Sentiment Classification on IMDb Movie Reviews: A Controlled Comparative Study of Classical, Deep-learning, Feature-fusion, and Fine-tuned Transformer Models

Santosh Kumar Banbhrani [1,2,3,*], Wang Shaojie [1,*], Liu Baiyang [1], Chen Wei [1], Pir Dino Soomro [4], Ali Raza Rang [2,3], Dhani Bux Talpur [2,3], and Asad Ullah Shah [3]

[1] School of Electrical Engineering, Shaoyang University, Shaoyang, China
[2] Department of Information and Computing, University of Sufism and Modern Science, Bhitshah, Matiari, Sindh, Pakistan
[3] Kulliyyah of Information and Communication Technology, International Islamic University Malaysia, Jalan Gombak, Kuala Lumpur, Selangor, Malaysia
[4] School of Information Science and Technology, Dalian Maritime University, Dalian, China
Email: santosh.kumar@hnsyu.edu.cn and banbhrani@gmail.com (S.K.B.); shaojiew@hnsyu.edu.cn (W.S.); 806717285@qq.com (L.B.); 2569@hnsyu.edu.cn (C.W.); pirdinosoomro@dlmu.edu.cn (P.D.S.); alirazarang@gmail.com (A.R.R.); dhanibux@hotmail.com (D.B.T.); asadullah@iium.edu.my (A.U.S.)
*Corresponding author

***Abstract*—Sentiment analysis is one of the most important tasks in natural language processing, as it allows opinions expressed in text to be automatically categorized. This study presents a systematically controlled comparative analysis of classical machine-learning, deep-learning, feature-fusion, and fully fine-tuned transformer models for binary sentiment classification on the Internet Movie Database (IMDb) Movie Reviews dataset. While the individual components of the framework, including Term Frequency-Inverse Document Frequency (TF-IDF), Bidirectional Encoder Representations from Transformers (BERT) embeddings, feature fusion, and Multi-Layer Perceptron (MLP) classification, are established techniques, this study contributes a comprehensive empirical evaluation of their combined effectiveness for the target classification task, providing insights into the practical benefits of integrating lexical and contextual representations. Under a single experimental framework, all models are evaluated on a commonly held-out test set with 95% bootstrap confidence intervals, paired McNemar significance testing, confusion-matrix and error analysis, and a computational-complexity comparison. The fully fine-tuned transformers achieve the highest performance Robustly Optimized BERT Pretraining Approach (RoBERTa), F1-Score = 0.9427. The TF-IDF and BERT feature-fusion model is significantly more accurate than the classical, deep-learning, and BERT-only baselines (all $p < 0.001$), but significantly less accurate than the fine-tuned transformers (all $p < 0.01$); furthermore, concatenating the [CLS] token and mean-pooled representations does not significantly improve upon mean pooling alone ($p = 0.48$). The study provides a transparent, evidence-based account of the accuracy and cost trade-offs among these model families.**

***Keywords*—sentiment analysis, Internet Movie Database (IMDb) dataset, machine learning, deep learning, hybrid models, Term Frequency-Inverse Document Frequency (TF-IDF), Bidirectional Encoder Representations from Transformers (BERT)**

I. Introduction

Sentiment analysis is a primary task in Natural Language Processing (NLP) that attempts to automatically recognize and categorize subjective data represented in textual data. User-generated content, such as the growing number of movie review systems, e-commerce sites, and social media, has been subject to exponential growth, and sentiment analysis in opinion mining, recommender systems, and decision making is gaining importance with each passing day. Usually, Internet Movie Database (IMDb) Movie Reviews are used as a benchmark dataset because of their size, class balance, and long-form text structure that creates realistic challenges such as contextual dependency, sentiment changes, negation, and sarcasm [1–3].

Traditionally, sentiment classification is done by handcrafted lexical features like Bag-of-Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), etc., and machine learning models. It is demonstrated that TF-IDF upon linear classifiers such as Support Vector Machines (SVM) are very good and efficient baselines, especially in the use of long-form data such as IMDb. Likewise, Bisht *et al*. [4] proved that well-calibrated linear models have proved to be highly competitive even in the face of the emergence of more complex models. These techniques, however, cannot capture long-range

Manuscript received February 23, 2026; revised March 10, 2026; accepted July 10, 2026; published September 24, 2026.

sentiment dependencies, contextual and semantic relationships.

In order to overcome these drawbacks, deep learning models like Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks have been developed. CNNs are good at learning local n-gram patterns, and Bidirectional LSTM (BiLSTM) models are good at learning sequential dependency, and they can learn forward and backward. Guha and Mohan [5] note that long textual sequences like reviews of movies are especially well-suited to the BiLSTM models. In a similar way, Abdullah and Ahmet [6] and Wang *et al.* [7] reported that pretrained embeddings like GloVe enhance semantic representation. However, such models are based on fixed embeddings and consequently fail to capture all the dynamic contextual variations found in complex textual data.

Further progress has been made on transformer-based models, such as Bidirectional Encoder Representations from Transformers (BERT), that leverage self-attention mechanisms to add contextualized representations of words. Kokab *et al*. [8] suggest that BERT is a good tool that is able to capture the bidirectional semantic relationship, and it can model the nuance and polarity of a context. Correspondingly, González-Carvajal and Garrido-Merchán [9] also found that transformer-based models are superior to conventional architectures in a variety of sentiment classification tasks. Nevertheless, with these benefits, transformer models are computationally intensive and do not necessarily exhibit consistent improvements when applied alone. As Ipadeola *et al.* [10] and Elmitwalli and Mehegan [11] note, under some conditions, even the traditional TF-IDF-based models can achieve comparable performance. Further, Papa *et al.* [12] highlighted the computational trade-offs associated with transformer-based architectures.

As a result, recent research has examined the combination of lexical and contextual representations. Krouska *et al*. [13] and Aprilio *et al.* [14] report that the combination of TF-IDF features and BERT-based embeddings increases classification performance by taking advantage of complementary strengths of sparse lexical patterns and dense contextual semantics. Likewise, Permana *et al*. [15] have shown that such integration is effective in sentiment classification tasks, especially in low-resource settings. Furthermore, Hossain *et al.* [16] suggested a multi-branch architecture that integrates TF-IDF, deep learning modules, and transformer models, which has high performance, but at the expense of higher computational complexity. Similarly, Sun *et al*. [17] presented a hybrid framework based on ranking in which TF-IDF and BERT are used in a sequential manner, as opposed to a collaborative one.

Although these developments have been made, the existing literature is mainly differentiated by the feature extraction and fusion strategies, and is more often based on simplified representations, dimensionality reduction algorithms, or loosely coupled architectures. Specifically, a variety of methods use a single contextual representation (e.g., [CLS] (classification) token) or apply dimensionality reduction (e.g., TruncatedSVD) before fusion, which can cause loss of semantic richness [13, 14]. Moreover, most of the previous studies are limited to short-text datasets and constrained evaluation environments, which limits their generalizability to more complex long-form sentiment classification problems.

In contrast to studies that reduce representational power before fusion, the present work evaluates a direct feature-level integration of lexical and contextual representations. TF-IDF features are then combined with the BERT [CLS] token representation and the mean-pooled token embeddings, as the input to a Multi-Layer Perceptron (MLP) classifier. This study emphasizes that this design uses established components and is examined here as a representative feature-fusion approach rather than as a novel architecture.

Moreover, the proposed framework does not reduce the representational power of both TF-IDF and contextual embeddings, as opposed to the previous methods that reduce the dimensionality of features before integration. Direct feature-level concatenation is used to perform feature integration, and information is not lost during the process. Furthermore, an MLP classifier is used to allow the learning of a single decision boundary across non-homogeneous feature spaces, which is a more expressive modeling capability than independent or sequential fusion strategies.

Additionally, this study adopts a controlled and unified experimental framework, where machine learning, deep learning, and the proposed architecture are evaluated under identical preprocessing, data partitioning, and evaluation protocols. Diri *et al.* [2] and Bhargavi [18] state that such consistency is a prerequisite to reliable comparative analysis, which is often not the case in previous studies.

Accordingly, the value of this work lies not in a new modeling paradigm but in a controlled and transparent empirical comparison. The components of the feature-fusion model are established techniques, and we do not claim them as innovations. Instead, the study clarifies, with statistical testing, how lexical, deep-learning, feature-fusion, and fully fine-tuned transformer approaches compare on this task, and at what computational cost.

II. Methodology

A controlled comparative experimental design is adopted in this work to assess the effectiveness of classical machine-learning, deep-learning, feature-fusion, and fully fine-tuned transformer models for binary sentiment classification. The diagram of the proposed model is displayed in Fig 1. Each model is evaluated under identical preprocessing, data partitioning, and evaluation conditions, so that differences in performance can be attributed to the models rather than to the experimental setup. As noted above, the feature-fusion model combines established components and is examined as a representative approach rather than as a methodological innovation.

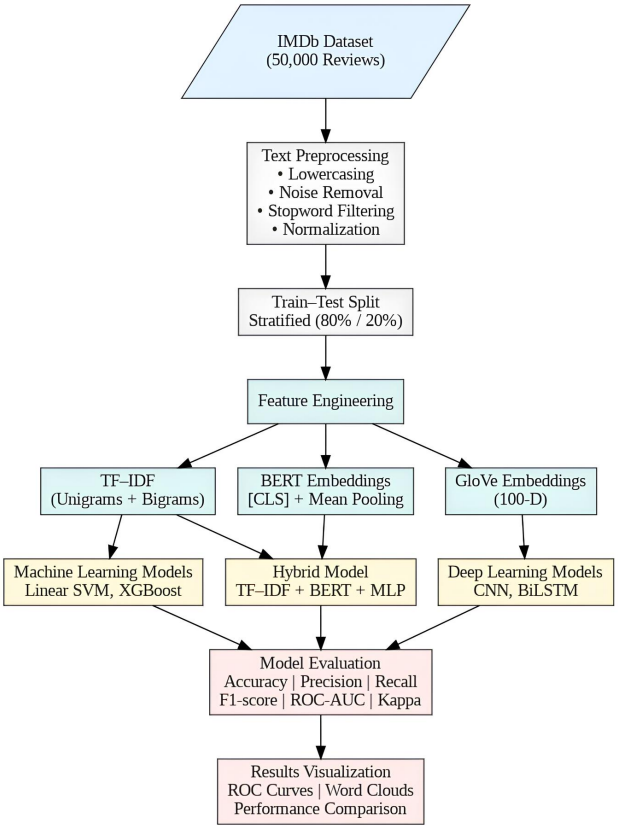

Fig. 1. Word clouds illustrating the most frequent terms in positive and negative IMDb movie reviews after preprocessing.

A. Dataset Description

The empirical test is conducted using the IMDb Movie Reviews dataset, which is also a popular benchmark when conducting sentiment analysis work on long form textual data. The dataset was 50,000 English-language movie reviews, which are evenly balanced between the positive and negative sentiment classes, hence, removing any issues of class imbalance. The binary sentiment label associated with each review is numerically encoded to be used as a model train and evaluation variable. The dataset has been downloaded through a publicly accessible repository on GitHub and was chosen as it has been previously used to compare the traditional and neural sentiment classification models [5, 19].

B. Experimental Protocol and Reproducibility

In order to provide an adequate comparability of all models, the same evaluation protocol is used:

- Training Phase: The training set is only used to train all the models.
- Hyperparameter Optimization: The hyperparameters are chosen by stratified k-fold cross-validation from the training partition to maximize F1-Score. Cross-validation is used solely for hyperparameter tuning and never for reporting; all final results, for every model without exception, are reported on the commonly held-out test set.
- Final Evaluation: All findings are only reported on the test set that is held-out.
- Statistical Stability Analysis: Bootstrap resampling (1000 resamples) of test-set predictions is used to estimate the performance variability to compute an average performance and 95% confidence intervals.

This coherent design guarantees that all the models are tested in the same conditions.

The same global random seed (42) is employed across all the experiments in Python, NumPy, PyTorch, TensorFlow, and scikit-learn to ensure the reproducibility of the experiments. Training and evaluation were done on a single NVIDIA Tesla T4 GPU. The entire software environment as automatically captured at run time, was: Python 3.12.13, PyTorch 2.11.0 (CUDA 12.8 build), Transformers 5.10.2, scikit-learn 1.6.1, NumPy 2.0.2, XGBoost 3.2.0, and TensorFlow 2.20.0. The MLP meta-classifier has two hidden layers (512 and 128 units), Rectified Linear Unit (ReLU) activations, dropout 0.30, Adam optimizer (learning rate 1×10^{-3}), batch size 64, and early stopping with patience 3. The fully fine-tuned transformers are trained with a learning rate of 2×10^{-5}, three epochs, a maximum sequence length of 256, and a batch size of 16.

C. Text Preprocessing

To reduce noise and ensure uniformity between models, all reviews are processed using a standard text processing format. These involve eliminating URLs, punctuations, special characters, and non-alphabetic characters and then converting all the text to lower case. Stopwords in English are deleted to minimize redundancy and enhance the quality of features, and superfluous whitespace is removed. Duplicates are eliminated and blank texts deleted.

Mathematically, for a raw review text x, the cleaned one $\tilde{x}$ is written: x = Normalize (Lowercase (Remove Punctuation (Remove Stop words (x)))).

This pre-processing solution is compatible with both machine learning and deep learning models [20, 21], and consistent with the existing sentiment analysis pipeline.

D. Feature Representations

1) TF-IDF vectorization

In traditional machine learning and hybrid frameworks, textual data is transformed into a numerical representation using TF-IDF. Local context dependencies in text are captured using both unigram and bigram representations.

The TF-IDF weight of a term t in document d is defined as:

$$TF - IDF(t,d) = TF(t,d) \times \log\left(\frac{N}{DF(t)}\right) \quad (1)$$

where $TF(t, d)$ is the frequency of term t in document d, $DF(t)$ is the number of documents that contain term t, and N is the total number of documents in the corpus. The top 5000 informative features are then used as the feature space to trade-off representational richness with computational efficiency, as previous studies based on IMDb have done [10, 19].

2) Word embeddings for deep learning models

Pre-trained GloVe embeddings with 100-dimensional features are used to initialize the embedding layer of a deep learning model. The embedding matrix E = RV × 100: each token is represented by a semantic dense vector; V: vocabulary size. The embeddings are not trained to maintain the original semantic structure that has been learned in large external corpora. To achieve equal input sizes to neural networks, which is common in sentiment classification tasks, each review is tokenized and padded to a constant sequence length [5].

E. Model Architectures

In this case, a BERT-base, RoBERTa-base, and DistilBERT were fully fine-tuned with the same protocol, and the final results are presented in Table I. RoBERTa achieves the best F1-Score (0.9427), followed by BERT (0.9260) and DistilBERT (0.9181). For completeness, three BERT-only models (based on the [CLS] vector, the mean-pooled vector, or a concatenation of the two) are also reported, without TF-IDF, in Table I.

TABLE I. PERFORMANCE COMPARISON OF MACHINE LEARNING, DEEP LEARNING, AND HYBRID MODELS ON THE IMDB MOVIE REVIEWS DATASET

Model	Accuracy (Mean ± CI)	Precision (Mean ± CI)	Recall (Mean ± CI)	F1-Score (Mean ± CI)	ROC-AUC (Mean ± CI)
RoBERTa-ft	0.9422 ± 0.0046	0.9371 ± 0.0065	0.9484 ± 0.0061	0.9427 ± 0.0047	0.9847 ± 0.0020
BERT-ft	0.9253 ± 0.0047	0.9209 ± 0.0069	0.9312 ± 0.0069	0.9260 ± 0.0049	0.9770 ± 0.0025
DistilBERT-ft	0.9173 ± 0.0054	0.9123 ± 0.0074	0.9240 ± 0.0071	0.9181 ± 0.0056	0.9720 ± 0.0028
Hybrid (CLS + mean)	0.9069 ± 0.0057	0.9051 ± 0.0078	0.9098 ± 0.0076	0.9074 ± 0.0060	0.9679 ± 0.0029
Hybrid (mean only)	0.9052 ± 0.0054	0.8906 ± 0.0080	0.9245 ± 0.0071	0.9072 ± 0.0056	0.9678 ± 0.0028
Hybrid (CLS only)	0.9014 ± 0.0058	0.9064 ± 0.0079	0.8960 ± 0.0084	0.9011 ± 0.0061	0.9655 ± 0.0031
TF-IDF + MLP	0.8888 ± 0.0058	0.8728 ± 0.0086	0.9110 ± 0.0076	0.8915 ± 0.0062	0.9601 ± 0.0033
Linear SVC	0.8845 ± 0.0063	0.8817 ± 0.0088	0.8891 ± 0.0086	0.8854 ± 0.0064	0.9564 ± 0.0034
BERT-only (mean) + MLP	0.8728 ± 0.0065	0.8555 ± 0.0093	0.8982 ± 0.0082	0.8763 ± 0.0067	0.9474 ± 0.0040
BiLSTM	0.8731 ± 0.0066	0.8764 ± 0.0088	0.8697 ± 0.0095	0.8730 ± 0.0071	0.9457 ± 0.0042
XGBoost	0.8698 ± 0.0067	0.8574 ± 0.0096	0.8881 ± 0.0086	0.8725 ± 0.0069	0.9469 ± 0.0039
BERT-only (CLS + mean) + MLP	0.8698 ± 0.0066	0.8606 ± 0.0093	0.8835 ± 0.0090	0.8719 ± 0.0067	0.9461 ± 0.0040
CNN	0.8590 ± 0.0070	0.8597 ± 0.0094	0.8592 ± 0.0101	0.8595 ± 0.0071	0.9380 ± 0.0044
BERT-only (CLS) + MLP	0.8574 ± 0.0069	0.8412 ± 0.0101	0.8823 ± 0.0090	0.8612 ± 0.0073	0.9358 ± 0.0046

Note: [1]Confidence intervals are estimated using bootstrap sampling (1000 iterations) on the test set predictions.

1) Machine learning model

The main traditional machine learning model used is A Linear Support Vector Classifier (Linear SVC). The model will seek to determine an optimal hyperplane that will maximize the margin between positive and negative sentiment classes in the TF-IDF feature space. The objective of optimization is presented as:

$$\min_{w,b,\xi_i} \frac{1}{2}\|w\|^2 + c\sum_{i=1}^{n}\xi_i \tag{2}$$

Subject to:

$$y_i\left(w^T x_i + b\right) \geq 1 - \xi_i, \quad \xi_i > 0 \tag{3}$$

where w is the weight term, b is the bias, ξ_i are the slack variables, and C is a regularization parameter. The model is tuned via cross-validation on the training set and tested on the commonly held-out test set, as per the unified protocol.

2) Deep learning models

a) Convolutional Neural Network (CNN)

CNN system focuses on local features of n-grams by convolving word embeddings with convolution filters. The convolution of a filter of width k is represented as:

$$c_i = f\left(w \cdot x_{i:i+k} + b\right) \tag{4}$$

where $x_{i:i+k}$ depicts a window of word embeddings, w is the convolution kernel, b is the bias, and $f(x)$ is the ReLU activation function. A global max-pooling layer is used, which clusters the most salient features, then fully connected layers and a sigmoid output neuron are used to achieve binary classification. CNNs are added because they have been shown to be effective at capturing the sentiment patterns at the phrase level [5].

b) Bidirectional LSTM (BiLSTM)

BiLSTM model represents the long-term dependencies through the reversal processing of the input sequence. The backward and forward hidden states are calculated as:

$$\overrightarrow{h_t} = LSTM_f(x_t), \quad \overleftarrow{h_t} = LSTM_b(x_t) \tag{5}$$

The final representation at time t is obtained by concatenating the two states:

$$h_t = \left[\overrightarrow{h_t}, \overleftarrow{h_t}\right] \tag{6}$$

Sentiment prediction takes place by using dense layers with sigmoid activation of the output representation. BiLSTM is chosen because of its ability to capture contextual dependencies in long reviews [22].

c) Hybrid model: TF-IDF and BERT feature fusion

A hybrid-model is suggested to use both the lexical and contextual representations, and it is proposed to use sparse TF-IDF features together with dense BERT representations. A pre-trained BERT-base model is not fine-tuned to generate contextual embeddings. In each review, two representations are computed: the [CLS]

token representation and the average-pooled representation of all token representations. The last BERT characteristic vector is determined as:

$$b = \left[h_{CLS}; \frac{1}{T} \sum_{t=1}^{T} h_t \right] \tag{7}$$

Creating a 1536-dimensional rich-textbook element in every review. This is the concatenation of this vector with the TF-IDF feature vector x_{tfidf} to create the hybrid:

$$x_{hybrid} = \left[x_{tfidf}; b \right] \tag{8}$$

Multi-Layer Perceptron (MLP) is a meta-classifier on the combined feature space. It has two hidden layers with ReLU activation and regularization by early stopping. The output is computed as:

$$\hat{y} = \sigma\left(w_2 f\left(w_1 x_{hybrid} + b_1 \right) + b_2 \right) \tag{9}$$

The rationale behind this hybrid design is the previous evidence that lexical and contextual feature combination produces better sentiment classification [21, 23, 24].

F. Optimization Settings

The models are proposed and trained with the Adam optimizer with the following learning rates: 0.001 with the MLP classifier and 2×10^{-5} with BERT-based features. Binary sentiment classification is used with the binary cross-entropy loss function. Training is done with a batch size of 64 and a maximum of 20 epochs, and early stopping (patience = 3 epochs) is applied to prevent overfitting.

G. Evaluation Metrics

All the models are tested on the held-out test set on various performance measures such as Accuracy, Precision, Recall, F1-Score, Cohen Kappa, and Receiver Operating Characteristic Area Under the Curve (ROC-AUC). The accuracy is defined as:

$$Accuracy = \frac{T_p + T_N}{T_p + T_N + F_p + F_N} \tag{10}$$

Precision, Recall, and F1-Score are computed as:

$$Precession = \frac{T_p}{T_p + F_p} \tag{11}$$

$$Recall = \frac{T_p}{T_p + F_N} \tag{12}$$

$$F1 = 2 \times \frac{Precession \times Recall}{Precession + Recall} \tag{13}$$

III. Results and Discussion

A. Performance Comparison

The experimental analysis shows that there are significant performance differences between traditional machine learning, deep learning, and a hybrid sentiment classification model on the IMDb Movie Reviews dataset. A full quantitative comparison of Accuracy, Precision, Recall, F1-Score, Cohen Kappa, and ROC-AUC is given in Table I, allowing a more detailed evaluation of both predictive and classification strength.

As seen in Table I, the fully fine-tuned transformers perform best, with RoBERTa reaching the highest Accuracy of 0.9422 and F1-Score of 0.9427 followed by the fine-tuned BERT and DistilBERT. With the most powerful feature-fusion model ([CLS]+mean), the TF-IDF model achieves an F1-Score of 0.9074, outperforming the classical model, the deep-learning model, as well as the BERT-only model, while underperforming all three fine-tuned transformers.

The feature-fusion combination of TF-IDF features with the contextual BERT embeddings and an MLP meta-classifier is competitive with the strongest classical baselines and definitely outperforms the deep-learning and BERT-only models. It does not, however, match the fully fine-tuned transformers, a difference we examine for statistical significance in the McNemar analysis below.

The Linear SVC model using TF-IDF features achieved a competitive accuracy of 0.8845 and an F1-Score of 0.8854, confirming that well-tuned linear models remain strong baselines on long-form reviews. By contrast, the XGBoost model performed less well, with an accuracy of 0.8698 and an F1-Score of 0.8725, as shown in Table I.

The CNN and BiLSTM deep learning models that used GloVe embeddings showed mediocre performance. BiLSTM model had the highest recall (0.8949) compared to the CNN, which implies that the BiLSTM model is more sensitive to recognizing positive sentiment cases. This result is consistent with previous studies that BiLSTM models tend to be strong, especially in the modeling of long-range dependencies in long textual sequences [5, 22]. Nevertheless, the reduced accuracy of BiLSTM demonstrates a trade-off between specificity and sensitivity, which implies a higher false positive rate.

In addition to quantitative values, qualitative lexical patterns within the dataset are presented as visualizations of word clouds. After preprocessing, word clouds of positive and negative movie reviews were given in Fig. 2, indicating which terms often carry emotions following common words. The words excellent, amazing, and great prevail in positive reviews, and bad, boring, and worst are pre-eminent in negative ones. These visual tendencies substantiate the success of TF-IDF-based lexical representation in the majority of cases of sentiment-discriminative vocabulary, thus providing an explanation for the high success of the linear classifiers that were demonstrated in Table I.

ROC-AUC is accepted as an evaluation metric that is threshold-independent and that estimates the ability of the models to distinguish between positive and negative sentiment classes with all possible classification thresholds. IMDb data is balanced (i.e., the distribution of classes is balanced as well), but ROC-AUC is a significant evaluation measure because it captures the

overall ranking ability and separability of the classifiers beyond fixed decision thresholds. The usage of ROC-AUC in the study is along with threshold-dependent metrics like Accuracy and F1-Score to come out with a more comprehensive evaluation of the model performance. Fig. 3 shows the corresponding ROC curves and the relative performance of all models in discriminating between all possible classes.

Fig. 2. Word clouds illustrating the most frequent terms in positive and negative IMDb movie reviews after preprocessing.

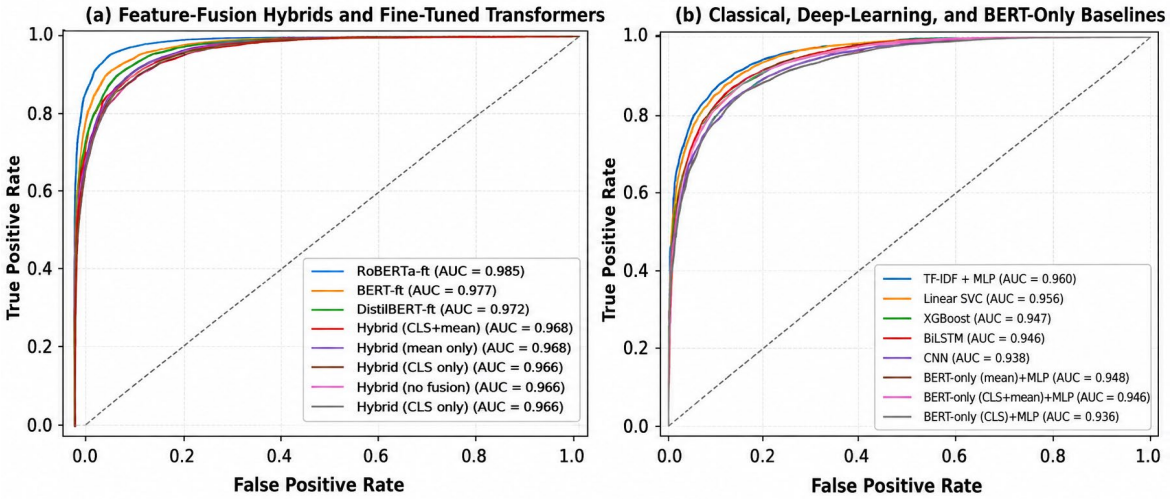

Fig. 3. Word clouds illustrating the most frequent terms in positive and negative IMDb movie reviews after preprocessing.

Taken together, the results show that, although classical and feature-fusion models remain competitive and computationally economical, the fully fine-tuned transformers provide the strongest sentiment classification on this dataset. The feature-fusion model is best understood as an inexpensive alternative rather than a replacement for full fine-tuning when accuracy is the priority.

B. *Strategy Ablation*

Because the feature-fusion model can pool BERT representations in different ways, three configurations were evaluated: the [CLS] vector alone, the mean-pooled vector alone, and their concatenation. As shown in Table II, the concatenated [CLS] + mean configuration (F1 = 0.9074) performed only slightly better than mean pooling alone (F1 = 0.9072), and the difference was not statistically significant (McNemar, p = 0.48). The additional [CLS] vector therefore contributes little beyond mean pooling on this task, a result we report directly rather than presenting the concatenation as a meaningful improvement.

C. *Statistical Significance Analysis*

The paired McNemar test is employed with the feature-fusion model ([CLS] + mean) as reference. The results are presented in Table III. The feature-fusion model is significantly more accurate than all classical, deep-learning, and BERT-only baselines (all $p < 0.001$). However, it is far less accurate than any of the three transformers that have been fully fine-tuned: RoBERTa ($\chi^2 = 145.70$, $p < 0.001$), BERT ($\chi^2 = 38.83$, $p < 0.001$), and DistilBERT ($\chi^2 = 11.89$, $p < 0.001$). Among feature-fusion methods, the [CLS] + mean configuration performs significantly better than the [CLS]-only configuration ($\chi^2 = 5.47$, $p = 0.019$), but does not differ significantly from mean pooling alone ($\chi^2 = 0.51$, $p = 0.48$).

TABLE II. ABLATION OF THE FEATURE-FUSION MODEL'S POOLING CONFIGURATIONS (HELD-OUT TEST SET)

Pooling Configuration	Accuracy (Mean ± CI)	Precision (Mean ± CI)	Recall (Mean ± CI)	F1-Score (Mean ± CI)
Hybrid (CLS + Mean)	0.9069 ± 0.0057	0.9051 ± 0.0078	0.9098 ± 0.0076	0.9074 ± 0.0060
Hybrid (Mean only)	0.9052 ± 0.0054	0.8906 ± 0.0080	0.9245 ± 0.0071	0.9072 ± 0.0056
Hybrid (CLS only)	0.9014 ± 0.0058	0.9064 ± 0.0079	0.8960 ± 0.0084	0.9011 ± 0.0061

TABLE III. PAIRED MCNEMAR COMPARISONS WITH THE FEATURE-FUSION MODEL ([CLS]+MEAN) AS REFERENCE

Model	χ^2 statistic	p-value	Comparison (vs. Hybrid CLS+mean)
RoBERTa-ft	145.70	< 0.001	RoBERTa-ft
BERT-ft	38.83	< 0.001	BERT-ft
DistilBERT-ft	11.89	< 0.001	DistilBERT-ft
Hybrid (mean only)	0.51	0.48	not significant
Hybrid (CLS only)	5.47	0.019	Hybrid (CLS + mean)
Linear SVC	55.31	< 0.001	Hybrid (CLS + mean)
TF-IDF + MLP	38.34	< 0.001	Hybrid (CLS + mean)
XGBoost	119.95	< 0.001	Hybrid (CLS + mean)
BiLSTM	101.32	< 0.001	Hybrid (CLS + mean)
CNN	178.17	< 0.001	Hybrid (CLS+mean)
RoBERTa-ft	145.70	< 0.001	RoBERTa-ft

TABLE IV. COMPUTATIONAL CHARACTERISTICS OF REPRESENTATIVE MODELS

Model	Trainable Parameters	Flops/Sample	Stored Size
RoBERTa-ft	124.6 M	21.8 G	498.7 MB
BERT-ft	109.5 M	21.8 G	438.0 MB
DistilBERT-ft	67.0 M	10.9 G	267.9 MB
Hybrid (CLS+mean)	3.4 M (MLP head)	21.8 G (BERT features)	13.7 MB
Linear SVC	5001	N/A	0.04 MB

D. Computational Complexity Analysis

Table IV reports the trainable-parameter count, forward-pass Floating Point Operations (FLOPs) per sample, and stored size of representative models. The Linear SVC is by far the most economical. The feature-fusion model trains only a small MLP classifier on top of frozen BERT features, and is therefore inexpensive to train; obtaining its features, however, still requires a full forward pass through the frozen BERT encoder. The fully fine-tuned transformers update 67–125 million parameters and require 268–499 MB of storage each. These costs should be weighed against the accuracy differences reported in Table IV.

E. Error Analysis

Confusion matrices for all fourteen models are presented in Figs. 4–6. The feature-fusion model ([CLS] + mean) commits 921 errors out of 9917 (474 false positives and 447 false negatives), fewer than the classical and deep-learning baselines (for example, 1144 for Linear SVC and 1396 for CNN) but more than the fine-tuned transformers (571 for RoBERTa and 738 for BERT). A negation- and length-sensitive analysis reveals that all models have higher error rates on reviews with explicit negation than on reviews without it, and higher on long reviews (over 200 words) than on short reviews. The feature-fusion model has an error rate of 0.082 on non-negated reviews, 0.096 on negated reviews, and 0.083 on short reviews, 0.107 on long reviews, and so do the other two models; similar or larger trends are seen on the other two models. This means that negation and long-range context remain a challenge for lexical approaches, deep-learning, fusion, and transformer approaches.

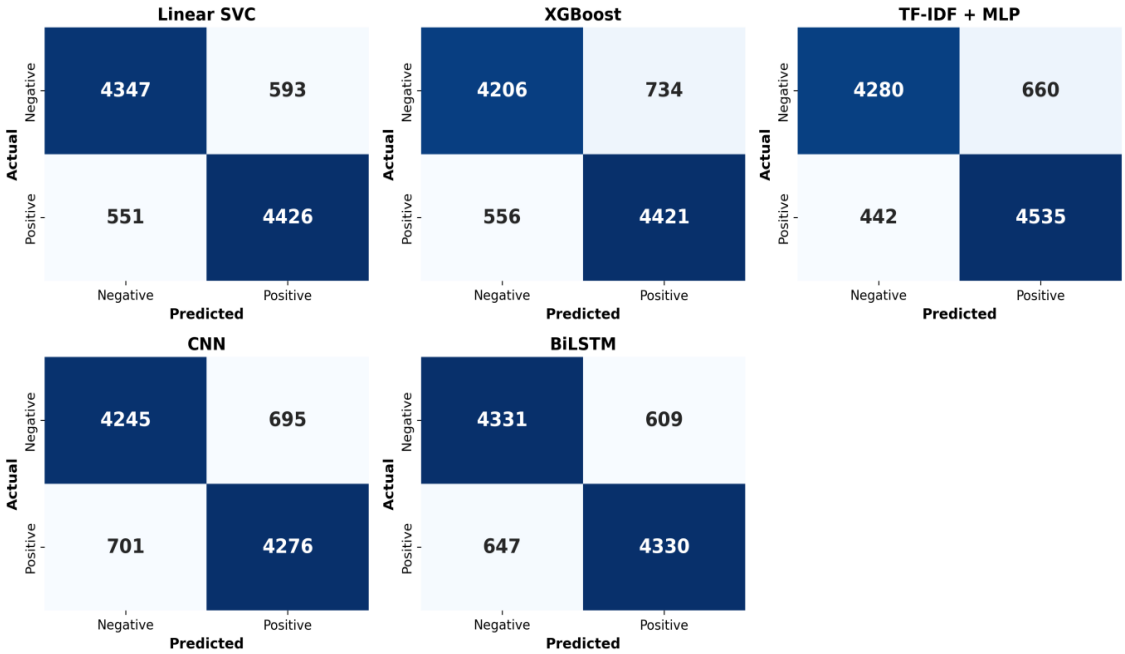

Fig. 4. Confusion matrices for the classical and deep-learning baseline models (n = 9917).

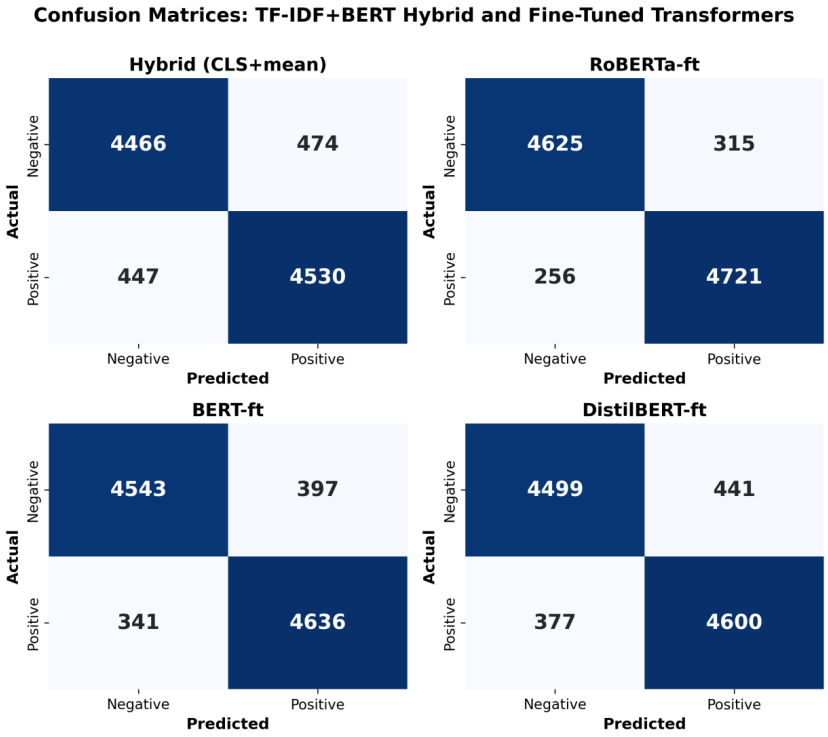

Fig. 5. Confusion matrices for the feature-fusion model and the three fully fine-tuned transformers.

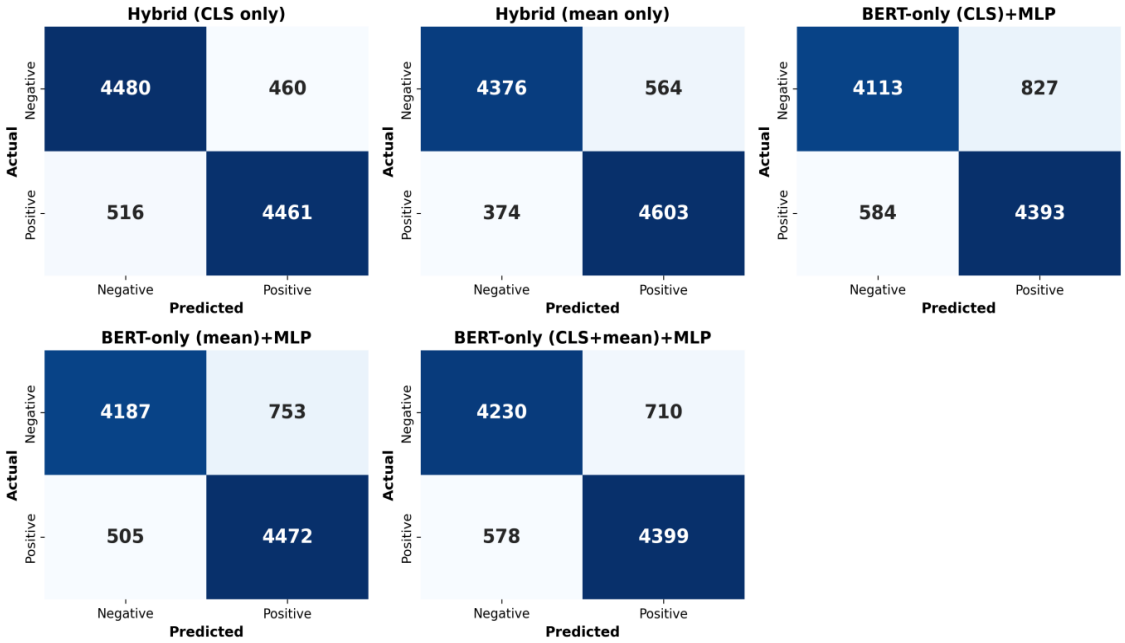

Fig. 6. Confusion matrices for the feature-fusion pooling variants and the BERT-only baselines.

F. Discussion

The following discussion interprets the results above, rather than merely restating them, and relates them to prior work.

The feature-fusion model is significantly better than the classical model and the deep-learning model, but still not as good as fine-tuning the full model, which can be attributed to the different representations each model can access. The feature-fusion model also leverages the fact that it is based on frozen BERT embeddings that hold contextual (bidirectional) information that is not captured by sparse TF-IDF vectors or static GloVe embeddings, thus explaining its statistically significant improvement over the Linear SVC, XGBoost, the CNN, and the BiLSTM (all $p < 0.001$). However, the model is not trained to learn the representations of the BERT encoder, and only a shallow MLP is added on top of the BERT features, which means the model cannot tune the representations of the BERT encoder to the sentiment task. The situation is different for full fine-tuning, which processes the task loss across all transformer layers,

letting the encoder reroute its attention and hidden states to the cues of the sentiment, like negation, contrast and intensifiers. This end-to-end adaptation is the key factor that makes the fine-tuned transformers substantially outperform the frozen feature-fusion model, and accounts for the large difference between them and RoBERTa ($\chi^2 = 145.70$, $p < 0.001$), despite their shared architectural family.

The lack of significant gain ($p = 0.48$) from concatenating the [CLS] and mean-pooled vectors follows from how they are represented in a frozen encoder. If the model is not fine-tuned, the [CLS] token does not exhibit any special properties for summarizing sentences, since it is part of the task-specific training; in such a model, the [CLS] token representation thus provides little discriminative information beyond mean pooling. Mean pooling, which aggregates information across all token positions, is in the frozen setting the more stable and informative sentence representation. Concatenating the two consequently introduces largely redundant rather than complementary signal, as the ablation confirms. This result cautions against the common assumption that combining multiple pooling strategies is automatically beneficial: in the absence of fine-tuning, the additional representation may contribute little.

Finally, these findings both confirm and qualify existing research. They agree with studies reporting that combining lexical (TF-IDF) and contextual (BERT) features improves upon classical baselines, and with the broad consensus that transformer-based models outperform classical and deep-learning approaches on sentiment classification. They qualify, however, the stronger claims in parts of the hybrid-fusion literature that feature-fusion approaches can rival end-to-end fine-tuning: under a controlled protocol with significance testing, the frozen feature-fusion model is clearly and consistently inferior to fully fine-tuned transformers on IMDb. This discrepancy is most plausibly attributable to differences in experimental protocol, dataset composition, and the absence of significance testing in studies that report near-parity. Our controlled comparison therefore indicates that, although feature fusion remains a useful and computationally inexpensive option, it should not be presented as a substitute for fine-tuning when maximum accuracy is the objective.

IV. Conclusion

This paper presented a controlled, reproducible comparison of classical machine-learning, deep-learning, feature-fusion, and fully fine-tuned transformer models for sentiment classification on the IMDb dataset, under a single evaluation protocol with bootstrap confidence intervals, paired McNemar significance testing, confusion-matrix and error analysis, and a computational-complexity comparison. We do not claim the feature-fusion model as a methodological innovation, as its components are established; the contribution is an evidence-based account of the accuracy and cost trade-offs among the model families. The fully fine-tuned transformers achieved the highest performance (RoBERTa, F1 = 0.9427), while the feature-fusion model proved significantly more accurate than the classical, deep-learning, and BERT-only baselines yet significantly less accurate than the fine-tuned transformers. Future work will extend the comparison to fine-grained, multilingual, and aspect-based sentiment tasks.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

S.K.B.: Conceptualization, methodology, software, validation, formal analysis, resources, data curation, writing—original draft preparation, review and editing, visualization, funding acquisition. W.S.: writing—original draft preparation, supervision, project administration and funding acquisition. L.B.: methodology, software, validation. C.W.: methodology, software, validation, investigation, writing—review and editing, visualization. P.D.S.: formal analysis, writing—review and editing. A.R.R.: software, formal analysis, writing—review and editing. D.B.T.: software, investigation, writing—review and editing. A.U.S.: supervision. all authors had approved the final version.

Funding

This work is supported by National Natural Science Foundation of China Project, Project No.: 52577124, and 52207125, and the Key Research and Development Program of Hunan Province Project, Project No.: 2024JK2040 and also Shaoyang University Doctoral Research Start-up Special Project, Project No.: 25KYQD166.

References

[1] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proc. 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, OR, USA, 2011, pp. 142–150.

[2] G. Diri, P. Obiorah, and H. Du, "Comparative study of sentiment analysis techniques: Traditional machine learning vs. deep learning approaches," in *Proc. InSITE Conference*, 2025. doi: 10.28945/5490

[3] M. Cherradi and A. Haddadi, "Comparative analysis of machine learning algorithms for sentiment analysis in film reviews," *Acadlore Transactions on Machine Learning*, vol. 3, no. 3, pp. 137–147, 2024. doi: 10.56578/ataiml030301

[4] M. Bisht, M. Gupta, V. Kharakwal, S. Jain, and S. Verma, "Comparing machine learning algorithms for sentiment analysis in movie reviews," *Journal of Graphic Era University*, vol. 13, no. 1, pp. 205–2202025. doi: 10.13052/jgeu0975-1416.13110

[5] T. Guha and K. G. Mohan, "A hybrid deep learning model for long-term sentiment classification," *Webology*, vol. 17, no. 2, pp. 663–676, 2020. doi: 10.14704/web/v17i2/web17059

[6] T. Abdullah and A. Ahmet, "Deep learning in sentiment analysis: Recent architectures," *ACM Computing Surveys*, vol. 55, no. 8, pp. 1–37, 2022. doi: 10.1145/3548772

[7] C. Wang, P. Nulty, and D. Lillis, "A comparative study on word embeddings in deep learning for text classification," in *Proc. 4th International Conference on Natural Language Processing and Information Retrieval*, Seoul, South Korea, 2020, pp. 37–46. doi: 10.1145/3443279.3443304

[8] S. T. Kokab, S. Asghar, and S. Naz, "Transformer-based deep learning models for the sentiment analysis of social media data," *Array*, vol. 14, 100157, 2022.
[9] S. González-Carvajal and E. C. Garrido-Merchán, "Comparing BERT against traditional machine learning text classification," arXiv preprint, arXiv:2005.13012, 2021.
[10] I. O. Ipadeola, J. A. Ojo, and I. G. Adebayo, "Sentiment analysis of movie reviews using word embeddings and machine learning techniques," *LAUTECH Journal of Engineering and Technology*, vol. 19, no. 3, pp. 155–160, 2025. doi: 10.36108/laujet/5202.91.0341
[11] S. Elmitwalli and J. Mehegan, "Sentiment analysis of COP9-related tweets: A comparative study of pre-trained models and traditional techniques," *Frontiers in Big Data*, vol. 7, 1357926, 2024. doi: 10.3389/fdata.2024.1357926
[12] L. Papa, P. Russo, I. Amerini, and L. Zhou, "A survey on efficient vision transformers: Algorithms, techniques, and performance benchmarking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 12, pp. 7682–7700, 2024. doi: 10.1109/TPAMI.2024.3392941
[13] A. Krouska, C. Troussas, and M. Virvou, "The effect of preprocessing techniques on Twitter sentiment analysis," in *Proc. 7th Int. Conf. Information, Intelligence, Systems & Applications (IISA)*, 2016, pp. 1–5. doi: 10.1109/IISA.2016.7785373
[14] P. Aprilio, M. Felix, P. S. Nugraha, and H. Fahmi, "Hybrid feature combination of TF-IDF and BERT for enhanced information retrieval accuracy," *Jurnal Informatika dan Sains*, vol. 8, no. 1, pp. 8–15, 2025.
[15] I. Permana *et al.*, "Benchmarking BoW, TF-IDF and BERT-based sentiment analysis in low-resource Minang language," in *Proc. 2025 Tenth International Conference on Informatics and Computing (ICIC)*, 2025, pp. 1–6.
[16] M. M. Hossain, M. S. Hossain, M. Safran, S. Alfarhood, M. Alfarhood, and M. F. Mridha, "A hybrid attention-based transformer model for Arabic news classification using text embedding and deep learning," *IEEE Access*, vol. 12, pp. 198046–198066, 2024. doi: 10.1109/ACCESS.2024.3522061
[17] J. W. Sun, J. Q. Bao, and L. P. Bu, "Text classification algorithm based on TF-IDF and BERT," in *Proc. 11th International Conference of Information and Communication Technology (ICTech)*, 2022, pp. 1–4. doi: 10.1109/ICTech55460.2022.00009
[18] A. D. Bhargavi, "Comparative study of static and contextual text vectorization for sentiment analysis," *International Journal for Research in Applied Science and Engineering Technology*, vol. 13, no. 4, pp. 484–488, 2025. doi: 10.22214/ijraset.2025.73045
[19] D. D. N. Cahyo, F. Farasalsabila, V. B. Lestari, H. Hanafi, T. Lestari, F. R. Islami, and M. A. Maulana, "Sentiment analysis for IMDb movie review using Support Vector Machine (SVM) method," *Inform: Jurnal Ilmiah Bidang Teknologi Informasi dan Komunikasi*, vol. 8, no. 2, 2023. doi: 10.25139/inform.v8i2.5700
[20] M. S. Başarslan and F. Kayaalp, "Sentiment analysis on social media reviews datasets with deep learning approach," *Sakarya University Journal of Computer and Information Sciences*, vol. 4, no. 1, pp. 1–14, 2021. doi: 10.35377/saucis.04.01.833026
[21] M. Chiny, M. Chihab, O. Bencharef, and Y. Chihab, "LSTM, VADER and TF-IDF based hybrid sentiment analysis model," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 7, pp. 265–275, 2021. doi: 10.14569/IJACSA.2021.0120730
[22] G. Xu, Y. Meng, X. Qiu, Z. Yu, and X. Wu, "Sentiment analysis of comment texts based on BiLSTM," *IEEE Access*, vol. 7, pp. 51522–51532, 2019. doi: 10.1109/ACCESS.2019.2909919
[23] I. Kanwal, F. Wahid, S. Ali, A. Rehman, A. Alkhayyat, and A. Al-Radaei, "Sentiment analysis using hybrid model of stacked auto-encoder-based feature extraction and LSTM-based classification approach," *IEEE Access*, vol. 11, pp. 124181–124197, 2023. doi: 10.1109/ACCESS.2023.3313189
[24] A. Albladi, M. K. Uddin, M. Islam, and C. Seals, "TWSSenti: A novel hybrid framework for topic-wise sentiment analysis using transformer models," arXiv preprint, arXiv:2504.09896, 2025.

Copyright © 2026 by the authors. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited (CC BY 4.0).