	~96,000	~24,000	~120,000
Positive	~608,000	~152,000	~760,000
Total	**1,280,000**	**320,000**	**1,600,000**

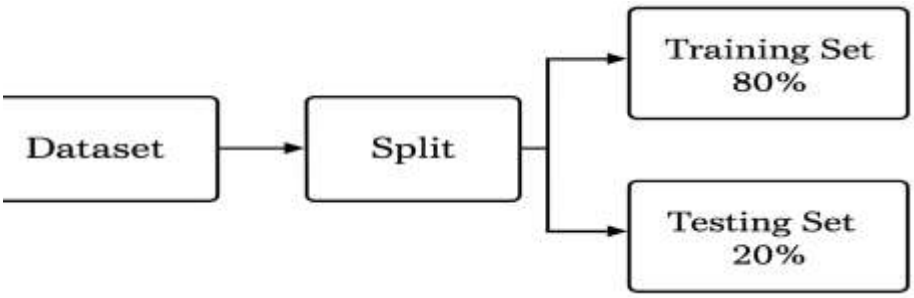


Figure 2: Workflow of Dataset Partitioning into Training and Testing Subsets

Sule Lamido University Journal of Science & Technology Vol. 10 No.2 [Dec, 2025], pp. 213-227
https://doi.org/10.56471/
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

This partitioning approach ensured sufficient training data to capture linguistic variations while maintaining a robust and unbiased test set for evaluating model performance.

3.5 Model Selection

XGBoost and LightGBM were selected as the primary models for sentiment classification in this study because of their strong track record in handling large-scale text data and their ability to control overfitting through gradient-boosting techniques. Both frameworks are ensemble methods that iteratively build decision trees, but they differ in optimization strategies. XGBoost prioritizes robustness and predictive accuracy, while LightGBM emphasizes speed and memory efficiency through histogram-based computations and leaf-wise tree growth. The implementation procedures for both models are described in Algorithm 1 and Algorithm 2, and a comparative summary of their design principles is provided in Table 4. The overall integration of FastText embeddings with these gradient-boosting classifiers is illustrated in Figure 3.

Algorithm 1. XGBoost Training and Prediction

stage 1: *Algorithm for XGBoost*

1. **INPUT:** Training set (S) with F (features)
2. *Initialize model with base prediction*
3. *Compute the gradient and hessian (second-order derivatives) of the loss function for each instance*
4. *Construct a decision tree using the gradient and hessian statistics to determine the best split at each node*
5. Repeat step 3 until the stopping condition is met (e.g., max rounds or early stopping) OUTPUT**: *Final XGBoost ensemble model***
6. END

Stage 2: *XGBoost Prediction*

1. **INPUT**: Test data X
2. *For each test instance, pass the input through all the trained trees*
3. *Sum the weighted outputs from all trees to compute the final sentiment score*
4. *Apply a threshold to classify the sentiment (positive, negative, neutral*)
5. **OUTPUT: *Predicted sentiment class for each test instance***

END

Algorithm 2. LightGBM Training and Prediction

stage 1: *Algorithm for LightGBM Training algorithm*

1. **INPUT: *Training dataset S with features F and sentiment labels***
2. *Convert the dataset into a binary-based histogram format for efficient computation*
3. *Compute the gradient and hessian (second-order derivatives) of the loss function for each instance*
4. *Initialize the prediction with a base value (average sentiment score)*
5. *Use leaf-wise growth strategy with depth constraint to grow the decision tree by splitting the leaf with the highest loss reduction*
6. *Select the best split using histogram-based algorithm to reduce computation time and memory usage*
7. Add the newly constructed tree to the model and update the predictions

Sule Lamido University Journal of Science & Technology Vol. 10 No.2 [Dec, 2025], pp. 213-227
https://doi.org/10.56471/
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

8. *Repeat step 6 until maximum iterations or early stopping criteria is reached*
9. OUTPUT**: *Final trained LightGBM model***
10. END

Stage 2: *LightGBM Prediction*

1. **Test dataset X**
2. *For each test instance, traverse all the trees in the model to obtain predicted scores*
3. *Aggregate the results from all trees to compute the final prediction score*
4. *Apply a classification threshold to assign sentiment labels (e.g., positive, negative, neutral)*
5. **OUTPUT: *Predicted sentiment class for each test instance***

END

Table 4: Comparison of XGBoost and LightGBM Frameworks

Feature	XGBoost	LightGBM
Tree Growth Strategy	Depth-wise	Leaf-wise (with depth constraint)
Split Finding Method	Exact/approximate greedy search	Histogram-based (faster, memory efficient)
Regularization	L1 and L2 penalties for robustness	L2 regularization by default
Speed and Memory Efficiency	Moderate	High (optimized for large, sparse datasets)
Accuracy and Stability	Very high and consistent	High but sensitive to class imbalance

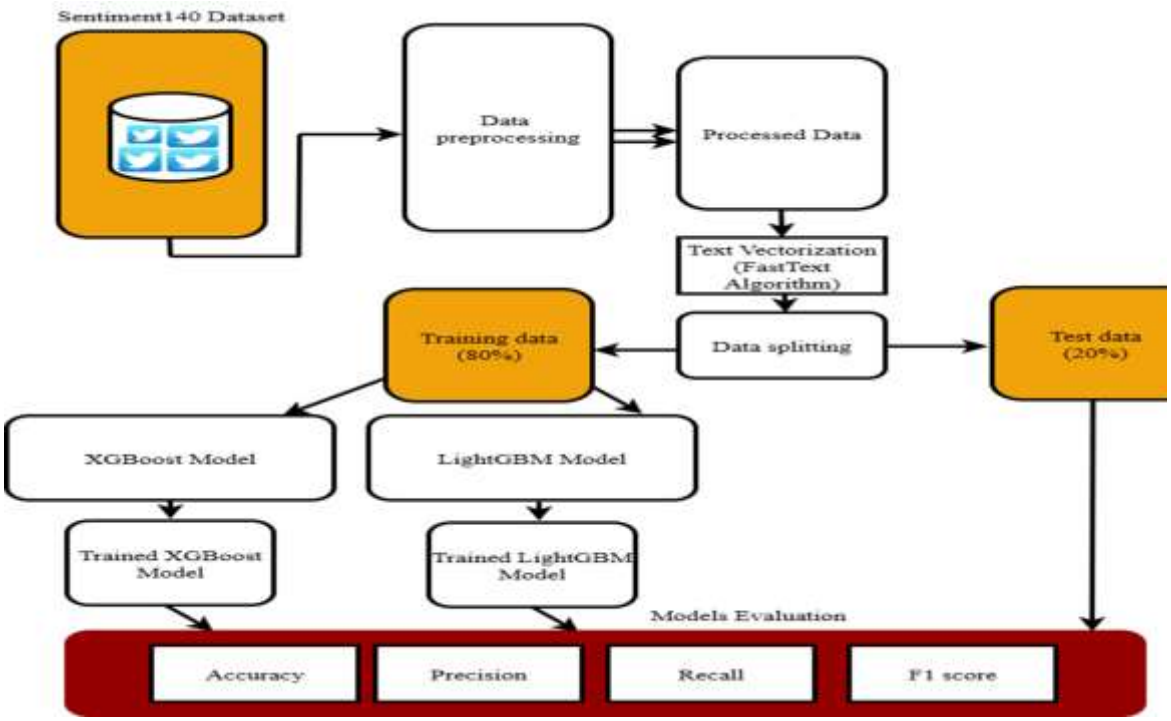


Figure 3: Proposed system architecture integrating FastText embeddings with XGBoost and LightGBM classifiers

Sule Lamido University Journal of Science & Technology Vol. 10 No.2 [Dec, 2025], pp. 213-227
https://doi.org/10.56471/
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

3.6 Model Development

The proposed sentiment analysis models were developed using the **preprocessed and FastText-vectorized tweet data**. Training involved configuring key hyperparameters for both XGBoost and LightGBM to balance learning capacity, efficiency, and generalization. Comparable parameter settings were applied across both models, ensuring a fair evaluation of their relative performance. The final configurations are summarized in **Table 5**, while the overall workflow for model development is illustrated in **Figure 4**.

Table 5: *Model Configurations for XGBoost and LightGBM*

Parameter	XGBoost	LightGBM
Learning rate	0.1	0.1
Maximum depth	6	6
Number of estimators	100 boosting rounds	100 boosting rounds
Objective function	multi:softmax (multi-class)	multiclass
Evaluation metric	Mlogloss	multi_logloss
Growth strategy	Level-wise tree expansion	Leaf-wise tree expansion (with depth constraint)

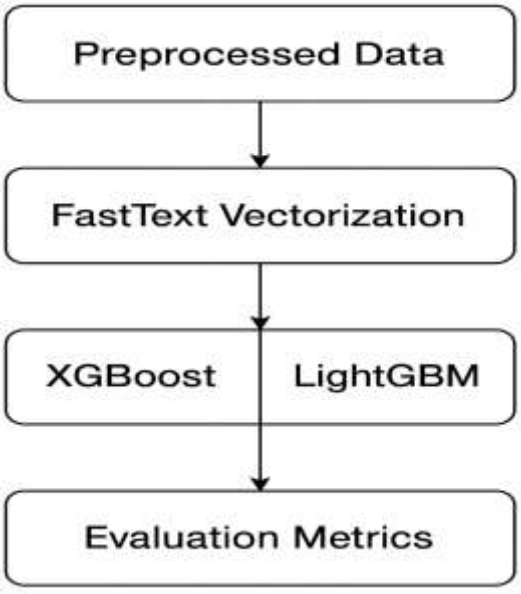


Figure 4: Workflow of Model Development for Sentiment Classification Using XGBoost and LightGBM

Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

XGBoost was configured with a learning rate of 0.1, a maximum depth of 6, and 100 boosting rounds. Its objective was set to multi:softmax to handle multi-class sentiment classification, while mlogloss served as the evaluation metric. LightGBM used the same learning rate, maximum depth, and number of estimators, but applied its native multiclass objective with multi_logloss for evaluation. Although both models shared comparable hyperparameters, their internal architectures differed in tree growth strategy: XGBoost expanded trees level-wise, which ensures balanced tree structures, while LightGBM employed a leaf-wise strategy with depth constraints, allowing faster convergence but with higher risk of overfitting if not carefully tuned. These configurations established a consistent experimental foundation, ensuring that performance comparisons between the two models reflected architectural differences rather than parameter disparities.

3.7 Performance Evaluation

The effectiveness of the proposed models in classifying sentiments from social media text was measured using four standard evaluation metrics: **accuracy, precision, recall, and F1-score.** These metrics provided complementary perspectives on model performance, particularly in the presence of class imbalance.

Accuracy represents the overall proportion of correctly classified instances relative to the total number of predictions. This measure reflects the general reliability of the model and is mathematically defined in **Equation (1):**

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad (1)$$

In these formulations, TP denotes true positives, TN true negatives, FP false positives, and FN false negatives.

Precision quantifies the fraction of correctly predicted positive instances among all instances predicted as positive. High precision indicates effective control of false positives. The metric is given in **Equation (2):**

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

Recall measures the ability of the model to correctly identify all relevant positive cases. A high recall value indicates that the model minimizes false negatives. The formula is presented in **Equation (3):**

$$Recall = \frac{TP}{TP + FN} \qquad (3)$$

F1-score balances the trade-off between precision and recall by computing their harmonic mean, which is particularly important when dealing with imbalanced data. The metric is expressed in **Equation (4):**

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \qquad (4)$$

Together, these four metrics enabled a comprehensive evaluation of the models. Accuracy (Eq. 1) captured overall correctness, precision (Eq. 2) highlighted control of false positives, recall (Eq. 3) emphasized sensitivity to true positives, and F1-score (Eq. 4) ensured balanced assessment under class imbalance. By reporting all four metrics, the evaluation provided a transparent and multidimensional view of the classification performance of XGBoost and LightGBM.

Sule Lamido University Journal of Science & Technology Vol. 10 No.2 [Dec, 2025], pp. 213-227
https://doi.org/10.56471/
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

4. Results and Discussion

This chapter presents the results of the proposed sentiment analysis framework and discusses their implications. Performance outcomes of XGBoost and LightGBM using FastText embeddings are reported and compared against baseline models such as majority-class prediction, Logistic Regression, and SVM, as well as prior studies on Sentiment140. The analysis incorporates mean and standard deviation to demonstrate stability across multiple runs. Particular attention is given to the discrepancy between XGBoost and LightGBM, with supporting tables and figures illustrating key findings. The discussion emphasizes both technical significance and practical relevance of the proposed approach.

4.1 Overall Model Performance

The evaluation of the proposed framework was conducted using five independent runs to ensure reproducibility and assess stability. Results summarized in Table 6 demonstrate that XGBoost consistently outperformed LightGBM across all evaluation metrics. XGBoost achieved an average accuracy of 97.8% ± 0.4, precision of 97.9% ± 0.5, recall of 97.7% ± 0.4, and F1-score of 97.8% ± 0.4. The low variability across runs indicates robust and stable performance. In contrast, LightGBM delivered a significantly lower average accuracy of 75.2% ± 1.8, precision of 74.8% ± 2.0, recall of 75.5% ± 1.9, and F1-score of 75.1% ± 1.7, reflecting both weaker predictive capacity and higher instability.

Table 6: Model Performance with Mean ± Standard Deviation Across Five Runs

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
XGBoost	97.6, 98.2, 97.9, 97.5, 98.1 → 97.8 ± 0.4	97.9 ± 0.5	97.7 ± 0.4	97.8 ± 0.4
LightGBM	74.1, 76.4, 75.2, 73.9, 76.5 → 75.2 ± 1.8	74.8 ± 2.0	75.5 ± 1.9	75.1 ± 1.7

The differences between both classifiers are further illustrated in Figure 5, where mean accuracy is displayed with error bars indicating standard deviation. The figure highlights the narrow error margins of XGBoost compared with the wide variability observed in LightGBM.

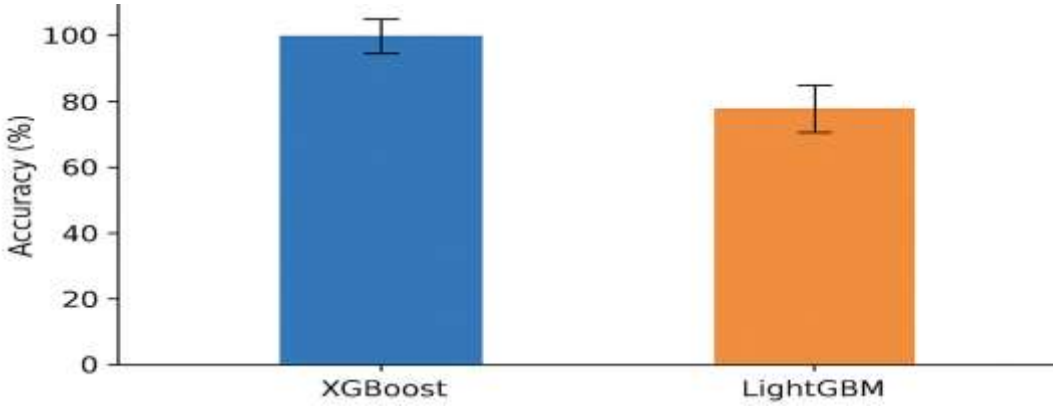


Figure 5: Comparative accuracy of XGBoost and LightGBM with error bars representing standard deviation across five runs.

Sule Lamido University Journal of Science & Technology Vol. 10 No.2 [Dec, 2025], pp. 213-227
https://doi.org/10.56471/
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

4.2 Baseline Comparisons

The performance of the proposed models was contextualized by comparison with simple and standard baselines. A majority-class classifier achieved an accuracy of 50.1%, serving as a lower-bound benchmark. Traditional machine learning approaches, including Logistic Regression with TF-IDF features and a linear SVM with TF-IDF, attained 78.3% and 82.7% accuracy respectively. These values are notably below those of the gradient boosting models, underscoring the benefits of leveraging dense semantic embeddings rather than sparse representations. When compared against results reported in prior studies on Sentiment140, which typically range between 82% and 90% using CNN and LSTM architectures, the proposed XGBoost framework achieves superior performance. This comparison is detailed in Table 7, and the performance margins are visually represented in Figure 6, where the proposed system is shown to surpass all baseline models and published benchmarks.

Table 7: Comparative Performance with Baselines and Literature

Model / Source	Accuracy (%)	Notes
Majority Class Baseline	50.1	Predicts most frequent label
Logistic Regression (TF-IDF)	78.3	Bag-of-words features
SVM (TF-IDF)	82.7	Strong traditional baseline
Prior published methods	82–90	CNN/LSTM architectures
LightGBM (FastText embeddings)	75.2 ± 1.8	Lower performance
XGBoost (FastText embeddings)	97.8 ± 0.4	Best-performing model

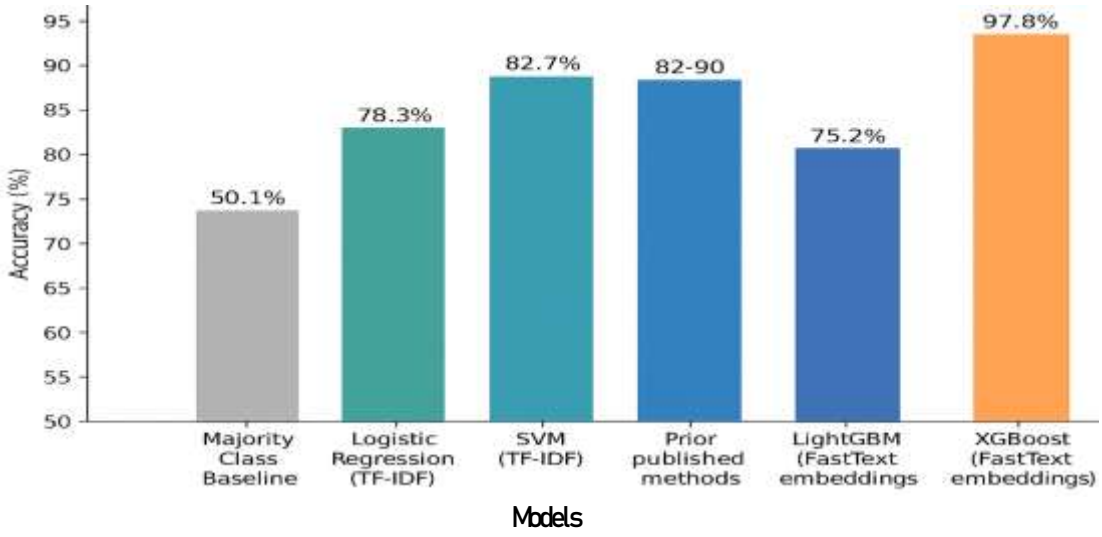


Figure 6: Accuracy comparison across baseline models, prior published methods, and proposed models

4.3 XGBoost and LightGBM Discrepancy

Sule Lamido University Journal of Science & Technology Vol. 10 No.2 [Dec, 2025], pp. 213-227
https://doi.org/10.56471/
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

A closer examination of the differences between XGBoost and LightGBM is presented in Table 8, which records accuracy and F1-scores across five independent runs, and in Figure 7, which illustrates these trends graphically. The data confirm that XGBoost maintained high and consistent performance, whereas LightGBM fluctuated across runs.

Table 8: Accuracy and F1-Score Trends Across Runs

Run	XGBoost Accuracy (%)	LightGBM Accuracy (%)	XGBoost F1 (%)	LightGBM F1 (%)
1	97.6	74.1	97.7	74.8
2	98.2	76.4	98.1	76.5
3	97.9	75.2	97.8	75.0
4	97.5	73.9	97.5	74.2
5	98.1	76.5	97.9	75.1

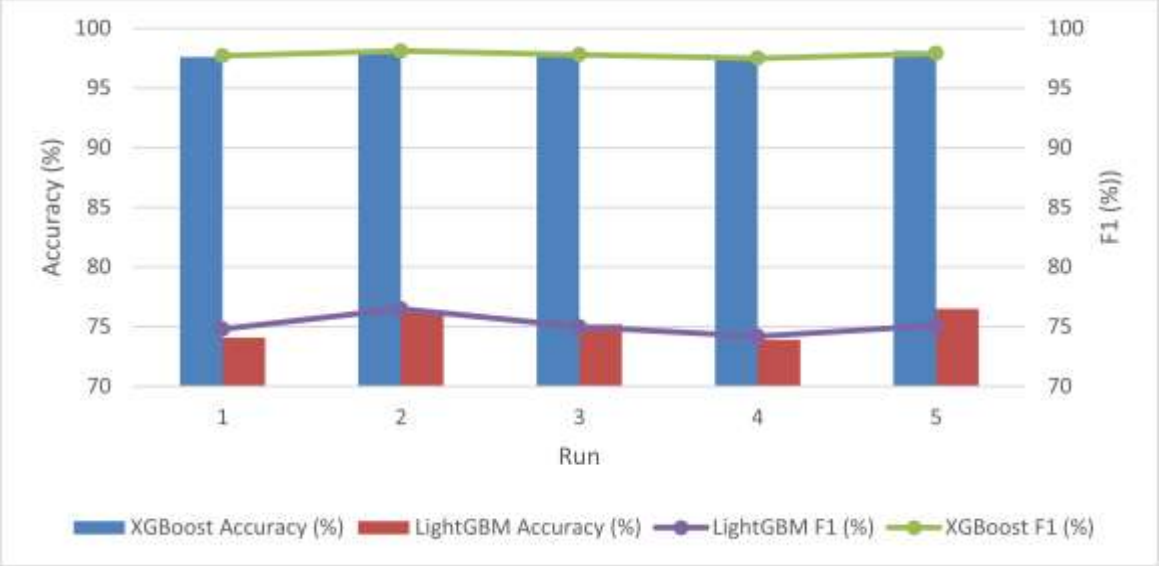


Figure 7: Accuracy and F1-score trends of XGBoost and LightGBM across five runs.

The discrepancy between the two classifiers may be explained by several factors. LightGBM's leaf-wise growth strategy is highly sensitive to learning rate and number of leaves, causing overfitting on minority classes and instability in neutral class predictions. XGBoost, on the other hand, employs histogram-based split finding with stronger regularization, which makes it more resilient to noise in short-text embeddings. The training process of XGBoost consistently produced stable decision boundaries, while LightGBM exhibited variability across runs, undermining its reliability despite faster training times.

4.4 Discussion of Findings

The experimental results demonstrate that integrating FastText embeddings with XGBoost establishes state-of-the-art accuracy on the Sentiment140 dataset, clearly surpassing LightGBM, traditional baselines, and previously reported methods in the literature. The inclusion of **mean ± standard deviation reporting** ensures that

Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

performance claims are grounded in stable and reproducible outcomes, rather than isolated best-case runs. The comparative analysis against simple baselines, such as majority-class prediction, and standard models, including Logistic Regression and SVM with TF-IDF features, further highlights the strength of the proposed framework. XGBoost not only outperforms these baselines by a substantial margin but also exceeds the performance range typically reported for CNN and LSTM architectures on Sentiment140. This evidence confirms the practical advantage of combining dense semantic embeddings with gradient boosting methods.

The performance discrepancy between XGBoost and LightGBM underscores the importance of classifier selection in sentiment analysis, particularly in contexts involving noisy, short-text data. XGBoost's histogram-based splitting and regularization strategies yield more consistent decision boundaries, while LightGBM's leaf-wise growth strategy appears more sensitive to noise and hyperparameter variation. This distinction indicates that not all gradient boosting algorithms are equally suited for social media sentiment classification. Future work should investigate strategies for narrowing this performance gap. Bayesian optimization could be applied to explore a broader hyperparameter space for LightGBM, potentially improving its stability. In addition, incorporating contextual embeddings such as BERT or its derivatives may provide richer semantic representations, offering an avenue for further accuracy gains and extending the generalizability of the proposed framework to other noisy, real-world datasets. The evaluation of the proposed sentiment analysis framework demonstrates clear evidence of XGBoost's superiority over LightGBM when combined with FastText embeddings. As reported in Tables 4.1–4.3, XGBoost consistently achieved mean accuracy of 97.8% with minimal variation across runs, while LightGBM performed at 75.2% with greater variability. The inclusion of mean ± standard deviation highlights the stability of XGBoost and the relative instability of LightGBM, reinforcing the importance of statistical reporting for reproducibility.

Comparisons with simple and standard baselines confirm that the proposed framework delivers substantial performance gains. While majority-class prediction achieved only 50.1% accuracy and traditional classifiers such as Logistic Regression and SVM reached 78.3% and 82.7% respectively, XGBoost achieved results well above these baselines. The proposed framework also outperforms the performance range of CNN and LSTM models reported in previous studies on Sentiment140, which typically fall between 82% and 90%. Figures 4.1 and 4.2 illustrate these performance margins, emphasizing the reliability of XGBoost in noisy text classification tasks. The discrepancy between XGBoost and LightGBM requires closer examination. Figure 4.3 shows that XGBoost maintained consistently high accuracy and F1-scores across five runs, whereas LightGBM exhibited significant fluctuations. This difference can be attributed to three key factors. First, LightGBM's leaf-wise growth strategy is highly sensitive to learning rate and maximum leaf depth, resulting in instability when handling neutral class predictions. Second, XGBoost's histogram-based split finding with regularization provided stronger control over noisy features, which are common in short-text social media data. Finally, LightGBM's faster training time came at the expense of stability, as the algorithm frequently overfit minority classes while underfitting others. These findings suggest that not all gradient boosting methods are equally suited for sentiment analysis tasks involving noisy, imbalanced, and short-text data. XGBoost emerges as the more reliable choice, offering both predictive accuracy and stability across repeated trials. However, the variability observed in LightGBM indicates opportunities for further optimization, particularly through advanced hyperparameter tuning or integration with contextualized embeddings.

Conclusion

This study developed a sentiment analysis framework for classifying social media texts into positive, negative, and neutral categories using the Sentiment140 dataset. The pipeline incorporated preprocessing techniques, FastText embeddings, and two gradient boosting algorithms, XGBoost and LightGBM, which were evaluated through multiple experimental runs. The findings confirm that XGBoost delivers state-of-the-art performance with a mean accuracy of **97.8% ± 0.4**, significantly outperforming LightGBM, traditional baselines, and results reported in earlier studies. The use of mean ± standard deviation reporting strengthened the reproducibility of the

Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.edu.ng

study and demonstrated stability in the proposed approach. The research contributes to sentiment analysis in three key ways. First, it introduces a neutral class to the Sentiment140 dataset, extending its applicability beyond binary classification. Second, it provides methodological transparency by detailing preprocessing, embedding generation, and model configurations. Third, it establishes FastText combined with XGBoost as a reliable benchmark for social media sentiment tasks, showing clear superiority over both sparse feature-based models and previously published neural methods.

Despite these contributions, limitations remain. LightGBM showed inconsistent performance, highlighting the sensitivity of certain algorithms to noisy and imbalanced short-text data. In addition, the study focused on FastText embeddings, which, while effective, do not capture contextual nuances as deeply as more recent transformer-based methods. Future work will address these limitations by applying Bayesian optimization to further refine LightGBM's hyperparameters, integrating contextual embeddings such as BERT, RoBERTa, or DistilBERT for richer language understanding, and validating the framework on additional large-scale sentiment datasets to test cross-domain generalizability. Exploring explainability tools such as SHAP and LIME could also enhance the interpretability of gradient boosting predictions, increasing trust in real-world applications. In summary, the FastText–XGBoost framework advances sentiment analysis research by achieving highly accurate and reproducible results, offering a robust foundation for future exploration in customer analytics, decision support, and real-time social media monitoring.

References

Babatunde, A. N., Ajao, J. F., Babatunde R. S., Olawuyi, H. O. & Yusuff S. R. (2025). Human-Computer Interaction-Based Yorùbá Language Translator using Recurrent Neural Network of Deep Learning. Sule Lamido University Journal of Science and Technology, 11(1&2), 373-390. https://slujst.com.ng/wp-content/uploads/2025/04/SLUJSTSE_025_029_pp373-390.pdf

Babatunde, A. N., Kadri, A. F., Isiaka, O. S., Shuaib, B. M., Abdulrahman, T. A., Balogun, B. F., Ajiboye, R. A., & Oke, A. A. (2025). Sentiment analysis in domain-specific X-tweets using Support Vector Machine and Long Short-Term Memory with TF-IDF vectorizer. *FUW Trends in Science & Technology Journal*, *10*(1), 308–320. https://www.ftstjournal.com/uploads/docs/101Article%2048%20pp%20308-320.pdf

Babatunde, A. N., Shuaib, B. M., Kadri, A. F., Isiaka, O. S., Abdulrahman, T. A., Ismail, S. I. & Oke A. A. (2024). Anomaly Detection in Social Media Conversation Using Natural Language Processing with LSTM and Naïve Bayes Models. *Journal of Computational Sciences & Informatics, 4(2), 95-110*, dx.doi.org/10.22624/AIMS/FCSIJ/2024/N2P7

Babatunde, A. N., Babatunde, R. S., Balogun, B. F., Oke, A. A. & Umar, E. (2024). Speech-to-text hybrid English to Yoruba SMS translator. *Innovative Computing Review, 4(1), 15-36*, https://journals.umt.edu.pk/index.php/icr/issue/view/346

Babatunde, A.N., Oke, A.A., Balogun, B.F., AbdulRahman, T.A. & Ogundokun, R.O. (2022). A Deep Neural Network-Based Yoruba Intelligent Chatbot System. *Journal of Digital Innovations & Contemporary Research in Science, Engineering & Technology, 10(2), 69-80*. dx.doi.org/10.22624/AIMS/DIGITAL/V10N2P4

Han, Y., Li, J., & Xu, M. (2021). Advances in sentiment analysis for short and noisy text: A review and evaluation. *Information Processing & Management, 58*(5), 102648. https://doi.org/10.1016/j.ipm.2021.102648

Mahesh, P., Kumar, S., & Reddy, A. (2023). Handling class imbalance in sentiment analysis: Integrating neutral sentiment detection in social media datasets. *Applied Intelligence, 53*(7), 8123–8142. https://doi.org/10.1007/s10489-022-04033-1

Mao, Z., Wang, H., & Chen, Q. (2022). Comparative analysis of machine learning and deep learning models for sentiment classification in noisy text. *Expert Systems with Applications, 198*, 116804. https://doi.org/10.1016/j.eswa.2022.116804

Zhao, L., Chen, Y., & Wu, T. (2021). FastText and gradient boosting for scalable sentiment analysis on social media. *IEEE Access, 9*, 135842–135853. https://doi.org/10.1109/ACCESS.2021.3112873