A Comparative Study of CNN-BiLSTM and TF-IDF–Naive Bayes for Sentiment Classification on Movie Reviews: Performance and Efficiency Trade-offs

Muhammad Fadhil Dwisaputra

Prodi Informatika, Fakultas Sains, Universitas Islam Negeri Sultan Maulana Hasanuddin Banten, Jalan Syech Nawawi Al-Bantani, Kelurahan Sukawana, Kecamatan Curug, Kota Serang, Banten, Indonesia 42123
Email: 241730019.muhammadfadhildwisaputra@uinbanten.ac.id

Abstract

Sentiment analysis on movie reviews has become an important task for understanding public opinion, yet many high-performing deep learning models proposed in the literature rely on increasingly complex architectures such as multichannel embeddings, kernel-based projections, or attention-augmented transformers. This added complexity often comes without a clear report of computational cost, making it difficult to judge whether the performance gain justifies the resource trade-off. This study investigates whether a deliberately simplified Convolutional Neural Network combined with a Bidirectional Long Short-Term Memory (CNN-BiLSTM) architecture can achieve competitive performance compared to a classical baseline, TF-IDF combined with Naive Bayes, while explicitly reporting training efficiency. Both models were trained and evaluated on the IMDB Dataset of 50,000 movie reviews, split into 80% training, 10% validation, and 10% testing sets. Experimental results show that the proposed CNN-BiLSTM achieved an accuracy of 86.58%, precision of 90.38%, recall of 81.88%, F1-score of 85.92%, and AUC of 0.946, while the TF-IDF-Naive Bayes baseline achieved an accuracy of 85.98%, precision of 86.58%, recall of 85.16%, F1-score of 85.86%, and AUC of 0.932. The proposed method required approximately 203 seconds of training time, compared to 0.07 seconds for the baseline. These findings indicate that the performance improvement offered by the deep learning approach is modest relative to its substantially higher computational cost, providing practical guidance for method selection in resource-constrained settings.

Keywords: *Baseline comparison; CNN-BiLSTM; Deep learning; Movie review; Naive Bayes, Sentiment analysis; TF-IDF.*

1. Introduction

The rapid growth of user-generated content on streaming platforms, e-commerce sites, and review aggregators has made sentiment analysis an essential tool for automatically understanding public opinion at scale. In the domain of movie reviews specifically, sentiment classification enables recommendation systems, content moderation, and market research to operate without manual reading of thousands of reviews. Over the past five years, deep learning approaches—particularly architectures combining Convolutional Neural Networks (CNN) with recurrent units such as Long Short-Term Memory (LSTM) or its bidirectional variant (BiLSTM)—have become the dominant paradigm for this task, frequently outperforming traditional machine learning baselines on benchmark datasets such as IMDB and Sentiment140.

However, a closer examination of recent literature reveals a recurring pattern: the models reporting the highest accuracy tend to introduce additional architectural complexity, such as positional embeddings combined with multichannel convolutions, kernel variance projection layers, or attention mechanisms layered on top of recurrent units. While these additions often yield measurable accuracy gains, the corresponding computational cost—specifically training time and resource consumption—is rarely reported or compared against simpler alternatives. This omission makes it difficult for practitioners, especially those operating with limited computational resources, to make an informed decision about whether the added complexity is justified.

This study addresses this gap by implementing a deliberately simplified CNN-BiLSTM architecture—using a single embedding layer without auxiliary components such as attention or kernel projections—and comparing it directly against a classical baseline, TF-IDF combined with Naive Bayes, on the same dataset (IMDB 50K Movie Reviews) under identical preprocessing and evaluation conditions. In addition to standard classification metrics (accuracy, precision, recall, F1-score, and AUC), this study explicitly measures and reports training time for both methods, providing a more complete picture of the performance-efficiency trade-off than is typically presented in prior work.

The contributions of this paper are threefold: (1) implementation of a simplified CNN-BiLSTM architecture evaluated on the full-scale IMDB 50K dataset; (2) a direct, controlled comparison against a classical TF-IDF + Naive Bayes baseline that includes both predictive performance and training-time efficiency; and (3) a discussion of the practical implications of these trade-offs for method selection in resource-constrained sentiment analysis applications.

2. Related Works

Research on deep learning-based sentiment analysis has progressed considerably in recent years, with most studies focusing on improving classification accuracy through architectural innovation. Myint et al. [1] proposed a multi-task transformer framework with attention mechanisms for simultaneous sentiment and emotion classification on crisis-related Twitter data, achieving a macro-F1 of 0.90 using BERTweet, though their evaluation was restricted to the crisis-event domain and did not extend to product or media reviews.

In the domain most relevant to this study, Atandoh et al. [9] introduced PEW-MCAB, a model combining positional and GloVe embeddings with multichannel CNN and attention-based BiLSTM, achieving 90.3% accuracy on the IMDB dataset. Similarly, Al-Saadi et al. [5] proposed a Kernel Variance Projection (KVP) combined with a ConvBiLSTM hybrid, reporting 88.39% accuracy and an F1-score of 88.43% on IMDB. Both studies demonstrate that augmenting CNN-BiLSTM architectures with additional components can push accuracy above the high-80s to low-90s range; however, neither study reports training time, computational cost, or a direct comparison against a non-deep-learning baseline on the same dataset scale, leaving the performance-efficiency trade-off unaddressed.

Other studies have explored related but distinct objectives. Ahmed and Wang [3] proposed an embedded-CNN with BiLSTM for fine-grained,

aspect-based product sentiment, reporting F1-scores around 80–82% on SemEval and product review datasets—lower than the document-level accuracy typically reported on IMDB, reflecting the added difficulty of aspect-level classification. Onan [6] introduced a bidirectional CNN-RNN architecture enhanced with a group-wise enhancement mechanism, evaluated across twelve benchmark datasets and achieving up to 95.04% accuracy on certain corpora, though IMDB was not among the datasets tested. Siddiqui et al. [2] integrated sentiment analysis into a CNN-LSTM pipeline for stock price forecasting rather than classification, reporting regression error (MSE) instead of classification metrics, which makes their results not directly comparable to standard sentiment classification benchmarks.

Several studies have also addressed sentiment analysis in low-resource or domain-specific settings. Kusumaningrum et al. [8] developed a CNN+LSTM model for multilevel sentiment analysis of Indonesian hotel reviews, notably including a web-based deployment (Sentilytics 1.0) and reporting an F1-score of 0.95 at the document level, demonstrating the practical applicability of deep learning sentiment models beyond benchmark datasets. Rumya et al. [10] applied hybrid CNN+LSTM and customized BERT models to Bengali text, reporting accuracy as high as 99.35%, while Chakraborty et al. [12] proposed an explainable ModernBERT framework for aviation workforce sentiment, achieving an F1-score of 95.69% with reduced encoder depth for efficiency, and incorporating LIME for interpretability.

Aspect-based sentiment analysis has also been explored through graph-based methods; Saeedmehr et al. [11] proposed an Attention-Enhanced Graph Convolutional Network (AE-GCN) leveraging affective knowledge, reporting a modest average improvement of 0.44% in accuracy and 0.60% in F1-score over the best existing ABSA baselines on SemEval restaurant and laptop datasets—illustrating that increased architectural complexity does not always translate into substantial performance gains. Chen and Xie [7] applied a CNN-BiLSTM-Attention architecture to Japanese short text classification, reporting improvements in word-order sensitivity rather than standard accuracy metrics, limiting direct comparability with English-language sentiment benchmarks.

Across this body of literature, two gaps consistently emerge. First, the highest-performing models on IMDB-scale datasets rely on architectural additions whose computational trade-offs are not transparently reported. Second, head-to-head comparisons between deep learning methods and classical machine learning baselines on the same dataset, inclusive of training-time efficiency, remain scarce. This study is positioned to address both gaps directly.

3. Proposed Method

A. Research Framework

The overall research framework consists of five sequential stages: dataset collection, data preprocessing, model implementation (proposed method and baseline), model training, and evaluation. Both the proposed CNN-BiLSTM model and the TF-IDF + Naive Bayes baseline were trained and evaluated under identical data splits to ensure a fair comparison.

B. Dataset

This study uses the IMDB Dataset of 50K Movie Reviews, a publicly available dataset retrieved from Kaggle (https://www.kaggle.com/datasets/Lakshm-i25npathi/imdb-dataset-of-50k-movie-reviews), originally derived from the Stanford Large Movie Review Dataset. The dataset consists of 50,000 movie reviews labeled as either positive or negative, with a balanced class distribution of 25,000 reviews per class. The data was split into 80% training (40,000 reviews), 10% validation (5,000 reviews), and 10% testing (5,000 reviews), using stratified sampling to preserve class balance across splits.

C. Data Preprocessing

Each review underwent a cleaning procedure consisting of: lowercasing all text, removing HTML tags (notably
 tags present in the raw IMDB text), removing URLs, removing non-alphabetic characters and punctuation, and normalizing whitespace. For the proposed CNN-BiLSTM model, cleaned text was tokenized using a vocabulary limited to the 10,000 most frequent

words, with out-of-vocabulary tokens mapped to a designated <OOV> token. Sequences were padded or truncated to a fixed length of 200 tokens. For the baseline model, cleaned text was transformed using Term Frequency-Inverse Document Frequency (TF-IDF) vectorization, also limited to the top 10,000 features, to ensure both methods operate on a comparably sized feature space.

D. Proposed Architecture: CNN-BiLSTM

The proposed model adopts a deliberately compact architecture to test whether competitive performance can be achieved without auxiliary components used in prior work (e.g., attention mechanisms, dual embeddings, or kernel projections). The architecture consists of: (1) an Embedding layer mapping each token to a 128-dimensional dense vector; (2) a 1D Convolutional layer with 128 filters and a kernel size of 5, using ReLU activation, to extract local n-gram-like features; (3) a MaxPooling1D layer with pool size 2 to reduce dimensionality while retaining salient features; (4) a Bidirectional LSTM layer with 64 units to capture sequential dependencies in both forward and backward directions; (5) a Dropout layer (rate 0.5) for regularization; (6) a Dense layer with 64 units and ReLU activation; (7) a second Dropout layer (rate 0.3); and (8) a final Dense output layer with a single unit and sigmoid activation for binary classification.

The model was compiled using the Adam optimizer and binary cross-entropy loss. Training employed two callbacks: EarlyStopping, which monitors validation loss with a patience of 3 epochs and restores the best-performing weights, and ReduceLROnPlateau, which halves the learning rate after 2 epochs without improvement in validation loss, down to a minimum of 1×10^{-6}. The model was trained for a maximum of 15 epochs with a batch size of 64; in practice, training terminated early after 4 epochs once validation loss ceased to improve, with the best weights (epoch 1) automatically restored.

E. Baseline: TF-IDF + Naive Bayes

As the baseline (comparison) method, this study employs Multinomial Naive Bayes trained on TF-IDF-vectorized text. This combination was selected as the baseline because it represents a well-established, computationally inexpensive classical approach widely used as a reference point in sentiment analysis literature, allowing a clear assessment of the marginal benefit—if any—offered by the more computationally intensive deep learning approach.

4. Experimental Setup

All experiments were implemented in Python using TensorFlow/Keras for the CNN-BiLSTM model and scikit-learn for the TF-IDF + Naive Bayes baseline. Training was conducted on a system equipped with an NVIDIA GeForce RTX 3050 Laptop GPU; however, due to TensorFlow's discontinued native GPU support on Windows since version 2.11, training for the proposed model was executed on CPU. The evaluation was performed on a held-out test set of 5,000 reviews (2,500 positive, 2,500 negative), which was not used during training or hyperparameter selection. Five evaluation metrics were used: Accuracy, Precision, Recall, F1-score, and Area Under the ROC Curve (AUC), supplemented with confusion matrices for both models and training/validation curves for the proposed model. Training time, measured in wall-clock seconds, was recorded for both methods to assess computational efficiency.

5. Results and Discussion

A. Classification Performance

Table I summarizes the classification performance of both models on the test set.

Table 1. Performance Comparison

Model	Acc.	Prec.	Rec.	F1
CNN-BiLSTM	0.8658	0.9038	0.8188	0.8592
TF-IDF+NB	0.8598	0.8658	0.8516	0.8586

As shown in Table I, the proposed CNN-BiLSTM model achieves a marginally higher accuracy (86.58% vs. 85.98%, a difference of 0.60 percentage points) and a notably higher precision (90.38% vs. 86.58%) compared to the baseline. However, the baseline achieves a higher recall (85.16% vs. 81.88%), indicating that the proposed model is more conservative in predicting the positive class, producing fewer false positives at the cost of more false negatives. The resulting F1-

scores are nearly identical (0.8592 vs. 0.8586), suggesting that, despite their different architectures, both models achieve comparable overall balance between precision and recall on this dataset.

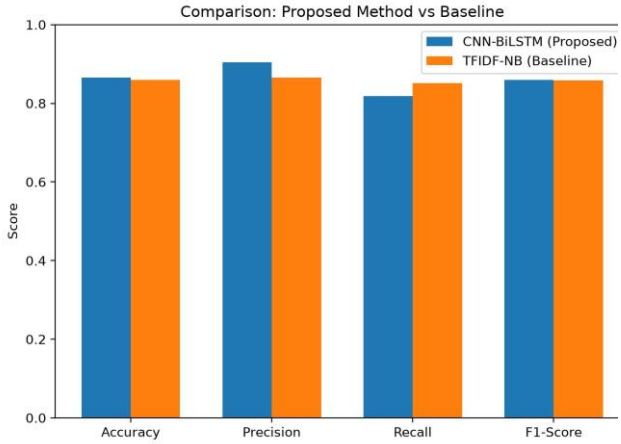

Fig. 1. *Comparison of accuracy, precision, recall, and F1-score between the proposed CNN-BiLSTM and the TF-IDF + Naive Bayes baseline.*

B. Confusion Matrix Analysis

Figures 2 and 3 present the confusion matrices for the proposed and baseline models, respectively. The CNN-BiLSTM model correctly classified 2,282 of 2,500 negative reviews and 2,047 of 2,500 positive reviews, with 453 positive reviews misclassified as negative (false negatives) and 218 negative reviews misclassified as positive (false positives). In contrast, the baseline correctly classified 2,170 negative and 2,129 positive reviews, with a more balanced error distribution of 330 false positives and 371 false negatives. This pattern confirms the precision-recall trade-off observed in Table I: the proposed model is biased toward minimizing false positives, while the baseline distributes its errors more evenly between classes.

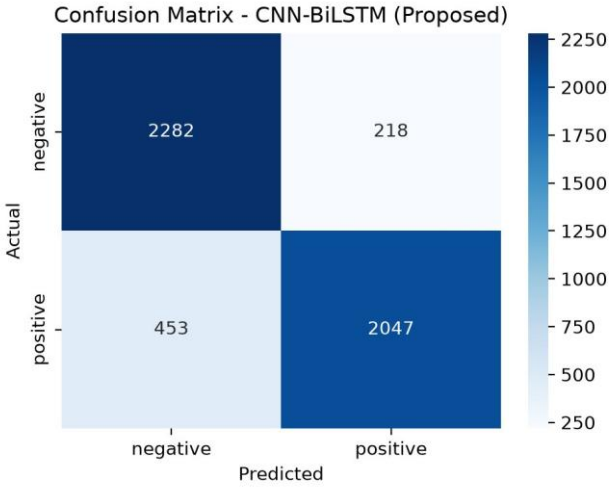

Fig. 2*. Confusion matrix of the proposed CNN-BiLSTM model on the test set.*

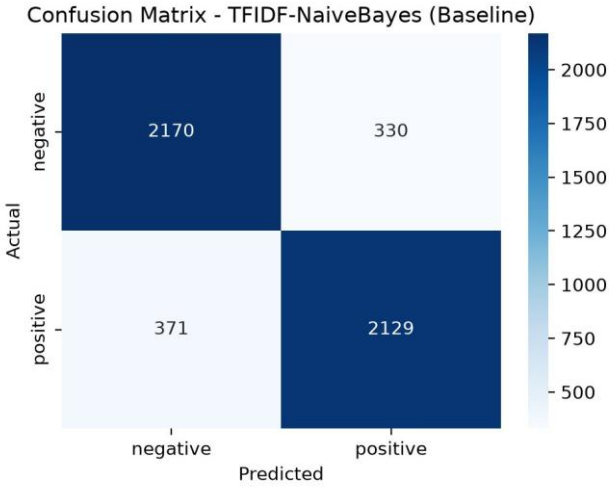

Fig. 3*. Confusion matrix of the TF-IDF + Naive Bayes baseline on the test set.*

C. ROC Curve and AUC

The ROC curves in Figures 4 and 5 further support the performance comparison. The proposed CNN-BiLSTM achieved an AUC of 0.946, slightly higher than the baseline's AUC of 0.932. This indicates that, across all classification thresholds, the proposed model maintains a marginally better ability to discriminate between positive and negative reviews, even though the gain (0.014 AUC points) is relatively small in absolute terms.

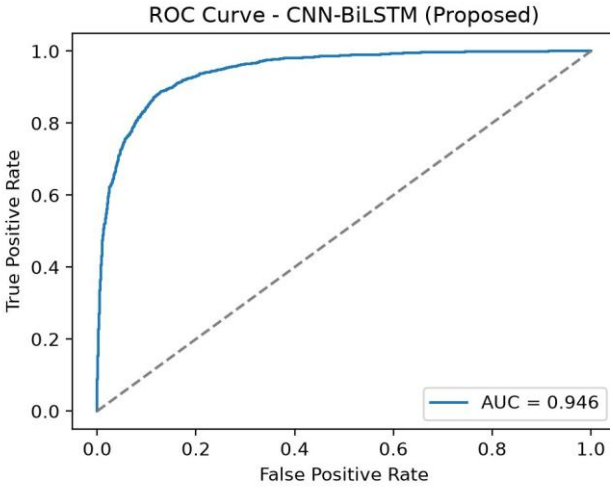

Fig. 4. *ROC curve of the proposed CNN-BiLSTM model (AUC = 0.946).*

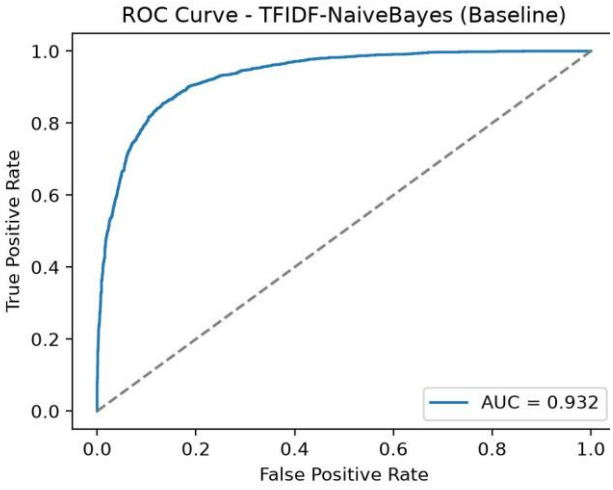

Fig. 5. *ROC curve of the TF-IDF + Naive Bayes baseline (AUC = 0.932).*

D. Training Behavior and Efficiency

Figure 6 shows the training and validation loss and accuracy curves for the proposed CNN-BiLSTM model. Training accuracy increased steadily from 80.6% to 96.7% across four epochs, while training loss decreased from 0.404 to 0.101, indicating that the model was effectively learning patterns from the training data. However, validation loss exhibited a different trend: after an initial value of 0.290 at epoch 1, it increased to 0.332, then sharply to 0.567 at epoch 3, before partially recovering to 0.378 at epoch 4. This divergence between decreasing training loss and increasing validation loss is a classic indicator of overfitting, where the model increasingly memorizes training-specific patterns rather than generalizing. The EarlyStopping callback correctly detected this trend and halted training after epoch 4, restoring the model weights from epoch 1—the point of lowest validation loss—for final evaluation.

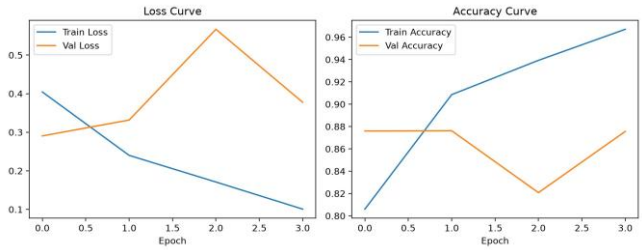

Fig. 6. *Training and validation loss/accuracy curves for the proposed CNN-BiLSTM model across 4 epochs (early-stopped).*

In terms of computational efficiency, the proposed CNN-BiLSTM required approximately 203 seconds (3.4 minutes) to train on a CPU, while the TF-IDF + Naive Bayes baseline required only 0.07 seconds—a difference of approximately four orders of magnitude. Table II summarizes this trade-off.

Table 2. Training Time Trade-Off

Model	F1	AUC	Time
CNN-BiLSTM	0.8592	0.946	~203 s
TF-IDF+NB	0.8586	0.932	~0.07 s

This result directly addresses the research gap identified in Section II: the proposed simplified CNN-BiLSTM achieves only a 0.06 percentage-point improvement in F1-score and a 0.014 improvement in AUC over the classical baseline, while requiring roughly 2,900 times more training time. This suggests that, for the binary movie review sentiment classification task on this dataset, the marginal performance gain from the deep learning approach may not justify its substantially higher computational cost in resource-constrained or rapid-deployment scenarios—though the higher precision and AUC of the proposed model may still be preferable in applications where minimizing false positives is critical.

E. Comparison with Prior Work

Compared to prior studies evaluated on the same IMDB dataset, the proposed model's accuracy of 86.58% is lower than the 90.3% reported by Atandoh et al. [9] using the more complex PEW-MCAB architecture, and the 88.39% reported by Al-Saadi et al. [5] using KVP combined with

ConvBiLSTM. This is consistent with expectations, as both prior models incorporate additional components (multichannel CNN with positional embeddings, and kernel variance projection, respectively) that this study deliberately omitted to isolate the effect of architectural simplicity. The gap of 3.7–4.7 percentage points in accuracy may be considered the approximate cost of foregoing those additional components, providing a reference point for practitioners weighing architectural complexity against implementation and training simplicity.

F. Strengths and Limitations

The primary strength of the proposed approach lies in its architectural simplicity and transparency: it uses standard, well-understood layers without requiring specialized components, making it easier to implement, train, and interpret than more complex alternatives in the literature. Additionally, by explicitly reporting training time alongside accuracy metrics, this study provides a more complete basis for method selection than prior work that omits such efficiency reporting.

Nonetheless, several limitations should be acknowledged. First, the early stopping at epoch 4 suggests the model may not have fully converged to its optimal performance; with regularization adjustments (e.g., increased dropout or additional data augmentation), higher accuracy may be achievable without sacrificing generalization. Second, the experiment was conducted on a single dataset (IMDB); generalizability to other domains (e.g., product reviews, social media) was not tested. Third, training time was measured on CPU due to TensorFlow's native Windows GPU limitations, and results may differ if executed under a GPU-enabled environment such as Linux or WSL2.

G. Future Work

Future research directions include: (1) evaluating the proposed architecture under GPU-accelerated training to obtain a more representative efficiency comparison; (2) extending the comparison to include additional deep learning baselines such as standalone LSTM, standalone CNN, and transformer-based models (e.g., DistilBERT) to position the simplified CNN-BiLSTM more comprehensively within the performance-efficiency spectrum; (3) testing generalizability on other sentiment domains, such as product or hotel reviews; and (4) deploying the trained model as a simple web-based inference tool, building on the predict.py inference script developed as part of this study's reproducible artifact.

6. Conclusion

This study implemented and evaluated a simplified CNN-BiLSTM architecture for binary sentiment classification on the IMDB 50K Movie Reviews dataset, comparing it directly against a classical TF-IDF + Naive Bayes baseline under identical data conditions. The proposed model achieved an accuracy of 86.58%, F1-score of 0.8592, and AUC of 0.946, marginally outperforming the baseline's accuracy of 85.98%, F1-score of 0.8586, and AUC of 0.932. However, this modest performance gain came at the cost of approximately 2,900 times longer training time (203 seconds vs. 0.07 seconds). These findings suggest that architectural simplicity in deep learning sentiment models can yield results competitive with—though not dramatically superior to—classical baselines, and that training efficiency should be considered alongside accuracy when selecting a sentiment analysis method for practical deployment. The complete source code, trained models, and experimental artifacts produced in this study are made publicly available to support reproducibility and further research.

Acknowledgment

The author would like to thank the instructor of the Artificial Intelligence course for guidance throughout this research, and acknowledges the use of the publicly available IMDB Dataset of 50K Movie Reviews from Kaggle, originally derived from the Stanford Large Movie Review Dataset.

References

[1] A. Daza, N. D. Gonzalez Rueda, M. S. Aguilar Sanchez, W. F. Robles Espiritu, and M. E. Chauca Quinones, "Sentiment Analysis on E-Commerce Product Reviews Using Machine Learning and Deep Learning Algorithms: A Bibliometric Analysis, Systematic Literature Review, Challenges and Future Works," International Journal of Information Management Data Insights, vol.

4, no. 2, 2024, Art. no. 100267.

[2] 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, 2011, pp. 142–150.

[3] A. O. Agboola, K. T. Ladoja, and O. F. W. Onifade, "Movie Recommendation system with sentiment analysis using deep learning algorithms," Egyptian Informatics Journal, vol. 33, 2026, Art. no. 100905.

[4] A. Onan, "Bidirectional convolutional recurrent neural network architecture with group-wise enhancement mechanism for text sentiment classification," Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 5, 2022.

[5] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in Proc. 3rd International Conference on Learning Representations (ICLR), 2015.

[6] E. Atandoh, F. Zhang, M. A. Al-antari, P. M. Addo, and D. Gu, "Scalable deep learning framework for sentiment analysis prediction for online movie reviews," Heliyon, vol. 10, no. 12, 2024.

[7] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[8] F. Siddiqui, M. R. Shakeel, S. Alam, M. Manzoor, S. Siddiqui, and P. Ranjan, "Integrating sentiment analysis with CNN-LSTM models for stock price forecasting," IIMB Management Review, vol. 38, no. 1, 2026, Art. no. 100675.

[9] H. Al-Saadi, C. Chow, K. Wong, and M. Khairuddin, "Sentiment analysis using Kernel Variance Projection and LR-BiLGMP-Skd deep learning model," Ain Shams Engineering Journal, 2026.

[10] K. Myint, S. C. B. Lo, and Y. Zhang, "Unveiling the dynamics of crisis events: Sentiment and emotion analysis via multi-task learning with attention mechanism," Information Processing & Management, vol. 61, no. 5, 2024.

[11] M. Abadi et al., "TensorFlow: A System for Large-Scale Machine Learning," in Proc. 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI), 2016, pp. 265–283.

[12] M. Ahmed and H. Wang, "A fine-grained deep learning model using embedded-CNN with BiLSTM for exploiting product sentiments," Alexandria Engineering Journal, 2022.

[13] M. Saeedmehr, M. Shams, and N. Alambardar Meybodi, "The impact of attention mechanisms on aspect-based sentiment analysis performance using affective knowledge-enhanced GCNs," Computer Speech & Language, 2026.

[14] M. Z. Ayman, S. Akash, F. Akhter, M. Mridul, and S. Sutradhar, "BanglaEcomReviewCorpus: A dataset for e-commerce product review sentiment analysis," Data in Brief, 2026.

[15] P. Rumya, A. Al Maruf, and Z. Aung, "Multi-label emotion and sentiment detection from Bengali text using hybrid deep learning model," e-Prime - Advances in Electrical Engineering, Electronics and Energy, 2026.

[16] R. Kusumaningrum, S. K. Nisa, A. P. Jayanto, R. Nawangsari, and A. Wibowo, "Deep learning-based application for multilevel sentiment analysis of Indonesian hotel reviews," Heliyon, vol. 9, no. 6, 2023.

[17] S. Chakraborty, P. Das, F. Al Farid, F. H. Bhoyan, F. Y. Sadeque, J. Uddin, and H. Abdul Karim, "An explainable transformer framework for sentiment analysis in aviation workforce data," Decision Analytics Journal, vol. 18, 2026, Art. no. 100693.

[18] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[19] Y. Chen and X. Xie, "Japanese Short Text Classification Based on CNN-BiLSTM-Attention," Procedia Computer Science, 2025.

[20] Y. Kim, "Convolutional Neural Networks for Sentence Classification," in Proc. 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1746–1751.