Kwaghe International Journal of
Engineering and Information Technology

ISSN : 1595-8035

Index: Harvard, Boston, Sydney University, Dimensions, Lens, ResearchGet Scilit, Semantic, Google Scholar, Base etc

https://doi.org/10.58578/KIJEIT.v2i2.5485

Tiktok Through AI Eyes: A Deep Learning Approach to Sentiment Analysis

Hambali Moshood Abiola[1*], Ayo Iyanuoluwa[2], Akinyemi Adesina A.[3], Adamu Muhammed Gadafi[4], Ashraf Ishaq[5]
Federal University Wukari, Taraba State, Nigeria
hambali@fuwukari.edu.ng; iseiyanu04@gmail.com

Article Info:

Submitted:	Revised:	Accepted:	Published:
Mar 23, 2025	Apr 7, 2025	Apr 19, 2025	Apr 24, 2025

Abstract

Background: The rapid growth of social media has transformed communication, with TikTok standing out among younger users for its short-form videos. Understanding user sentiment on these platforms is key to analyzing public opinion, trends, and engagement. **Aim:** This study explores sentiment analysis of TikTok user reviews using deep learning approaches, specifically Recurrent Neural Networks with Long Short-Term Memory (RNN-LSTM) and Deep Belief Networks (DBN). With over 144,000 reviews collected from Google Play and Apple App stores, the dataset was preprocessed using techniques such as lemmatization, tokenization, and GloVe word embeddings. The reviews were then classified into positive and negative sentiments. Both models were trained and evaluated based on metrics including accuracy, precision, recall, F1-score, and ROC-AUC. **Result:** Experimental results revealed that the RNN-LSTM model outperformed the DBN, achieving an accuracy of 81.99% and an AUC of 0.8874, compared to DBN's 78.53% accuracy and 0.8577 AUC. The findings demonstrate the effectiveness of deep learning—particularly LSTM—in capturing sentiment from noisy, user-generated content on platforms like TikTok. This work contributes to the growing field of AI-driven sentiment analysis and provides a foundation for future improvements through hybrid or multimodal approaches.

Keywords: LSTM, DBN, TikTok review, Sentiment analysis, Deep Learning, NLP

INTRODUCTION

The explosive expansion of social media platforms has profoundly reshaped how individuals interact and disseminate information. Among these platforms, TikTok has gained substantial traction, particularly among younger audiences, due to its engaging short-form video format. Gaining insights into user sentiment on such platforms is crucial for understanding public opinion, emerging trends, and user engagement patterns. Sentiment analysis, a key area within natural language processing (NLP), aims to detect and classify opinions embedded in textual data (Ashoshi & Hambali, 2024). It has become integral to various domains such as market research, consumer feedback analysis, and social media monitoring. Traditional sentiment analysis approaches—often based on lexicons and conventional machine learning methods—frequently struggle to interpret the complex and informal expressions common on social media (Medhat et al., 2014).

TikTok's distinctive blend of visual media and user-generated comments introduces both complexities and opportunities for sentiment analysis. Research indicates that combining textual and non-textual cues can enhance sentiment detection on multimedia platforms (Agu, Bako, & Hambali, 2023; Ashoshi & & Hambali, 2024). Furthermore, the platform's dynamic and fast-evolving content ecosystem demands sentiment analysis approaches that are both scalable and efficient. TikTok's distinct features present notable challenges for accurately identifying and categorizing user sentiment. The platform's ever-changing, short-form video content and the associated user comments often exceed the capabilities of traditional methods. In contrast, deep learning has significantly advanced sentiment analysis by enabling more effective modelling of intricate language patterns. Techniques like Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and, more recently, Transformer-based architectures such as BERT (Bidirectional Encoder Representations from Transformers) have shown marked improvements in sentiment classification tasks (Asoshi, Hambali, & Chukwudi, 2022;

Devlin et al., 2019). These models benefit from pre-training on extensive text datasets and subsequent fine-tuning on domain-specific corpora to deliver high performance. While deep learning approaches such as Recurrent Neural Networks (RNNs) and Transformer-based models have proven effective in comparable contexts (Devlin et al., 2019), the use of Deep Belief Networks (DBNs) in this specific area has received limited attention.

DBNs offer unique strengths, including the ability to learn complex hierarchical data structures and to extract meaningful features in an unsupervised manner (Bengio, 2009). These qualities suggest that DBNs could be particularly advantageous for sentiment analysis, especially in cases where understanding the interplay between visual content and textual feedback is essential. Nevertheless, the effectiveness of DBNs in analysing and interpreting sentiment within TikTok's environment remains largely unexplored. In this work, RNN and DBN algorithms are used to classify sentiment for TikTok reviews and then compared.

Related Works

This section briefly reviews related works published in the TikTok sentiment analysis domain. Li, Yang, & Gao (2020) aimed to analyse public sentiment toward viral content by classifying the sentiment polarity of TikTok comments. They applied traditional natural language processing methods, utilising Term Frequency–Inverse Document Frequency (TF-IDF) for feature extraction combined with Support Vector Machine (SVM) classifiers. Their approach resulted in a sentiment classification accuracy of 78%.

Al-Rubaie, Abdulkareem, & Al-Mashaqbeh (2021) set out to create a framework for identifying hate speech and toxic behaviour using sentiment analysis techniques. They implemented a hybrid deep learning approach that integrated Convolutional Neural Networks (CNN) with Bidirectional Long Short-Term Memory (BiLSTM). Their model demonstrated enhanced effectiveness in detecting toxic sentiment, achieving an F1-score of 0.84.

Chen, & Zhu (2022) explored the use of sentiment analysis to gain insights into user engagement and marketing impact on TikTok. They utilised BERT for contextual embedding of TikTok comments to analyse sentiment. Their findings revealed a direct correlation between sentiment polarity and user engagement metrics such as likes, shares, and comments.

In the study of Wahyudi & Sibaroni (2022), they proposed an Aspect-Based Sentiment Analysis (ABSA) approach using TikTok app reviews from the Google Play Store. Despite TikTok having 65.2 million active users globally, including 8.5 million from Indonesia, research utilising datasets from the TikTok app remains limited. This study focuses on classifying sentiment based on specific aspects: features, business, and content. The researchers employed a deep learning model—Recurrent Neural Network with Long Short-Term Memory (RNN-LSTM)—enhanced with BERT word embeddings. Their findings revealed that sentiment classification for the business aspect achieved the highest accuracy at 0.94, followed by the content aspect at 0.91, while the features aspect had the lowest accuracy at 0.85.

Aufa et al. (2023) conducted a study comparing three feature weighting techniques—Term Frequency–Inverse Document Frequency (TF-IDF), Term Frequency–Relevance Frequency (TF-RF), and Word2Vec—using an RNN classifier to analyse TikTok app reviews. The results indicated that TF-RF outperformed the other methods, achieving an accuracy of 87.6%, compared to 86% for TF-IDF and 80% for Word2Vec. This study contributes to the field by examining how different feature weighting approaches can improve sentiment analysis performance, offering insights that can support better decision-making.

Isnan et al. (2023) conducted sentiment analysis on TikTok app reviews from the Google Play Store. They used the VADER (Valence Aware Dictionary for sEntiment Reasoning) method to initially categorise reviews as positive, neutral, or negative. To manage class imbalance in the dataset, they applied Random Under-sampling (RUS) and Random Over-sampling (ROS). The data underwent preprocessing steps such as case folding, noise removal, normalisation, and stopword elimination. A Support Vector Machine (SVM) model was then trained for sentiment classification. Their findings showed that the SVM model performed best without any resampling, achieving an F1-score of 0.80. However, the study was limited by its exclusive focus on Google Play Store reviews and by the imbalance in the dataset, which had a predominance of positive reviews, complicating the analysis.

Majid, Nugraha, & Adhinata (2023) examined user reviews of the TikTok app to offer suggestions for improvement based on user feedback. They collected 1,000 reviews, which were manually labelled as positive or negative by five individuals. Two datasets were created for analysis—one with Natural Language Processing (NLP) preprocessing and one without. The preprocessing included stemming and stopword removal. The study compared the performance of both datasets to evaluate the impact of NLP techniques. The dataset that underwent NLP preprocessing achieved an accuracy of 76.92%, with a precision of 80.00% and recall of 74.07%. In contrast, the dataset without preprocessing performed worse, underscoring the benefits of NLP in sentiment analysis. However, the study's findings were limited by the small sample size and the potential for subjective bias in manual labelling.

Setiawan, & Pramudito (2023) focused on automatically classifying Indonesian TikTok reviews into negative, neutral, and positive sentiments using machine learning techniques. They employed two models: Long Short-Term Memory (LSTM) and IndoBERTweet, a variant of BERT optimised for Indonesian vocabulary using Twitter data. The models were trained on TikTok reviews collected from the Google Play Store. The results showed that IndoBERTweet outperformed LSTM, achieving an accuracy of 80%, while LSTM reached 78%. This suggests that transformer-based models like IndoBERTweet are more effective for sentiment analysis in this context. However, their study was limited to Indonesian-language reviews, which may restrict the applicability of the results to other languages or regions.

Singh, Patel, & Chang (2024) sought to examine cross-cultural sentiment trends to better understand how content is received in different regions. They used multilingual BERT (mBERT) for sentiment analysis across various languages, utilising annotated datasets gathered through TikTok's public API and web scraping methods. Their findings highlighted that cultural variations in sentiment significantly influence both content virality and brand perception. Notably, they observed a stronger positivity bias among Western TikTok users compared to those in Asian regions.

Suhaimi, & Lestari (2024) carried out a comprehensive sentiment analysis of TikTok user reviews in Indonesia, comparing the effectiveness of three machine learning models: Random Forest, Support Vector Machine (SVM), and Naive Bayes. They gathered 10,000 reviews from the Google Play Store through web scraping. The preprocessing process involved case folding, tokenisation, normalisation, stopword removal, and stemming. To assess model performance, they tested two dataset splits—80/20 and 70/30. Among the models, Random Forest achieved the highest accuracy, outperforming both

SVM and Naive Bayes. The study also highlighted that changes in the training-to-testing data ratio influenced the accuracy and consistency of the models. However, their analysis was limited to Google Play Store reviews, potentially overlooking sentiments from other platforms or user groups.

Thakur et al. (2024) research was to develop a dataset of videos related to the ongoing measles outbreak, shared on platforms such as YouTube, TikTok, and other similar sites. The dataset consists of 4,011 videos posted on 264 websites between January 1 and May 31, 2024, with YouTube and TikTok accounting for 48.6% and 15.2% of the videos, respectively. After compiling the dataset, the researchers performed sentiment analysis using VADER, subjectivity analysis with TextBlob, and fine-grained sentiment analysis with the DistilRoBERTa-base model. They classified each video title and description according to three factors: (i) overall sentiment—positive, negative, or neutral; (ii) subjectivity—highly opinionated, neutral opinionated, or least opinionated; and (iii) fine-grained sentiment—fear, surprise, joy, sadness, anger, disgust, or neutral. These results were incorporated as separate attributes in the dataset, which can be used for training and evaluating machine learning models in sentiment and subjectivity analysis, as well as in other related applications.

METHODOLOGY

The methodology employed in this study utilises a deep learning framework and compares their performance. The deep learning employed in this study includes Deep Belief Network (DBN) and Recurrent Neural Network (RNN), specifically the Long Short-Term Memory (LSTM) model, to perform sentiment analysis on TikTok reviews. The process is carried out through several stages, including data collection, preprocessing, data augmentation, feature extraction, sentiment classification, and model evaluation. Figure 1 depicts proposed framework for this study.

Dataset Description

The dataset used in this study consists of TikTok user reviews sourced from platforms such as the Google Play Store and the Apple App Store. This dataset is publicly available at Kaggle repository (https://www.kaggle.com/datasets/shivkumarganesh/tiktok-google-play-store-review). It comprises a total of 144,773 comments, which are utilised for both training and testing purposes. Prior to analysis, the data undergoes a cleaning process to ensure quality and consistency. The distribution of ratings within the dataset is as follows: 66,471 reviews with 5 stars, 35,713 with 1 star, 16,928 with 4 stars, 14,292 with 3 stars, and 11,369 with 2 stars. For sentiment labelling, ratings of 1 and 2 are categorised as negative, while ratings of 3, 4, and 5 are classified as positive.

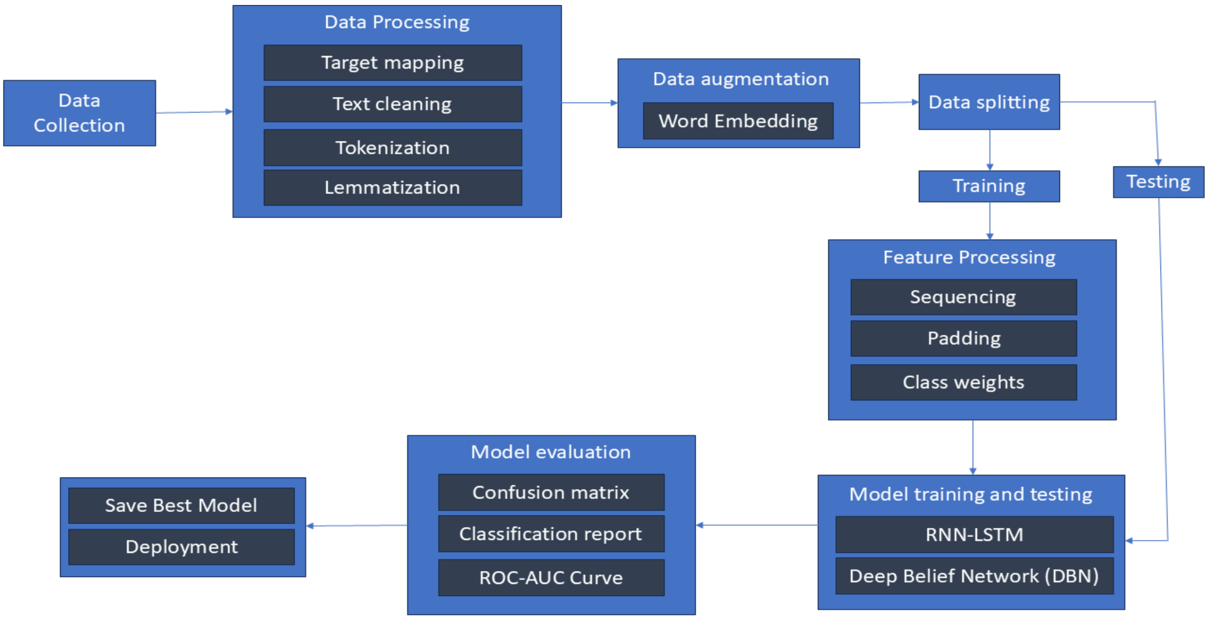

Figure 1: Proposed Framework

Data Preprocessing

To enhance the model's performance, several preprocessing techniques were implemented to clean and organise the dataset. These steps involved eliminating special characters and stopwords, applying lemmatisation, performing tokenisation, and padding the sequences to ensure a consistent input length suitable for deep learning models.

Target variable mapping

This technique involves converting categorical labels into a format better suited for classification tasks. In this study, the original sentiment ratings (ranging from 1 to 5) were reclassified into two binary categories: ratings of 1 and 2 were labelled as 0 (negative sentiment), while ratings of 3, 4, and 5 were labelled as 1 (positive sentiment). This transformation simplifies the classification task by reducing it to a binary problem, enabling the deep learning model to concentrate on distinguishing between positive and negative sentiments. As a result, the model can be trained more effectively, enhancing its ability to recognise sentiment patterns within TikTok reviews.

Text Cleaning Function

In Natural Language Processing (NLP), text cleaning is a crucial preprocessing step that transforms raw text data into a structured, noise-free format suitable for machine learning models. The clean_text() function used in this study processes TikTok reviews by systematically removing unnecessary elements while retaining keywords for sentiment analysis. The function performs several essential tasks, beginning with converting all text to lowercase to ensure consistency. It then removes URLs and HTML tags, discarding irrelevant content that doesn't contribute to sentiment analysis. The function also retains only alphabetic characters and spaces, removing numbers, punctuation, and special symbols that may add noise. Additional steps include eliminating extra spaces for uniformity and applying tokenisation to split sentences into individual words. Finally, it removes stopwords like "the," "is," and "and," which hold little value in sentiment analysis. These cleaning steps improve the quality of the input text, enabling the deep learning model to focus on meaningful linguistic features and enhance sentiment classification accuracy.

Tokenization

Tokenisation is the process of breaking down a sentence or phrase into individual words or subwords, simplifying text processing for machine learning models. In this study, the word_tokenize() function from NLTK was employed to split TikTok reviews into tokens, allowing each word to be analysed separately.

Lemmatization

Lemmatisation converts words to their base or root form, treating different variations of a word as a single entity. For instance, "running," "ran," and "runs" are all reduced to the root form "run." This process is especially beneficial for sentiment analysis, as it removes redundant variations and maintains consistency in word representation. In this study, the WordNet Lemmatizer from NLTK was used to standardise words before inputting them into the deep learning model.

GloVe Embeddings

The model utilised GloVe 25M 100d embeddings, a pre-trained word vector model that generates 100-dimensional vectors. This approach helped capture the semantic relationships between words, thereby enhancing the accuracy of sentiment classification.

Feature Processing

Sequencing

To enable deep learning models to process text, words must be transformed into numerical sequences. Tokenisation was carried out using the Keras Tokenizer, and each review was converted into a fixed-length sequence through padding, ensuring compatibility with the LSTM-RNN and DBN models.

Padding

Padding is an essential preprocessing step in NLP that ensures all input sequences are of uniform length before being processed by a deep learning model. Since text reviews vary in length, padding standardises the sequences by truncating longer texts or adding zeros to shorter ones to match a fixed length. In this study, the pad_X() function was used to ensure all tokenised sequences had a consistent length of 58 while preserving their 100-dimensional word embeddings. By adding a zero array, the function effectively pads shorter sequences and truncates longer ones to prevent shape mismatches. This step is crucial for batch processing in RNN-LSTM networks, enabling efficient input processing without distortion.

Class Weights

Class weights are applied in imbalanced datasets to prevent the model from favoring the majority class during training. In this study, the class weights were determined based on the frequency distribution of sentiment labels, yielding Class 0 (negative): 3.078 and Class 1 (positive): 1.481. These values reflect that the dataset has more positive reviews than negative ones, so the model needs to place greater emphasis on the minority class (negative reviews) to avoid bias. The calculated weights were incorporated during training to adjust the loss function, ensuring balanced learning across both sentiment classes. This approach enhances recall for the underrepresented class, resulting in a more accurate and unbiased sentiment classification model.

Classification Algorithm

RNN-LSTM

The LSTM model is a variant of Recurrent Neural Networks (RNNs) specifically designed to process sequential data such as text. Unlike conventional RNNs, LSTMs (Figure 2) include memory cells that capture long-term dependencies, effectively addressing vanishing gradient problems. In this study, the LSTM model is composed of embedding layers, LSTM layers, fully connected layers, and a softmax activation function for sentiment classification.

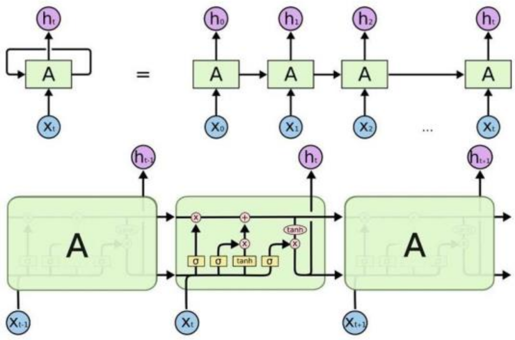

Figure 2: RNN-LSTM Architecture

Deep Belief Network

A Deep Belief Network (DBN) is a generative deep learning model made up of several layers of Restricted Boltzmann Machines (RBMs) that learn hierarchical feature representations from data. DBNs excel in unsupervised feature learning, identifying complex patterns and relationships in the input data before classification. Unlike traditional neural networks, DBNs use a layer-wise pre-training method, where each RBM is trained independently before fine-tuning the entire network. In NLP tasks like sentiment analysis, DBNs are effective at capturing meaningful text representations, which enhances the model's ability to distinguish between different sentiment classes. The strength of DBNs lies in their capacity to learn abstract, high-dimensional features, making them ideal for handling large-scale textual data. When combined with deep learning classifiers, DBNs improve model accuracy by providing well-organised feature representations, thus boosting sentiment classification performance. Figure 3 depicts DBN architecture.

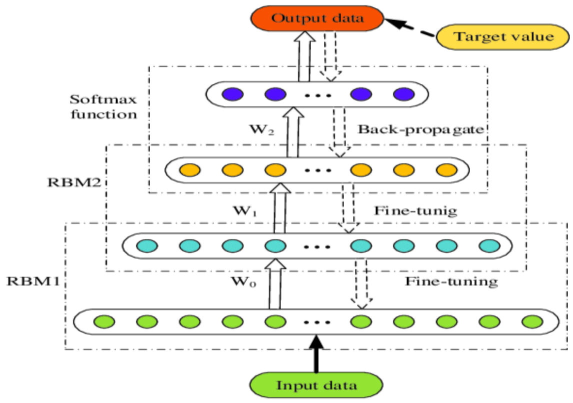

Figure 3: DBN Architecture

Performance Evaluation

1. Confusion Matrix – A Visualization of Correct and Incorrect Predictions: A confusion matrix is a tabular representation that provides a detailed breakdown of the model's classification performance by comparing actual and predicted sentiment labels. It helps to visualize the number of correct and incorrect predictions made by the model. A typical confusion matrix for binary classification (e.g., positive vs. negative sentiment) is structured as in Table 1:

***Table 1**: Confusion Matrix Interpretation*

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

- **True Positives (TP)**: Correctly classified positive sentiment reviews.
- **False Positives (FP)**: Negative sentiment reviews are incorrectly classified as positive.
- **True Negatives (TN):** Correctly classified negative sentiment reviews.
- **False Negatives (FN)**: Positive sentiment reviews are incorrectly classified as negative.

2. Precision: Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. It measures how many of the reviews that the model predicted as positive (or negative) were actually correct.

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

3. Recall (Sensitivity or True Positive Rate): Recall measures how well the model identifies actual positive cases. It calculates the proportion of correctly predicted positive reviews out of all actual positive reviews in the dataset.

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

4. Accuracy: Accuracy is the overall correctness of the model and is defined as the proportion of correctly classified reviews (both positive and negative) to the total number of reviews.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

5. F1-Score: The F1-Score is the harmonic mean of Precision and Recall, balancing both metrics in a single measure. It is useful when dealing with imbalanced datasets, ensuring that both precision and recall are considered.

$$F1\ Score = 2 * \frac{Precision * Recall}{Precison + Recall} \quad (4)$$

6. ROC-AUC Score – Evaluating the Model's Discriminative Ability

The Receiver Operating Characteristic (ROC) Curve and Area Under the Curve (AUC) score are used to measure how well the model distinguishes between positive and negative sentiment reviews.

- **ROC Curve:** A graph plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) at different classification thresholds.
- **AUC Score:** The area under the ROC curve, indicating the model's overall ability to separate positive and negative sentiment.

Mathematically, the True Positive Rate (TPR) and False Positive Rate (FPR) are calculated as follows:

$$TPR = \frac{TP}{TP+FN} \quad (5)$$

$$FPR = \frac{FP}{FP+TN} \quad (6)$$

The AUC score ranges from 0 to 1, where:

- **AUC = 1.0** → Perfect classification (ideal model).

- **AUC = 0.5** → Random guessing (poor model).
- **AUC < 0.5** → Worse than random guessing.

RESULTS AND DISCUSSION

This section presents the implementation details, environmental setup, and results of sentiment analysis on TikTok reviews using DBN and RNN-LSTM. Also, discussion on model training configurations and evaluation metrics, which include classification reports and the ROC AUC curve were presented.

Environmental Setup

The model was trained using Google Colab, a cloud-based Jupyter notebook environment that provides free access to GPUs and TPUs, improving training speed and efficiency. The system used for implementation was a Dell Latitude E7470 Intel(R) Core(TM) i5-6300U CPU @ 2.40GHz, 2501 Mhz, 2 Core(s), 4 Logical Processor(s) laptop for offline processing and debugging.

Parameter Setting

RNN-LSTM

The RNN-LSTM Parameter Setting table provides key hyperparameters for training an RNN with Long Short-Term Memory (LSTM) units. Here's a brief explanation of the parameters presented in Table 2:

- **Batch size (32)**: Determines the number of samples processed before updating the model's parameters.
- **Epoch (3)**: The number of complete passes through the training dataset.
- **Hidden Layer Activation Function (ReLU)**: The Rectified Linear Unit (ReLU) is used for non-linearity in hidden layers to improve learning efficiency.
- **Output Layer Activation Function (Sigmoid)**: Used to produce probabilities, typically for binary classification tasks.
- **Optimizer Learning Rate (Adam 0.001)**: Adam optimizer with a learning rate of 0.001 helps in efficient gradient descent optimization.

- **Dropout (0.2)**: A regularisation technique where 20% of neurons are randomly dropped during training to prevent overfitting.
- **Loss Function (Binary Cross-Entropy)**: A loss function suited for binary classification tasks, measuring the difference between predicted and actual values.

These settings help in optimizing the performance and generalisation of the LSTM model.

Table 2: RNN-LSTM Parameter setting

Parameters	**Value**
Batch size	32
Epoch	3
Hidden Layer activation function	ReLU
Output layer activation function	Sigmoid
Optimizer learning rate	Adam 0.001
Dropout	0.2
Loss function	Binary cross-entropy

DBN Parameter Setting

The DBN (Deep Belief Network) Parameter Setting table defines the key hyperparameters used for training a DBN model. Here's a brief explanation of each parameter presented in Table 3:

- **Batch size (32)**: The number of training samples processed in each iteration before updating the model weights.
- **Epoch (3)**: The total number of times the model iterates over the entire dataset during training.
- **Hidden Layer Activation Function (ReLU)**: The Rectified Linear Unit (ReLU) is used to introduce non-linearity, improving learning efficiency.
- **Output Layer Activation Function (Sigmoid)**: This function is used for binary classification tasks, converting outputs into probabilities.
- **Optimizer Learning Rate (Adam 0.001)**: The Adam optimizer with a learning rate of 0.001 helps in adaptive gradient-based optimization.
- **Dropout (0.2)**: A regularization technique where 20% of neurons are randomly dropped to prevent overfitting.

- **Loss Function (Binary Cross-Entropy)**: A loss function suitable for binary classification, measuring the difference between predicted and actual labels.

These parameter settings help in training the DBN model efficiently while minimising overfitting and improving generalisation.

Table 3: DBN Parameter Setting

Parameters	Value
Batch size	32
Epoch	3
Hidden Layer activation function	ReLU
Output layer activation function	Sigmoid
Optimizer learning rate	Adam 0.001
Dropout	0.2
Loss function	Binary cross-entropy

Model Architecture

The model consisted of an embedding layer, LSTM layers, dense layers, and a dropout layer to prevent overfitting. Table 4 and 4 summarise Architecture model of RNN-LSTM and DBN respectively.

Table 4: RNN-LSTM Model summary

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 58, 64)	42,240
dropout (Dropout)	(None, 58, 64)	0
lstm_1 (LSTM)	(None, 58, 128	98,816
dropout_1 (Dropout)	(None, 58, 128	0
lstm_2 (LSTM)	(None, 58, 64)	49,408
dropout_2 (Dropout)	(None, 58, 64)	0
flatten (Flatten)	(None, 3712)	0
dense (Dense)	(None, 1)	3,713

Table 5: DBN Model summary

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 58, 64)	42,240
dropout (Dropout)	(None, 58, 64)	0
lstm_1 (LSTM)	(None, 58, 128	98,816
dropout_1 (Dropout)	(None, 58, 128	0
lstm_2 (LSTM)	(None, 58, 64)	49,408
dropout_2 (Dropout)	(None, 58, 64)	0
flatten (Flatten)	(None, 3712)	0
dense (Dense)	(None, 1)	3,713

Percentage Split Technique

The dataset was split using an 80-10-10 ratio: 80% for training, 10% for validation, and the last 10% for testing.

Model Results

Confusion Matrix

The LSTM model demonstrated strong performance in sentiment classification, as observed in its confusion matrix (Figure 4). It correctly classified 5,671 Positive reviews (True Positive - TP) and 12,135 Negative reviews (True Negative - TN), indicating high reliability in distinguishing between sentiments. The False Positive (FP) count of 1,377 means that a small number of negative reviews were incorrectly classified as positive. Additionally, the False Negative (FN) count of 2,533 shows that some positive reviews were misclassified as negative. However, the relatively low FN count suggests that the model effectively captures positive sentiments, making it more suitable for analysing user reviews where accurate sentiment classification is essential.

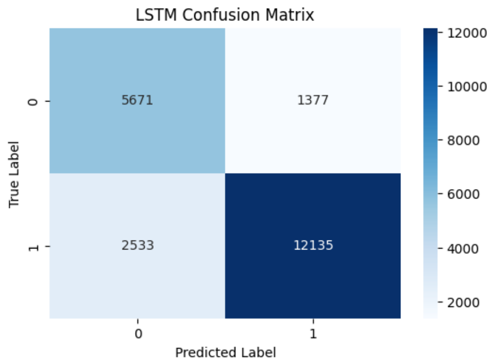

Figure 4: RNN-LSTM Confusion Matrix

The Deep Belief Network (DBN) model also performed well but had slightly lower accuracy compared to LSTM (Figure 5). It correctly classified 5,587 Positive reviews (TP) and 11,466 Negative reviews (TN), demonstrating good sentiment detection capabilities. However, its False Positive (FP) count of 1,461 was higher than that of LSTM, meaning it misclassified more negative reviews as positive. Additionally, 3,202 positive reviews were misclassified as negative (FN), indicating that DBN struggled more with identifying positive sentiments. The higher FN count suggests that DBN is less effective in capturing true positive reviews, making it more prone to underestimating user satisfaction in sentiment analysis.

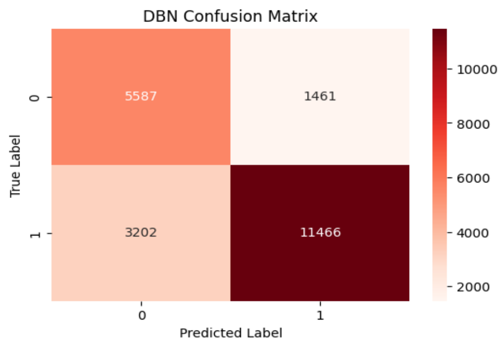

Figure 5: DBN Confusion Matrix

Classification Report

Table 6 presents classification results for both LSTM and DBN models.

Precision: Measures how many of the predicted positive and negative sentiments were actually correct. The LSTM model achieved a higher precision (83.09%) compared to DBN (80.54%), indicating that when LSTM classified a review as either positive or negative, it was more likely to be correct. Specifically, LSTM had a higher precision for negative sentiment (69.12%) than DBN (63.57%), meaning DBN had more false positives, incorrectly classifying negative reviews as positive. This suggests that LSTM is better at reducing misclassifications, which is particularly beneficial in applications where false positives must be minimised, such as brand monitoring or content moderation.

Recall: Evaluates how well the models identify actual positive and negative reviews, considering the proportion of correct predictions out of all actual occurrences. LSTM exhibited higher recall (81.99%) than DBN (78.53%), meaning it was more effective at capturing true sentiment, particularly for positive reviews (82.73% vs. 78.17%). This indicates that DBN struggled more with false negatives, missing actual positive or negative reviews. A higher recall in LSTM means it is more reliable for detecting sentiment patterns, making it better suited for applications where it is crucial to capture all relevant sentiment trends, such as customer feedback analysis.

Accuracy: Provides a general measure of how well the model classifies reviews correctly across both sentiment classes. LSTM attained 81.99% accuracy, outperforming DBN, which achieved 78.53%. This shows that LSTM made fewer overall classification errors. The superior accuracy of LSTM can be attributed to its ability to capture sequential dependencies in text, allowing it to better understand context. While DBN performed well, its lower accuracy suggests that it made more mistakes in distinguishing between positive and negative reviews, making LSTM the better choice for real-world sentiment analysis tasks requiring higher reliability.

F1-Score: F1-score is the harmonic mean of precision and recall, providing a balanced evaluation of model performance. LSTM achieved a higher weighted F1-score (82.30%) compared to DBN (79.03%), confirming its superior overall performance. The F1-score for negative sentiment in LSTM was 74.36% versus 70.56% in DBN, further demonstrating that LSTM is better at identifying sentiment in a balanced manner. Since F1-score considers both false positives and false negatives, LSTM proves to be a more robust model for sentiment classification, particularly in handling complex user-generated content like TikTok reviews.

Table 6: Classification Report

Metric	LSTM	DBN
Precision	83.09	80.54
Recall	81.99	78.53
F1-Score	82.30	79.03
Accuracy	81.99	78.53

ROC AUC Result

The Receiver Operating Characteristic - Area Under the Curve (ROC-AUC) is a crucial metric for evaluating the performance of a classification model by measuring its ability to distinguish between positive and negative sentiment classes. A higher AUC score indicates that the model is better at differentiating between sentiments, with 1.0 representing perfect classification and 0.5 indicating random guessing. In this study, the LSTM model achieved an AUC of 0.8874 (Figure 6), while the DBN model recorded an AUC of 0.8577, demonstrating that LSTM has a stronger ability to correctly classify sentiments. The LSTM validation loss (0.4140) and accuracy (81.65%) further confirm its reliability, outperforming DBN, which had a higher validation loss (0.4808) and lower accuracy (78.41%). The ROC curve, when plotted, shows that LSTM maintains a steeper rise towards the top-left corner of the graph compared to DBN, indicating fewer false positives and false negatives. These results suggest that LSTM is a more effective model for sentiment analysis on TikTok reviews, as it provides a better trade-off between sensitivity and specificity. However, DBN still performs well and could be improved by integrating it with LSTM in a hybrid model to enhance classification performance further.

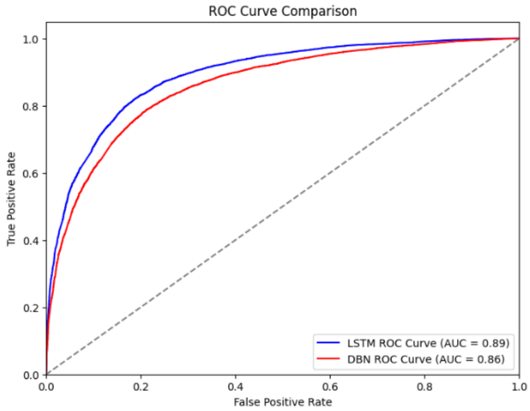

Figure 6: ROC AUC score curve

CONCLUSION

The study successfully demonstrated the effectiveness of deep learning models, particularly LSTM networks, in analyzing sentiment in TikTok user reviews. By leveraging advanced text preprocessing techniques and word embeddings, the model was able to accurately classify user sentiment with high reliability. The 83% accuracy obtained suggests that deep learning approaches can be effectively applied to sentiment analysis on social media platforms, where textual data is abundant but often noisy. Despite the promising results, challenges such as handling sarcasm, emojis, and multimodal content remain areas for further improvement. Overall, the project contributes to the growing field of sentiment analysis by providing a robust framework for extracting meaningful insights from TikTok user reviews.

REFERENCES

Agu, E. O., Bako, J. Z., & Hambali, M. A. (2023). Analyzing Election Sentiments in Tweets with Gated Recurrent Units (GRU). *Asian Journal of Research in Computer Science*, *16*(4), 125 – 132. https://doi.org/10.9734/ajrcos/2023/v16i4376

Al-Rubaie, M., Abdulkareem, S., & Al-Mashaqbeh, I. (2021). Deep learning-based sentiment and toxicity analysis of TikTok user comments. *Journal of Social Media Mining*, *9*(2), 134–149.

Ashoshi, P. A. & Hambali, M. A. (2024). Application of hybrid SVM-LR algorithm for sentiment analysis. In *Optimization and Computing using Intelligent Data-Driven Approaches for Decision-Making* (pp. 283–308). CRC Press, Taylor & Francis.

Asoshi, P. A., Hambali, M. A. & Chukwudi, J. I. (2022). Review of Sentiment Analysis: An

Hybrid Approach. *Middle East Journal of Applied Science & Technology*, *5*(4), 24–54.

Aufa, R. N., Prasetiyowati, S. S., & Sibaroni, Y. (2023). The Effect of Feature Weighting on Sentiment Analysis TikTok Application Using The RNN Classification. *Building of Informatics , Technology and Scienece (BITS)*, *5*(1), 345–353. https://doi.org/10.47065/bits.v5i1.3597

Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and Trends® in Machine Learning*, *2*(1), 1–127.

Chen, Y., & Zhu, T. (2022). Exploring user sentiment and engagement on TikTok using BERT-based models. *Social Network Analysis and Mining*, *12*(1), 45.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186.

Isnan, M., Natanael, G., & Pardamean, B. (2023). Sentiment Analysis for TikTok Review Using VADER Sentiment and SVM Model. *8th International Conference on Computer Science and Computational Intelligence (ICCSCI 2023)*, *227*, 168–175. https://doi.org/10.1016/j.procs.2023.10.514

Li, X., Yang, J., & Gao, H. (2020). Sentiment classification of TikTok comments using machine learning algorithms. *Proceedings of the 2020 International Conference on Artificial Intelligence and Big Data (ICAIBD)*, 79–84.

Majid, A., Nugraha, D., & Adhinata, F. D. (2023). Sentiment Analysis on TikTok Application Reviews Using Natural Language Processing Approach. *Journal of Embedded Systems, Security and Intelligent Systems*, *4*(1). https://doi.org/10.26858/jessi.v4i1.41897

Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, *5*(4), 1093–1113.

Setiawan, J. C., & Pramudito, A. (2023). Sentiment Analysis of Indonesian TikTok Review Using LSTM and IndoBERTweet Algorithm. *Jurnal Informatika Dan Perekayasaan Informasi*, *8*(3). https://doi.org/10.29100/jipi.v8i3.3911

Singh, R., Patel, S., & Chang, Y. (2024). Cross-cultural sentiment analysis on TikTok: A deep learning perspective. *IEEE Transactions on Affective Computing*. https://doi.org/10.1109/TAFFC.2024.3200012

Suhaimi, N. A., & Lestari, M. (2024). Sentiment Analysis of TikTok App Reviews on Google Play Using Several Machine Learning Methods. *International Journal of Global Operations Research*, *5*(4), 275–287.

Thakur, N., Su, V., Shao, M., & Patel, K. A. (2024). A Labelled Dataset for Sentiment Analysis of Videos on YouTube , TikTok , and Other Sources about the 2024 Outbreak of Measles. *26th International Confer- Ence on Human-Computer Interaction (HCII 2024), Washington, USA*, *July*, 1–19.

Wahyudi, D., & Sibaroni, Y. (2022). Deep Learning for Multi-Aspect Sentiment Analysis of TikTok App using the RNN-LSTM Method. *Building of Informatics , Technology and Scienece (BITS)*, *4*(1), 169–177. https://doi.org/10.47065/bits.v4i1.1665