IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 15, No. 6, December 2026, pp. 5822~5837
ISSN: 2252-8938, DOI: 10.11591/ijai.v15.i6.pp5822-5837

Machine learning techniques for sentiment analysis classification: a comparative study

Billal El Khattabi[1], Hicham Omara[2], Adnan Souri[1]
[1]Department of Computer Science Faculty of Sciences, Abdelmalek Essaadi University, Tetouan, Morocco
[2]Department of Computer Science, Faculty of Polydisciplinary, Sidi Mohammed Ben Abdellah University, Taza, Morocco

Article Info

Article history:

Received Oct 13, 2025
Revised Aug 11, 2026
Accepted Aug 31, 2026

Keywords:

Deep learning
Hybrid approaches
Machine learning
Natural language processing
Sentiment analysis
Text classification

ABSTRACT

With the exponential growth of online communication, there is an urgent need for robust tools to decode human sentiment in digital text. This research utilized the Sentiment140 dataset, capturing a wide range of real-world Twitter (now X) messages, both positive and negative. This study conducted a rigorous comparative evaluation of various algorithms, including traditional classifiers like naive Bayes (NB), support vector machine (SVM), and logistic regression (LR), alongside deep learning architectures such as convolutional neural networks (CNN), recurrent neural networks (RNN), long-short term memory (LSTM), and bidirectional long-short term memory (BiLSTM). To enhance representational quality, feature engineering was approached through unigram, bigram, term frequency-inverse document frequency (TF-IDF), and advanced embeddings like word to vector (Word2Vec) and global vectors for word representation (GloVe). The analysis revealed a clear leader: transformer-based models, with bidirectional encoder representations from transformers (BERT) achieving an impressive accuracy of 92.76%, surpassing both classical and deep architectures. Notably, hybrid approaches like NB-SVM-maximum entropy (ME) and robustly optimized BERT pretraining approach (RoBERTa)-LSTM also demonstrated competitive performance, underscoring the ongoing relevance of feature-driven and ensemble methods. This finding prompts essential reflection: while transformers excel in contextual understanding, the choice of feature representation and hybridization strategies remains critical for optimizing performance. Ultimately, this study highlights the transformative role of modern architectures in sentiment analysis while reinforcing the importance of methodological diversity in advancing natural language processing (NLP).

This is an open access article under the CC BY-SA license.

Corresponding Author:

Billal El Khattabi
Department of Computer Science, Faculty of Sciences, Abdelmalek Essaadi University
Tetouan, Morocco
Email: b.khattabi@gmail.com

1. INTRODUCTION

Sentiment analysis, a key area within natural language processing (NLP), focuses on the automatic recognition and classification of emotions conveyed in written content [1]. The surge in social media usage has made public opinions more readily available, positioning sentiment analysis as a vital instrument for gauging public emotions across multiple sectors, such as business and politics [1]. This analytical process comprises several fundamental phases: initial data cleaning, extraction of relevant features, and the final classification of sentiments [2]. During preprocessing, raw text data is cleaned to eliminate unnecessary elements, such as stop words, special characters, and numbers. This phase also includes transforming the text

into usable features through methods like term frequency-inverse document frequency (TF-IDF), global vectors for word representation (GloVe), fastText, tokenization, bag-of-words, part-of-speech tagging, and word to vector (Word2Vec). Following this, the processed text undergoes classification into different sentiments using various machine learning (ML) techniques, including logistic regression (LR), naive Bayes (NB), support vector machines (SVM), and random forests, as well as deep learning approaches like long short-term memory (LSTM), recurrent neural networks (RNN), bidirectional long-short term memory (BiLSTM) and transformer models such as bidirectional encoder representations from transformers (BERT) and robustly optimized BERT pretraining approach (RoBERTa), particularly utilizing the Sentiment140 dataset.

In this context, the proposed research explores the potential ML as a transformative approach in sentiment analysis. This study focuses on evaluating the accuracy of different algorithms to provide a comprehensive comparison of their performance. The structure of this paper is designed to guide the reader smoothly through our findings. After this introduction, section 2 delves into relevant literature, discussing previously used classification methods. Section 3 outlines the proposed chosen methodologies and the details of the Sentiment140 dataset. The core findings are presented in section 4, followed by a discussion of their implications in section 5. Finally, section 6 summarizes conclusions and suggests directions for future research. Throughout this study seek to answer critical questions: how do these ML models compare in their effectiveness for sentiment classification on the Sentiment140 dataset? Can they be considered reliable tools for this purpose?

2. RELATED WORK

Over the past two decades, research in sentiment analysis has advanced significantly, transitioning from conventional ML techniques to deep neural networks (DNN) and, more recently, transformer-based and hybrid paradigms. This section organizes previous literature into three main themes: classical ML algorithms, deep learning approaches, and modern transformer or hybrid models. Additionally, a comparative evaluation across dataset selection, feature representation, and computational performance highlights the specific research gaps addressed by this paper. Early research in sentiment analysis predominantly relied on traditional supervised ML algorithms combined with handcrafted feature representations. Commonly used classifiers include NB, SVM, LR, k-nearest neighbors (KNN), and decision trees.

Several studies demonstrated the effectiveness of probabilistic and margin-based classifiers when paired with lexical features such as unigram and bigram models. Rezwanul *et al.* [3] compared SVM and KNN on Twitter data (now X), reporting that n-gram representations significantly improved classification accuracy. Similarly, Jasy *et al.* [4] showed that multinomial naive Bayes (MNB) achieved competitive results on the Sentiment140 dataset using TF-IDF features, outperforming SVM in certain configurations. Harjule *et al.* [5] further confirmed that feature selection plays a decisive role, as LR and NB models exhibited varying performance depending on the embedding strategy.

While these traditional approaches offer advantages such as low computational cost, interpretability, and ease of deployment, their reliance on shallow features limits their ability to capture contextual semantics and long-range dependencies in text. Moreover, most of these studies focus on a single feature extraction technique or classifier, lacking a comprehensive comparative framework across multiple representations and datasets. To overcome the limitations of feature engineering, deep learning approaches introduced neural architectures capable of learning distributed representations automatically. Convolutional neural networks (CNNs), RNNs, LSTM, and BiLSTM have been widely adopted for sentiment analysis.

Kim and Jeong [6] demonstrated that CNNs are effective in extracting local n-gram features from text, particularly when combined with pretrained embeddings. Dang *et al.* [7] conducted a comparative study using CNN, RNN, and DNN models, revealing that word embeddings substantially outperform TF-IDF representations, especially for sequential models. Similarly, Vimali and Murugan [8] showed that BiLSTM architectures better capture contextual dependencies, achieving higher accuracy than unidirectional LSTM models.

Despite these improvements, deep learning models still suffer from several limitations. Their performance is highly sensitive to embedding quality, training data size, and hyperparameter tuning. Furthermore, most deep learning studies evaluate models in isolation, without systematically comparing them against classical ML baselines under consistent experimental settings.

The introduction of transformer-based models marked a paradigm shift in sentiment analysis. BERT [9] and its variants, including RoBERTa [10] and distilled BERT (DistilBERT) [11], leverage self-attention mechanisms to capture deep contextual representations. Multiple studies have reported substantial performance gains using these architectures. For instance, BERT achieved state-of-the-art results on Twitter sentiment classification tasks, significantly outperforming CNN and BiLSTM-based models [12].

However, transformer models are computationally expensive and may not always generalize optimally to short and noisy social media texts. To address these challenges, recent research has explored hybrid architectures that combine transformers with recurrent or convolutional networks. Tan *et al.* [13] proposed a RoBERTa-gated recurrent unit (GRU) model that balances contextual richness with computational efficiency. Rahman *et al.* [14] and Tan *et al.* [15] demonstrated that integrating RoBERTa with BiLSTM or LSTM layers enhances sequential modeling and improves classification accuracy. Additionally, Yan *et al.* [16] introduced Emotion-RGC Net, which integrates RoBERTa with graph neural networks (GNN) and conditional random fields (CRF) to capture relational and emotional dependencies in social media.

Recent scholarship has increasingly favored hybrid architectures that synergize the complementary strengths of convolutional and recurrent networks to enhance feature representation. He [17] proposed a BERT-CNN-BiLSTM model that successfully integrates local feature extraction with global semantic representation, while Xu and Song [18] demonstrated that such hybrid configurations consistently outperform standalone transformer or convolutional models. However, maximizing the potential of these complex architectures requires precise hyperparameter tuning—a challenge effectively addressed by meta-heuristic algorithms. As elucidated by Yang [19] and Slowik and Kwasnicka [20], nature-inspired methodologies like particle swarm optimization (PSO) efficiently traverse vast search spaces to resolve non-convex problems where traditional deterministic methods fail. Ramezani *et al.* [21] specifically highlighted PSO's superior convergence capabilities in high-dimensional spaces, making it ideal for optimizing critical parameters, such as learning rate and dropout, a complexity extensively investigated by Feurer and Hutter [22] and Ramamohan *et al.* [23]. The necessity of this approach is further validated by Prabhu *et al.* [24] and Londhe and Kumari [25], whose work on bio-inspired optimization (such as social eagle optimization) in sentiment analysis confirms that integrating robust algorithmic tuning with hybrid deep learning models significantly enhances both classification accuracy and convergence stability.

Despite their strengths, existing studies suffer from key limitations: they rarely compare models across traditional, deep learning, transformer, and hybrid paradigms within a unified framework, and often evaluate feature representations in isolation. Consequently, the relative effectiveness of hybrid models on large-scale Twitter datasets remains underexplored. To bridge these gaps, this study systematically benchmarks classical ML, deep learning, transformer, and hybrid architectures on the Sentiment140 dataset under identical experimental conditions. By evaluating diverse feature extraction strategies and model paradigms, this work clarifies performance trade-offs to guide model selection for real-world applications.

3. DATASET AND METHODS

3.1. Dataset description

In this study utilized the Sentiment140 dataset, which is a prominent resource for sentiment analysis in NLP. This dataset contains 1,600,000 tweets extracted using the Twitter API and is available for public use. Each tweet has been annotated to indicate its sentiment polarity, with values representing negative (0), neutral (2), and positive (4) sentiments, making it suitable for sentiment detection tasks. The official link to the dataset and its generation resources is available on Kaggle. This repository provides access to the dataset files as well as supporting information for researchers and practitioners. The foundational paper detailing the methodology used for its creation can be found in [26].

3.2. Methods

The intellectual exploration of sentiment classification through an ML perspective requires a robust and multifaceted methodology. Consequently, the proposed method investigation unfolded in a series of well-defined steps, each carefully designed to fulfill a specific purpose within a broader research framework. This systematic workflow ensures that the data is meticulously prepared, analyzed, and validated at every stage.

3.2.1. Data preprocessing

The sentiment analysis process involves several key steps. Initially, data cleaning and feature extraction are carried out as part of the preprocessing phase, followed by the employment of various deep learning models during the training phase, with detailed results to be provided later. Text cleaning is a crucial step that eliminates non-informative words and elements that could hinder the effectiveness of sentiment analysis. Typically, text data contains whitespace, punctuation, and stop words. The cleaning process includes removing Twitter-specific elements such as retweets (RTs), @mentions, #hashtags, and links; applying stemming or lemmatization; converting all text to lowercase; stripping out non-letter characters, including numbers; eliminating English stop words and punctuation; reducing excessive whitespace; and decoding hypertext markup language (HTML) into plain text. Depending on the dataset, specific processing

methods can aid in model development. For example, with the Sentiment140 dataset, non-essential columns—specifically "id," "date," "query_string," and "user" were discarded, and sentiment labels were reclassified from {0, 4} to {0, 1} (where 1 represents positive and 0 represents negative). After the datasets were cleaned, sentences were tokenized into individual words and lemmatized to their base forms, allowing conversion into feature vectors represented by continuous real numbers using various methods.

3.2.2. Machine learning methods

With the data landscape prepared, it was time to implement the leading ML algorithms. Four algorithms, each recognized for their unique strengths and inherent challenges, were selected for the sentiment analysis task. Evaluating these diverse models' side-by-side will highlight which approach best handles the nuances of the dataset.

i) Support vector machine

SVM is a prominent supervised classifier, recognized for its effectiveness in both regression and classification applications [27]. At its core, the algorithm operates by constructing hyperplanes in a high-dimensional space to separate data classes with the maximum margin [28]. To preprocess the raw tweets from the Sentiment140 dataset, we implemented a cleaning pipeline using the natural language toolkit (NLTK) library. This involved removing uniform resource locator (URLs), special characters, and numbers, as well as lowercasing and tokenizing the text. This study reduced noise by filtering out standard English stop words and used the TF-IDF technique for feature representation, limiting the maximum features to 5,000 for efficiency.

This study selected SVM as the core classifier due to its robustness in high-dimensional spaces, implementing it with the Scikit-learn library in Python. A linear kernel (kernel = 'linear') was chosen, given its superior speed and accuracy for this task. For the experimental setup, the dataset was divided using a stratified split ratio of 80% for training and 20% for testing, ensuring evaluation on unseen data while setting a random seed (random_state = 42) for reproducibility [29].

ii) Decision tree

The decision tree is a widely utilized classification technique in data mining [30], selected for this study due to its interpretable hierarchical structure where internal nodes represent attribute events and edges denote decision conditions. To effectively classify the high-dimensional TF-IDF vectors derived from tweets, this study implemented C4.5, which optimizes the standard ID3 approach by employing information entropy and gain ratio for attribute splitting [31]. This architecture offers distinct advantages, including robust handling of diverse data types and rapid inference speed during the testing phase. To ensure statistical reliability and mitigate overfitting, this study applied post-pruning strategies based on confidence factors and validated the model using a 10-fold cross-validation scheme [32].

iii) Naive Bayes

Based on the principles of probability, the NB classifier utilizes Bayes theorem [33], to predict sentiment polarity. While the standard algorithm operates under the "naïve" assumption of feature independence, this study specifically adopted the MNB variant for this study. This adaptation is particularly effective for text classification tasks, as it is optimized to handle the discrete distribution of TF-IDF feature vectors generated during preprocessing. To address the "zero-frequency" problem—where a term in the test set is absent from the training set—this study applied Laplace smoothing (α = 1.0), ensuring the stability of conditional probabilities. The implementation was executed using the Scikit-learn framework, with an 80:20 train/test split to validate the model's generalization capabilities. The algorithm calculates the posterior probability with (1).

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)} \quad (1)$$

In (1), $P(C|X)$ denotes the posterior probability of target class C given observation X. The term $P(X|C)$ represents the likelihood of predictor X conditioned on class C, $P(C)$ corresponds to the initial prior probability of class C, and $P(X)$ acts as the marginal probability or evidence of predictor X.

iv) Logistic regression

LR is a prominent discriminative method in classification, utilized here to assess the relationship between the binary dependent variable (sentiment) and the independent features (word vectors) [34]. Unlike complex "black-box" models, it offers a transparent probabilistic view. For this study, the model was adapted to process sparse TF-IDF vectors. The core mechanism involves the Sigmoid function (σ), which maps the weighted sum of input features to a probability value between 0 and 1. The mathematical representation is defined as (2).

$$\sigma(z) = \frac{1}{1+e^{-z}} \quad (2)$$

Where z represents the linear combination of weighted inputs $z = w^T x + b$.

For experimental implementation and tuning to classify the tweets, this study applied a decision threshold of 0.5: if the predicted probability $\sigma(z) > 0.5$, the tweet is classified as positive (1); otherwise, it is negative (0). The implementation was carried out using the scikit-learn framework with L2 regularization (Ridge) to penalize large coefficients and prevent overfitting on the high-dimensional textual data. Consistent with the methodology described in [5] the dataset was divided into 80% for training and 20% for testing.

3.2.3. Deep learning methods

The moment had arrived to deploy the top deep learning models. Five distinct architectures, each celebrated for their specific advantages and underlying complexities, were chosen for the sentiment analysis task. These selected frameworks will process the text data to evaluate and compare their predictive performance.

i) Convolutional neural network

Although originally designed for computer vision tasks [35], CNNs have proven highly effective in NLP when formulated as one-dimensional architectures. In this study deploy a 1D-CNN tailored for text classification to extract local feature patterns (n-grams) through sliding filter operations. The pipeline begins with an embedding layer initialized via pre-trained Word2Vec vectors to capture underlying semantic relationships. This is followed by a 1D convolutional layer featuring 128 filters with a kernel size of 5 and rectified linear unit (ReLU) non-linear activation. To retain the most dominant sentiment indicators while reducing feature dimensionality, a global max pooling layer is applied [36]. Built in Keras with a TensorFlow backend, the network is trained using the adaptive moment estimation (Adam) optimizer and binary cross-entropy loss over 20 epochs with a batch size of 128. A dropout rate of 0.5 is introduced before the final dense output layer to prevent overfitting. Finally, following standard benchmark protocols, the Sentiment140 dataset is divided into an 80:20 train–test split to ensure reproducible evaluation.

ii) Recurrent neural network

Feedforward neural networks typically process data straightforwardly, but to capture the dependencies between data points, a different architecture is needed. RNNs are a modified version of traditional neural networks designed specifically for sequential data [37]. RNNs can retain information from previous inputs, allowing them to influence the generation of subsequent outputs in a sequence [38].

The implementation and specific tuning of the model were carried out using the Keras framework. The architecture consists of several key components. Firstly, an embedding layer converts input tokens into dense vectors using pre-trained Word2Vec weights. Next, a SimpleRNN layer, configured with 128 hidden units and a tanh activation function, models sequential patterns in the data. To mitigate overfitting, which is a common issue in high-dimensional text data, a dropout layer with a rate of 0.5 is applied before the output layer. This final layer is a dense layer with a Sigmoid activation function that computes the binary sentiment probability. Regarding the experimental protocol, the training process utilized the Adam optimizer and the binary cross-entropy loss function over 20 epochs with a batch size of 128. Consistent with the study's methodology, the dataset followed an 80:20 train/test split.

iii) Bi-directional long short-term memory

To effectively address the vanishing gradient issue and capture the full semantic context of tweets, a BiLSTM [39] network was implemented. While standard LSTMs utilize a gating mechanism—comprised of input, forget, and output gates—to learn long-term dependencies, they are limited to processing sequences in a single direction. The proposed implementation advances this by utilizing two independent LSTM layers: one processing the input sequence forward and the other backward. This dual structure allows the network to preserve information from both past and future contexts simultaneously, generating a more robust feature representation for sentiment classification.

The overall architecture of the proposed model is illustrated in Figure 1, highlighting the flow of data from word embeddings through the recurrent units to the final classification step. Built using the Keras framework with a TensorFlow backend, the pipeline utilizes Word2Vec embeddings to transform raw tokens into dense vector representations before feeding them into forward and backward LSTM units of 128 units each. To prevent overfitting on the high-dimensional textual data, a dropout layer with a rate of 0.5 was applied prior to the final fully connected softmax activation layer. For the experimental protocol, the training was optimized using stochastic gradient descent (SGD) or Adam, minimizing the error rate over 20 epochs. The dataset was partitioned using an 80:20 train/test split ratio to ensure statistical validity.

iv) Bidirectional encoder representations from transformers

BERT is a language model introduced by Google researchers in October 2018; it leverages an encoder-only transformer framework and self-supervised learning to encode text as vector sequences. Its introduction marked a significant breakthrough in NLP, and by 2020, it had become a widely used benchmark in NLP research [40]. BERT's pre-training process involves masked token prediction and next

sentence prediction (NSP), enabling it to capture deep context-dependent word representations. As illustrated in Figure 2, the architecture tokenizes the input sequence—flanked by special [CLS] and [SEP] tokens—and passes it through multiple encoder layers to a feed-forward classification head. This end-to-end framework allows the model to effectively process textual sequences for downstream tasks such as sentiment analysis.

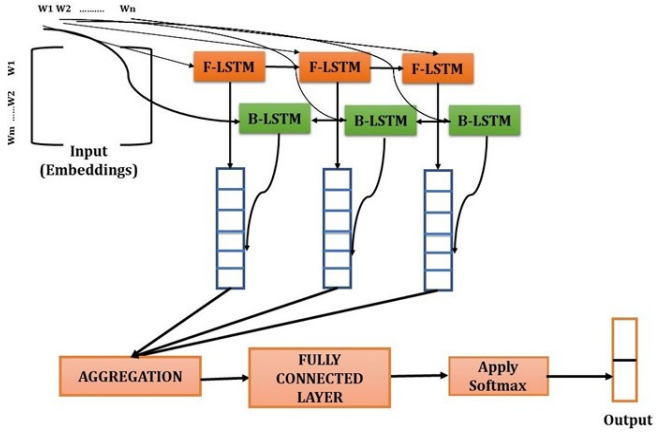


Figure 1. BiLSTM-based sentiment analysis [8]

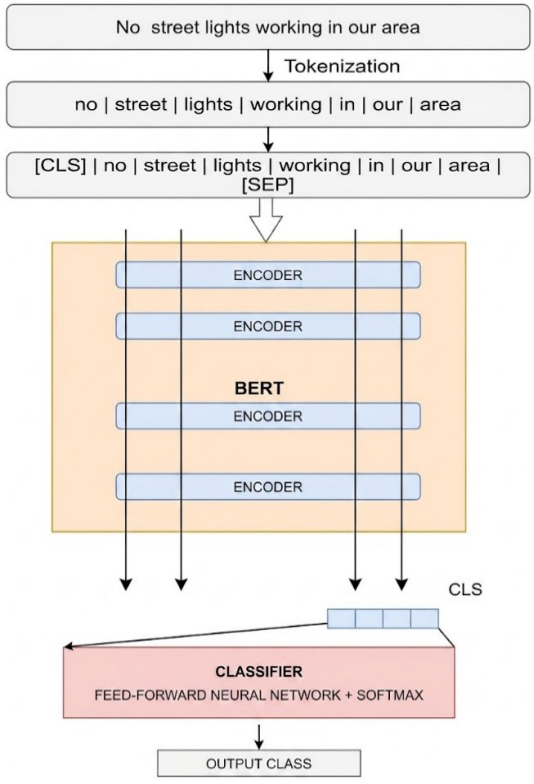


Figure 2. Architecture of the BERT model [41]

Implementation and hyperparameter tuning: the model was implemented using the Keras framework with a TensorFlow backend, executed on a high-performance computing (HPC) environment equipped with an NVIDIA Tesla V100 GPU (for the Sentiment140 dataset). The optimization strategy relied on the Adam optimizer, selected for its efficiency in handling sparse gradients. The loss function was adapted according to the dataset: binary cross-entropy for binary classification (Sentiment140). The model was fine-tuned over multiple epochs, monitoring training and validation loss to prevent overfitting [41].

v) Robustly optimized BERT pretraining approach

RoBERTa is an improved variant of BERT introduced by Facebook artificial intelligence (AI) in 2019, It enhances BERT's pretraining process by removing the NSP objective, using dynamic masking, increasing training data and batch sizes, and fine-tuning key hyperparameters. These optimizations allow RoBERTa to achieve superior performance on various NLP tasks compared to BERT [42]. Technical implementation the model was integrated into the proposed experimental pipeline using Keras and TensorFlow, ensuring methodological consistency with the other approaches. The "fine-tuning" process utilized the Adam optimizer to adjust network weights. Particular attention was paid to data preprocessing before input into the RoBERTa tokenizer, including lowercase conversion, removal of stop words and punctuation, and the application of lemmatization to reduce noise inherent in raw tweets.

Performance and comparison RoBERTa's performance was measured via accuracy on unseen test data. On the Sentiment140 dataset, the model achieved a training accuracy of 80.3% and a test accuracy of 78.20%. Although theoretically more robust, RoBERTa displayed lower performance than BERT (92.76%) on this specific task. This indicates that despite robust optimization and dynamic masking, the base architecture of BERT remains more suitable or easier to fine-tune for this specific volume and type of data, based on the recorded accuracy metrics.

3.2.4. Hybrid algorithms

Hybrid approaches integrate both lexicon-based and ML techniques. Sentiment lexicons typically play a central role in most of these strategies. By combining predefined dictionaries with learning algorithms, these models achieve higher accuracy across diverse text formats.

i) Convolutional neural network-bidirectional long-short term memory

A hybrid CNN-BiLSTM architecture was designed to synergize the structural strengths of two distinct deep learning models for sentiment analysis. As illustrated in Figure 3, the process initiates with a word embedding layer utilizing Word2Vec to transform raw tweets into dense semantic vectors before feeding them into a CNN feature extraction block. These vectors pass through convolutional filters and a max-pooling operation to extract salient local n-gram features while effectively reducing spatial dimensionality.

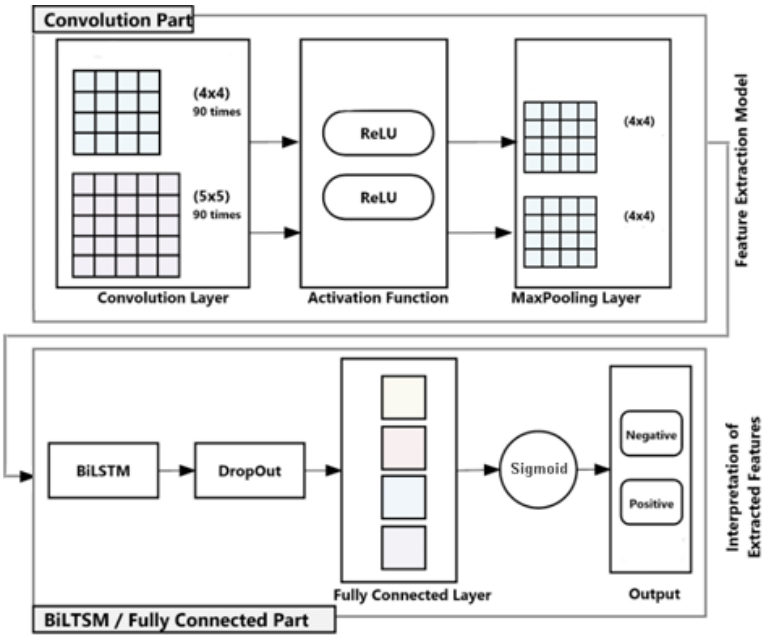


Figure 3. CNN-BiLSTM layer [12]

The local features extracted by the convolutional stage are then transmitted to a BiLSTM layer for sequential context interpretation. This recurrent layer processes information in both past and future directions, enabling the model to capture long-term dependencies and complete context that standalone CNNs cannot achieve. Executed in Python using the Keras framework with a TensorFlow backend, the pipeline incorporates ReLU activations for hidden layers, a dropout layer to mitigate overfitting, and a

sigmoid activation function at the output layer for binary classification. The model was trained and evaluated on a massive corpus comprising 1.6 million tweets partitioned according to a standard 80:20 train/test split ratio. Performance was rigorously assessed using four key metrics: accuracy, precision, recall, and F1-score.

ii) Bidirectional encoder representations from transformers-bidirectional long-short term memory

In order to bridge the gap between context-aware embeddings and sequential dependency learning, this study constructed a hybrid model that combines BERT and BiLSTM networks [43]. This innovative architecture utilizes BERT-Base, which consists of 12 layers and 768 hidden units, as a dynamic feature extractor rather than merely a static embedding layer. The adaptation process involves feeding the context-rich token embeddings produced by BERT into the BiLSTM layer. Unlike traditional pooling methods, this integration preserves the spatial context provided by BERT, while the BiLSTM effectively captures the temporal dependencies across a sequence length of 128 tokens. This setup is particularly adept at understanding sarcasm and intricate syntactic structures found in tweets.

The implementation of the model was carried out using TensorFlow and Keras. To fine-tune the substantial transformer weights along with the recurrent layers, this study employed the Adam optimizer, maintaining a strictly controlled learning rate of 2e-5 and a batch size of 32. Moreover, the architecture incorporates a dropout layer with a rate of 0.3, situated between the BiLSTM output and the final dense layer, to reduce the risk of overfitting, particularly on the Sentiment140 dataset. The training process spanned three epochs, a choice made to balance the need for quick convergence with the necessity of good generalization performance, especially given the computational demands of the transformer backbone.

For the evaluation protocol, a strict 80:20 train/test split was applied to the processed dataset. The efficacy of the hybrid model was gauged using various metrics including accuracy, precision, recall, and F1-score. Notably, the BERT-BiLSTM configuration exhibited a marked improvement in performance by merging the bidirectional context of transformers with the sequential memory capabilities of LSTMs, which resulted in enhanced classification stability compared to models operating independently.

iii) Robustly optimized BERT pretraining approach-gated recurrent unit

The proposed model integrates the contextual understanding of RoBERTa with the efficiency of a GRU network. RoBERTa acts as a feature extractor, generating rich contextual embeddings from text, while the GRU layer captures sequential dependencies more efficiently than traditional LSTM networks. The architecture begins with data preprocessing and tokenization using the RoBERTa tokenizer, followed by passing the resulting input identifier (IDs) and attention masks through a pretrained RoBERTa model. The contextual embeddings produced are then fed into a GRU layer, supplemented by a dropout layer to prevent overfitting, leading to a fully connected layer that maps feature to output classes, with a softmax function providing sentiment class probabilities.

To further enhance the model, we employed a word embedding-based data augmentation strategy, leveraging the natural language processing augmentation (NLPAug) library for synonym substitution and random word insertion to enrich the training dataset and prevent overfitting. The model was built using the PyTorch framework, with the Sentiment140 dataset split into 80:20 for training and testing. This study utilized the AdamW optimizer to minimize the cross-entropy loss, with a learning rate of 2e-5 and a batch size of 32, training over four epochs with early stopping. The model demonstrated impressive performance, achieving an accuracy of 89.59% on the test set, while being rigorously evaluated using metrics like accuracy, precision, recall, and F1-score.

iv) RoBERTa-convolutional neural networks-bidirectional long-short term memory

The RoBERTa-CNN-BiLSTM architecture is designed to extract hierarchical linguistic features [43]. This model modifies the standard pipeline by introducing a CNN layer immediately after the RoBERTa embeddings. This specific tuning allows the model to first capture local n-gram features (spatial correlations) through convolution filters before passing the data to the BiLSTM for long-range dependency modeling. RoBERTa was chosen for its robust optimization—such as dynamic masking and the removal of NSP—which helps provide a cleaner input signal to the downstream convolutional and recurrent layers. Executed within the TensorFlow/Keras framework, this architecture requires careful regularization due to its depth. This study applied L2 regularization and a dropout rate of 0.3 to prevent the vanishing gradient problem and overfitting. The model utilizes a global max pooling operation after the CNN layer to down-sample dimensions while retaining the most salient sentiment features. The training employed the categorical cross-entropy loss function, optimized via Adam with a learning rate of 2e-5, ensuring that the gradients from the recurrent layers effectively updated the pre-trained RoBERTa weights.

The model was evaluated on the unseen 20% testing split of the Sentiment140 corpus. By synergizing spatial feature extraction (CNN) with temporal modeling (BiLSTM) and robust embeddings (RoBERTa), this architecture aimed to maximize the F1-score and accuracy. The evaluation metrics confirmed that this tri-stage approach effectively handles the noise and informality of Twitter data better than single-stage architectures.

v) Emotion-RGC Net

Emotion-RGC Net is a deep learning framework specifically designed for emotion recognition on social media. This model stands out for its ability to address the complex, multi-layered aspects of emotion analysis by integrating a pre-trained language model (RoBERTa), a GNN [44], and (CRF) [45], as illustrated in Figure 4, the architecture comprises three key components: RoBERTa, which extracts emotional features from text; GNN, which models user interactions and captures emotion propagation; and CRF, which refines and optimizes the final emotion classification [16].

To address emotion propagation in social networks, Emotion-RGC Net combines text analysis with graph-based relational modeling. RoBERTa extracts context-aware semantic features that are treated as node representations in a social graph. These features are then processed by the GNN, which captures emotion propagation through message passing among neighboring users. The model was implemented in PyTorch with module-specific hyperparameter tuning. The GNN consists of two graph convolutional network (GCN) layers with 64 hidden dimensions, mean aggregation, and L2 regularization. It uses a learning rate of 1e-3, while the RoBERTa backbone is fine-tuned with a learning rate of 2e-5 and a batch size of 32.

For final prediction, the CRF layer applies Viterbi decoding with maximum likelihood estimation (MLE). The entire network was trained end-to-end for 10 epochs using the AdamW optimizer with step-wise learning rate decay. Performance was evaluated on the Sentiment140 and emotion datasets using accuracy, precision, recall, and F1-score.

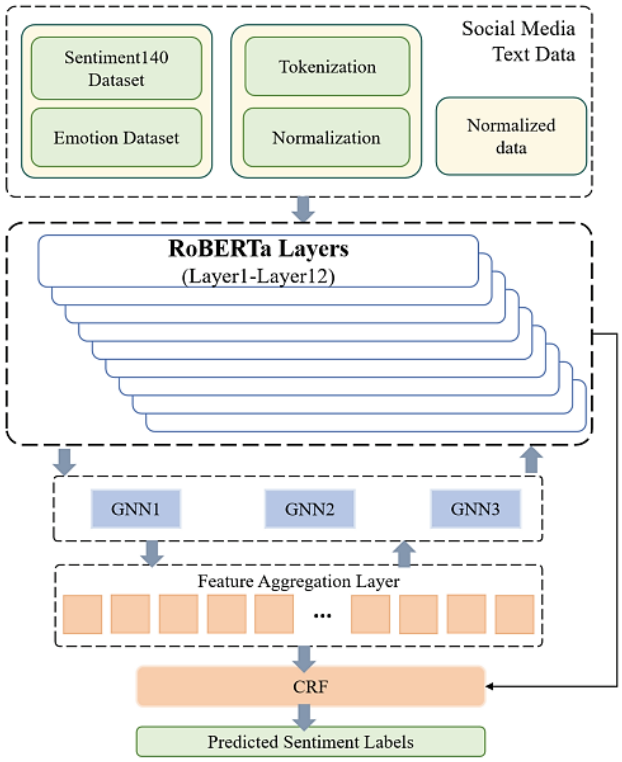


Figure 4. The architecture of emotion-RGC Net [16]

4. EXPERIMENTAL RESULTS

In the previous section (section 3) discussed the Sentiment140 dataset, including its description and the preprocessing steps applied. This study also presented a summary in Table 1 providing an overview of the sentiment analysis datasets used. In this section analyze the experimental findings to highlight the relative strengths and efficacy of each architecture.

Table 1. Summary of sentiment analysis dataset

Dataset	Classes	Positive	Negative	Total
Sentiment140	2	800,000	800,000	1,600,000

In this section present and analyze the results of various sentiment analysis models. This study also provide a comparison table that highlights the performance of these models when combined with different feature extraction techniques such as TF-IDF, Word2Vec, GloVe, bag of words, and word embeddings [46]. The accuracy and performance metrics corresponding to this experiment across various traditional ML algorithms are tabulated in Table 2. These empirical results highlight significant variations in classification efficacy depending on the choice of model and feature representation. Consequently, this comparative analysis provides clear insights into the strengths and limitations of classical approaches on short-text social media data.

As presented in Table 2 and illustrated in Figure 5, several ML algorithms were evaluated on the Sentiment140 dataset using primarily TF-IDF-based feature representations. Among the evaluated methods, the KNN classifier with TF-IDF features achieves the highest accuracy at 88%, along with strong F1-score, recall, and precision values indicating robust classification performance. The NB model also demonstrates competitive results, reaching an accuracy of 85% and achieving the highest precision (86%) among the reported approaches.

Table 2. Summary of ML methods

Literature	Methods	Features	Accuracy (%)	F1-score (%)	Recall (%)	Precision (%)
Patel *et al.* [29]	SVM	TF-IDF	84	84	84	85
Rathi *et al.* [32]	Decision tree	TF-IDF	84	84	84	84
Jasy *et al.* [4]	NB	TF-IDF	85	85	84	86
Harjule *et al.* [5]	LR	TF-IDF	76.44	76	76	76
Jasy *et al.* [4]	KNN	TF-IDF	88	88	86	88
Harjule *et al.* [5]	MNB	TF-IDF	76.52	76	77	77
Hourrane *et al.* [47]	AdaBoost	TF-IDF, n-gram	68.62	-	-	62.33
Hourrane *et al.* [47]	RidgeClassifier	TF-IDF, n-gram	76.84	-	-	70.92

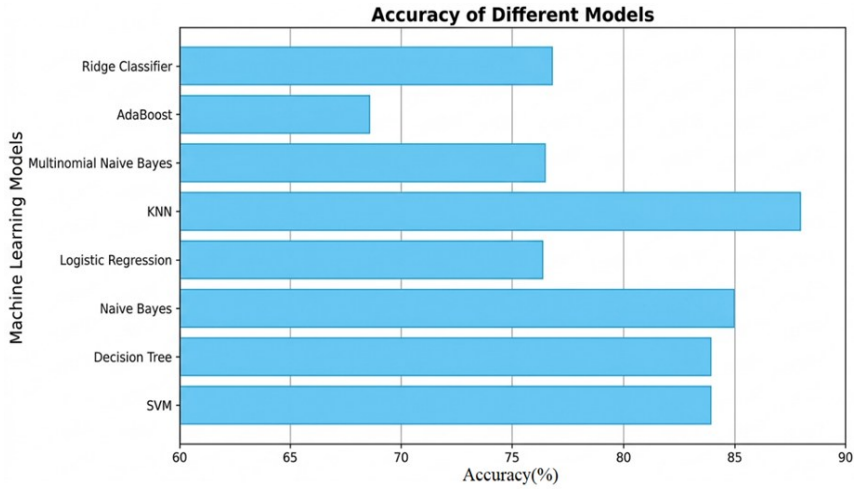


Figure 5. Performance of ML methods on the Sentiment140 dataset

Both SVM and decision tree classifiers obtain comparable and stable performances, each achieving an accuracy and F1-score of 84%. In contrast, LR and Ridge classifier models exhibit moderate effectiveness, with accuracies of 76.44% and 76.84%, respectively. Although ridge classifier incorporates TF-IDF and n-gram features, its performance remains limited compared to instance-based and probabilistic models.

The MNB model records an accuracy of 76.52%, which is lower than that of standard NB, suggesting that the choice of probabilistic formulation influences performance when combined with TF-IDF features. Finally, adaptive boosting (AdaBoost), despite leveraging TF-IDF and n-gram representations, yields the lowest performance, with an accuracy of 68.62%, highlighting the limitations of ensemble boosting methods for this dataset. Overall, these results confirm that feature representation and classifier choice play a crucial role in sentiment classification, with instance-based (KNN) and probabilistic (NB) approaches demonstrating superior effectiveness on short and informal texts such as tweets in the Sentiment140 dataset.

Table 3 provides a comparative evaluation of deep learning and transformer-based models on the Sentiment140 dataset using different feature representations. Among all the evaluated approaches, BERT

with bidirectional contextual embeddings achieves the highest accuracy (92.76%), clearly outperforming the other models. This result underscores the effectiveness of pretrained transformer architectures in capturing rich contextual and semantic information, which is particularly beneficial for sentiment classification of short and informal texts such as tweets.

Table 3. Summary of deep learning methods

Literature	Methods	Features	Accuracy (%)	F1-score (%)	Recall (%)	Precision (%)
Dang *et al.* [7]	CNN	TF-IDF/word embedding	76.68/80.06	75.93/80.06	74.07/80.08	74.07/80.08
Dang *et al.* [7]	RNN	TF-IDF/word embedding	56.95/82.81	64.04/82.96	77.73/83.69	77.73/83.69
Dang *et al.* [7]	DNN	TF-IDF/word embedding	76.49/78.81	76.38/78.87	75.77/79.09	75.77/79.09
Harjule *et al.* [5]	LSTM	GloVe	82	82	84	81
Vimali and Murugan [8]	BiLSTM	Word2Vec+SVD	78.53	-	-	-
Joshy and Sundar [41]	BERT	Bidirectional contextual embeddings	92.76	-	-	-
Joshy and Sundar [41]	RoBERTa	Dynamic contextual embeddings	78.2	-	-	-
Joshy and Sundar [41]	DistilBERT	Compressed contextual embeddings	77.1	-	-	-

Recurrent neural models also demonstrate competitive performance. The RNN architecture exhibits the most pronounced improvement when transitioning from TF-IDF features (56.95%) to word embeddings (82.81%), highlighting the strong dependency of sequence-based models on dense semantic representations. Similarly, BiLSTM and LSTM models achieve solid performance, with BiLSTM reaching an accuracy of 78.53% using Word2Vec combined with SVD, while LSTM with GloVe embeddings attains balanced precision, recall, and F1-score values (81–84%), confirming their ability to model long-range dependencies in textual data.

The GRU-based models reported in the literature perform competitively, whereas CNN and DNN architectures show moderate overall effectiveness. As illustrated in Figure 6, CNN performance improves from 76.68% with TF-IDF to 80.06% when word embeddings are employed, reflecting a clear jump in feature extraction capacity. Likewise, DNN accuracy increases from 76.49% to 78.81%, further emphasizing the critical role feature representation plays across deep learning paradigms.

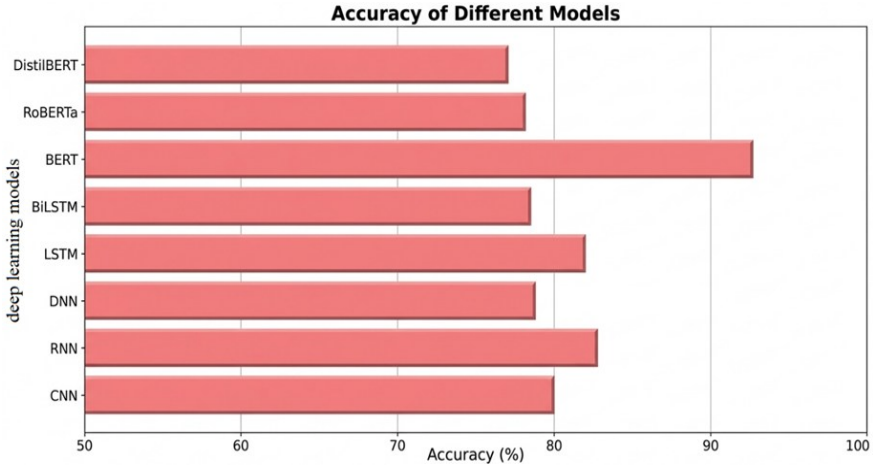


Figure 6. Performance of deep learning methods on the Sentiment140 dataset

In contrast, RoBERTa and DistilBERT achieve relatively lower accuracies of 78.2% and 77.10%, respectively. This suggests that not all transformer variants generalize equally well to sentiment analysis on short-text datasets, despite their architectural similarities. Overall, these findings confirm that pretrained transformer models, especially BERT, offer a substantial advantage in contextual understanding, while recurrent architectures such as RNN, LSTM, and BiLSTM remain strong alternatives when combined with appropriate word embedding representations.

Table 4 presents a comparative analysis of hybrid, ensemble, and transformer-enhanced models evaluated on the Sentiment140 dataset using diverse feature representations. Among the reported approaches,

models combining transformer architectures with RNN demonstrate the strongest overall performance. In particular, the RoBERTa–GRU and RoBERTa–LSTM models achieve high accuracies of 89.59% and 89.70%, respectively, accompanied by consistently high F1-score, recall, and precision values, indicating robust and well-balanced sentiment classification.

As illustrated in Figure 7, the Emotion-RGC Net model records a highly competitive accuracy of 89.7% with balanced 90% precision and recall metrics. This demonstrates how effectively bidirectional transformer structures capture fine-grained emotional and contextual nuances in short texts. Meanwhile, the classical ensemble combining NB, SVM, and maximum entropy (ME) achieves an impressive 88% accuracy, showing that well-engineered unigram, bigram, and lexicon features can closely rival deep transformer networks.

Table 4. Summary of hybrid methods

Literature	Methods	Features	Accuracy (%)	F1-score (%)	Recall (%)	Precision (%)
Varshney *et al.* [48]	LR + NB + SGD	TF-IDF	79.95	80	80	80
Iqbal *et al.* [49]	NB, SVM, ME	Unigram, Bigram, Lexicon	88	89	90	88
Tan *et al.* [13]	RoBERTa-GRU	Word embedding, GloVe	89.59	90	90	90
Tan *et al.* [15]	RoBERTa-LSTM	Word embedding, GloVe	89.70	90	90	90
Bouassida and Mezali [43]	BERT-BiLSTM	Contextual Embeddings	79	80	79	79
Bharal and Krishna [12]	CNN-BILSTM	Word2Vec	78.86	78	78	79
Derbentsev *et al.* [50]	CNN-LSTM	Word2Vec, GloVe	78.37	78.23	77.47	78.98
Bouassida and Mezali [43]	RoBERTa-CNN-BiLSTM	Contextual Embeddings	77	77	76	80
Rahman *et al.* [14]	RoBERTa-BiLSTM	Word embedding	82.25	82.25	82.5	82.25
Bouassida and Mezali [43]	DistilBERT-BiLSTM	Efficient Embeddings	81	81	80	82
Yan *et al.* [16]	Emotion-RGC Net	bidirectional transformer structure	89.7	88	90	90

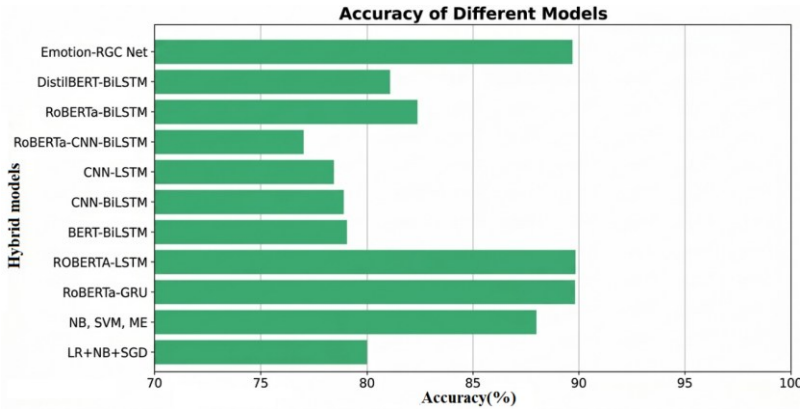


Figure 7. Performance of hybrid methods on the Sentiment140 dataset

In contrast, classical ensemble methods such as LR + NB + SGD with TF-IDF features achieve moderate performance, with an accuracy of 79.95%, while hybrid CNN–RNN architectures, including CNN–BiLSTM and CNN–LSTM, report accuracies in the range of 78–79%, reflecting limited gains from convolutional–recurrent combinations when relying on static word embeddings. Models integrating BERT or RoBERTa with BiLSTM layers exhibit mixed performance. While RoBERTa–BiLSTM and DistilBERT–BiLSTM achieve accuracies of 82.25% and 81%, respectively, BERT–BiLSTM and RoBERTa–CNN–BiLSTM yield lower accuracies of 79% and 77%, suggesting that additional recurrent or convolutional layers do not always translate into performance improvements over the base transformer representations. Overall, these results indicate that hybrid architectures leveraging pretrained transformers combined with lightweight recurrent components can effectively enhance sentiment classification performance on the Sentiment140 dataset. However, the observed variability across hybrid designs highlights the importance of model architecture selection and feature representation in achieving optimal performance.

5. DISCUSSION

This study provides a comprehensive comparative analysis of classical ML, deep learning, hybrid, and transformer-based models for sentiment analysis on the Sentiment140 dataset. The results reveal clear

performance differences across these model families. These findings highlight the impact of architectural design and feature representation on the effectiveness of sentiment classification.

5.1. Comparative interpretation of model performance

Among all evaluated approaches, transformer-based models, particularly BERT, achieve the highest performance, with an accuracy of 92.76%, confirming their strong ability to capture deep contextual and semantic information. This result aligns with recent findings in the literature and reinforces the dominant role of pretrained transformers in sentiment analysis tasks involving short and informal texts such as tweets. However, the results also demonstrate that hybrid architectures remain highly competitive. Models such as RoBERTa–LSTM, RoBERTa–GRU, and ensemble approaches combining NB, SVM, and ME achieve accuracies close to transformer-only models (around 89–90%). This indicates that combining contextual embeddings with recurrent structures or statistical classifiers can effectively balance representational power and model complexity.

Sequential deep learning models further illustrate the importance of feature representation. The RNN model, for instance, shows a substantial performance improvement when transitioning from TF-IDF features (56.95%) to word embeddings (82.81%), emphasizing that dense semantic representations are essential for sequence-based architectures. Similarly, CNN, LSTM, BiLSTM, and GRU models demonstrate moderate to strong performance when paired with word embeddings, though they remain under transformer-based and hybrid approaches. Interestingly, not all transformer variants perform equally well. RoBERTa and DistilBERT yield lower accuracies (approximately 77–78%), suggesting that architectural simplifications, pretraining strategies, and fine-tuning configurations can significantly influence performance on sentiment analysis tasks.

5.2. Error analysis

An examination of model behavior indicates that misclassifications predominantly occur in tweets containing sarcasm, implicit sentiment, informal language, abbreviations, or mixed emotions. Classical ML models relying on TF-IDF features struggle with contextual polarity shifts and negation, often misinterpreting sentiment when keywords appear without explicit emotional markers. Deep learning and hybrid models reduce these errors by leveraging sequential and contextual information; however, they still encounter difficulties with ambiguous or domain-specific expressions. Transformer-based models demonstrate improved robustness but remain challenged by sarcastic content and noisy user-generated text, which are common characteristics of the Sentiment140 dataset.

5.3. Limitations of the study

Despite the promising results, several limitations should be acknowledged. First, the evaluation is conducted exclusively on the Sentiment140 dataset, which focuses on binary sentiment classification and may not generalize to other domains or fine-grained sentiment tasks. Second, the analysis is restricted to English-language tweets, limiting the applicability of the findings to multilingual or low-resource language settings. Third, the study relies on offline experimental results reported in the literature, without real-time validation or deployment-based evaluation, preventing an assessment of model performance under dynamic, real-world conditions.

5.4. Discussion summary

Overall, the findings confirm that while transformer-based models such as BERT represent the current state of the art, hybrid and ensemble approaches continue to offer strong and competitive alternatives, particularly when computational efficiency or interpretability is a concern. The comparative analysis underscores that optimal sentiment classification performance emerges not from a single dominant paradigm, but from a careful alignment between model architecture, feature representation, and task characteristics. Future work should extend this evaluation to multilingual datasets, real-time environments, and more nuanced sentiment categories to further advance the robustness of sentiment analysis systems.

6. CONCLUSION AND FORWARD PATHWAYS

This study presented a comprehensive comparative review of classical ML, deep learning, hybrid, and transformer-based approaches for sentiment analysis on the Sentiment140 dataset. The findings confirm that pretrained transformer models, particularly BERT, achieve superior performance, owing to their strong contextual and semantic representation capabilities, while hybrid and ensemble models remain competitive alternatives when appropriately designed. The analysis also underscores the decisive role of feature representation, with word embeddings and contextual embeddings consistently outperforming traditional

TF-IDF features, especially in sequential and hybrid architectures. Looking forward, future research should extend this comparative evaluation to multilingual and domain-diverse datasets, explore real-time and streaming sentiment analysis, and investigate lighter and more efficient transformer variants to reduce computational cost. Additionally, incorporating fine-grained emotion categories, sarcasm detection, and domain adaptation strategies may further enhance robustness and generalization, paving the way for more reliable and scalable sentiment analysis systems in real-world social media environments.

FUNDING INFORMATION

We affirm on our honor that this study was conducted without any external funding.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Billal El Khattabi	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓
Hicham Omara	✓	✓		✓	✓	✓				✓		✓	✓	
Adnan Souri	✓	✓		✓	✓	✓				✓		✓		

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest

DATA AVAILABILITY

The data that support the findings of this study are openly available in Kaggle at https://www.kaggle.com/datasets/kazanova/sentiment140.

REFERENCES

[1] P. Chakriswaran, D. R. Vincent, K. Srinivasan, V. Sharma, C. Y. Chang, and D. G. Reina, "Emotion AI-driven sentiment analysis: a survey, future research directions, and open issues," *Applied Sciences*, vol. 9, no. 24, Dec. 2019, doi: 10.3390/app9245462.

[2] Y. G. Jung, K. T. Kim, B. Lee, and H. Y. Youn, "Enhanced naive Bayes classifier for real-time sentiment analysis with SparkR," in *2016 International Conference on Information and Communication Technology Convergence, ICTC 2016*, Oct. 2016, pp. 141–146, doi: 10.1109/ICTC.2016.7763455.

[3] M. Rezwanul, A. Ali, and A. Rahman, "Sentiment analysis on Twitter data using KNN and SVM," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 6, 2017, doi: 10.14569/ijacsa.2017.080603.

[4] M. D. H. Jasy, S. Al Hasan, M. I. K. Sagor, A. Noman, and J. M. Ji, "A performance evaluation of sentiment classification applying SVM, KNN, and Naive Bayes," in *2021 International Conference on Computing, Networking, Telecommunications & Engineering Sciences Applications (CoNTESA)*, Dec. 2021, pp. 56–60, doi: 10.1109/CoNTESA52813.2021.9657115.

[5] P. Harjule, A. Gurjar, H. Seth, and P. Thakur, "Text classification on Twitter data," in *2020 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things (ICETCE)*, Feb. 2020, pp. 160–164, doi: 10.1109/ICETCE48199.2020.9091774.

[6] H. Kim and Y.-S. Jeong, "Sentiment classification using convolutional neural networks," *Applied Sciences*, vol. 9, no. 11, Jun. 2019, doi: 10.3390/app9112347.

[7] N. C. Dang, M. N. M.-García, and F. De la Prieta, "Sentiment analysis based on deep learning: a comparative study," *Electronics*, vol. 9, no. 3, Mar. 2020, doi: 10.3390/electronics9030483.

[8] J. S. Vimali and S. Murugan, "A text-based sentiment analysis model using bi-directional LSTM networks," in *2021 6th International Conference on Communication and Electronics Systems (ICCES)*, Jul. 2021, pp. 1652–1658, doi: 10.1109/ICCES51350.2021.9489129.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *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)*, May 2019, vol. 1, pp. 4171–4186, doi: 10.18653/v1/N19-1423.

[10] Y. Liu *et al.*, "RoBERTa: a robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.

[11] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," *5th Workshop on Energy Efficient Machine Learning and Cognitive Computing - NeurIPS 2019*, 2020.

[12] R. Bharal and O. V. V. Krishna, "Social media sentiment analysis using CNN-BiLSTM," *International Journal of Science and Research*, vol. 10, no. 9, pp. 656–661, 2021, doi: 10.21275/sr21913110537.

[13] K. L. Tan, C. P. Lee, and K. M. Lim, "RoBERTa-GRU: a hybrid deep learning model for enhanced sentiment analysis," *Applied Sciences*, vol. 13, no. 6, Mar. 2023, doi: 10.3390/app13063915.

[14] M. M. Rahman, A. I. Shiplu, Y. Watanobe, and M. A. Alam, "RoBERTa-BiLSTM: a context-aware hybrid model for sentiment analysis," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 9, no. 6, pp. 3788–3805, Dec. 2025, doi: 10.1109/TETCI.2025.3572150.

[15] K. L. Tan, C. P. Lee, K. S. M. Anbananthen, and K. M. Lim, "RoBERTa-LSTM: a hybrid model for sentiment analysis with transformer and recurrent neural network," *IEEE Access*, vol. 10, pp. 21517–21525, 2022, doi: 10.1109/ACCESS.2022.3152828.

[16] J. Yan, P. Pu, and L. Jiang, "Emotion-RGC net: a novel approach for emotion recognition in social media using RoBERTa and graph neural networks," *PLOS ONE*, vol. 20, no. 3, Mar. 2025, doi: 10.1371/journal.pone.0318524.

[17] Y. He, "BERT-CNN-BiLSTM: a hybrid deep learning model for accurate sentiment analysis," in *2023 IEEE 5th International Conference on Power, Intelligent Computing and Systems (ICPICS)*, Jul. 2023, pp. 921–926, doi: 10.1109/ICPICS58376.2023.10235335.

[18] L. Xu and Y. Song, "Comparison of text sentiment analysis based on traditional machine learning and deep learning methods," in *2023 4th International Conference on Computer Engineering and Application (ICCEA)*, Apr. 2023, pp. 692–695, doi: 10.1109/ICCEA58433.2023.10135273.

[19] X.-S. Yang, *Nature-Inspired Optimization Algorithms*. Amsterdam, The Netherlands: Elsevier, 2014, doi: 10.1016/C2013-0-01368-0

[20] A. Slowik and H. Kwasnicka, "Evolutionary algorithms and their applications to engineering problems," *Neural Computing and Applications*, vol. 32, no. 16, pp. 12363–12379, Aug. 2020, doi: 10.1007/s00521-020-04832-8.

[21] F. Ramezani, J. Lu, and F. K. Hussain, "Task-based system load balancing in cloud computing using particle swarm optimization," *International Journal of Parallel Programming*, vol. 42, no. 5, pp. 739–754, Oct. 2014, doi: 10.1007/s10766-013-0275-4.

[22] M. Feurer and F. Hutter, "Hyperparameter optimization," in *Automated Machine Learning: Methods, Systems, Challenges*, Cham, Switzerland: Springer International Publishing, 2019, pp. 3–33, doi: 10.1007/978-3-030-05318-5_1.

[23] V. Ramamohan, S. Singhal, A. R. Gupta, and N. B. Bolia, "Discrete simulation optimization for tuning machine learning method hyperparameters," *Journal of Simulation*, vol. 18, no. 5, pp. 745–765, Sep. 2024, doi: 10.1080/17477778.2023.2219401.

[24] Prabhu, Vishnu, and S. Sujai, "Sentimental analysis of product rating," *International Academic Journal of Innovative Research*, vol. 9, no. 2, pp. 18–21, Jul. 2022, doi: 10.9756/IAJIR/V9I2/IAJIR0912.

[25] D. Londhe and A. Kumari, "Multilingual sentiment analysis using the social eagle-based bidirectional long short-term memory," *International Journal of Intelligent Engineering and Systems*, vol. 15, no. 2, pp. 479–493, Apr. 2022, doi: 10.22266/ijies2022.0430.43.

[26] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," *CS224N Project Report, Stanford University*, 2009, pp. 1–12.

[27] Y. Siri, S. Afroz, and R. U. Rani, "Enhancing sentiment analysis accuracy by optimizing hyperparameters of SVM and logistic regression models," *E3S Web of Conferences*, vol. 472, Jan. 2024, doi: 10.1051/e3sconf/202447201017.

[28] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995, doi: 10.1007/BF00994018.

[29] Z. Patel, M. Patel, R. Patel, and P. Patel, "Sentiment analysis on social media data using support vector machine," in *Intelligent Computing and Communication-(ICICC 2024)*, Springer, Singapore, 2025, pp. 189–197, doi: 10.1007/978-981-96-1267-3_17.

[30] H. Sharma and S. Kumar, "A survey on decision tree algorithms of classification in data mining," *International Journal of Science and Research*, vol. 5, no. 4, pp. 2094–2097, Apr. 2016, doi: 10.21275/v5i4.NOV162954.

[31] S. Singh and P. Gupta, "Comparative study of ID3, CART and C4.5 decision tree algorithm: a survey," *International Journal of Advanced Information Science and Technology*, vol. 27, no. 27, pp. 97–103, 2014, doi: 10.15693/ijaist/2014.v3i7.47-52.

[32] M. Rathi, A. Malik, D. Varshney, R. Sharma, and S. Mendiratta, "Sentiment analysis of tweets using machine learning approach," in *2018 Eleventh International Conference on Contemporary Computing (IC3)*, Aug. 2018, pp. 1–3, doi: 10.1109/IC3.2018.8530517.

[33] D. Heckerman, "Bayesian networks for data mining," *Data Mining and Knowledge Discovery*, vol. 1, no. 1, pp. 79–119, Mar. 1997, doi: 10.1023/A:1009730122752.

[34] J. Tolles and W. J. Meurer, "Logistic regression: relating patient characteristics to outcomes," *JAMA*, vol. 316, no. 5, pp. 533–534, Aug. 2016, doi: 10.1001/jama.2016.7653.

[35] K. Fukushima, "Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, Apr. 1980, doi: 10.1007/BF00344251.

[36] M. Umer *et al.*, "Impact of convolutional neural network and FastText embedding on text classification," *Multimedia Tools and Applications*, vol. 82, no. 4, pp. 5569–5585, Feb. 2023, doi: 10.1007/s11042-022-13459-x.

[37] M. Kaur and A. Mohta, "A review of deep learning with recurrent neural network," in *2019 International Conference on Smart Systems and Inventive Technology (ICSSIT)*, Nov. 2019, pp. 460–465, doi: 10.1109/ICSSIT46314.2019.8987837.

[38] S. Haykin, *Neural Networks and Learning Machines*, 3rd ed. Upper Saddle River, NJ, USA: Pearson Education, 2009.

[39] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Computation*, vol. 31, no. 7, pp. 1235–1270, Jul. 2019, doi: 10.1162/neco_a_01199.

[40] A. Rogers, O. Kovaleva, and A. Rumshisky, "A primer in BERTology: what we know about how BERT works," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 842–866, Dec. 2020, doi: 10.1162/tacl_a_00349.

[41] A. Joshy and S. Sundar, "Analyzing the performance of sentiment analysis using BERT, DistilBERT, and RoBERTa," in *2022 IEEE International Power and Renewable Energy Conference (IPRECON)*, Dec. 2022, pp. 1–6, doi: 10.1109/IPRECON55716.2022.10059542.

[42] A. R. Sajun, I. Zualkernan, and D. Sankalpa, "A historical survey of advances in transformer architectures," *Applied Sciences*, vol. 14, no. 10, May 2024, doi: 10.3390/app14104316.

[43] Y. Bouassida and H. Mezali, "Enhancing Twitter sentiment analysis using hybrid transformer and sequence models," *Japan Journal of Research*, vol. 6, no. 1, Jan. 2025, doi: 10.33425/2690-8077.1163.

[44] H. T. Phan, N. T. Nguyen, and D. Hwang, "Fake news detection: a survey of graph neural network methods," *Applied Soft Computing*, vol. 139, May 2023, doi: 10.1016/j.asoc.2023.110235.

[45] C. Sutton and A. McCallum, "An introduction to conditional random fields," *Foundations and Trends® in Machine Learning*, vol. 4, no. 4, pp. 267–373, Aug. 2012, doi: 10.1561/2200000013.

[46] S. Sıngh, K. Kumar, and B. Kumar, "Analysis of feature extraction techniques for sentiment analysis of Tweets," *Turkish Journal of Engineering*, vol. 8, no. 4, pp. 741–753, Oct. 2024, doi: 10.31127/tuje.1477502.
[47] O. Hourrane, N. Idrissi, and E. H. Benlahmar, "Sentiment classification on movie reviews and Twitter: an experimental study of supervised learning models," in *2019 1st International Conference on Smart Systems and Data Science (ICSSD)*, Oct. 2019, pp. 1–6, doi: 10.1109/ICSSD47982.2019.9003118.
[48] C. J. Varshney, A. Sharma, and D. P. Yadav, "Sentiment analysis using ensemble classification technique," in *2020 IEEE Students Conference on Engineering & Systems (SCES)*, Jul. 2020, pp. 1–6, doi: 10.1109/SCES50439.2020.9236754.
[49] N. Iqbal, A. M. Chowdhury, and T. Ahsan, "Enhancing the performance of sentiment analysis by using different feature combinations," in *2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2)*, Feb. 2018, pp. 1–4, doi: 10.1109/IC4ME2.2018.8465673.
[50] V. D. Derbentsev, V. S. Bezkorovainyi, A. V Matviychuk, O. M. Pomazun, A. V Hrabariev, and A. M. Hostryk, "A comparative study of deep learning models for sentiment analysis of social media texts," in *CEUR Workshop Proceedings, Vol-3465: Proceedings of the Selected and Revised Papers of 10th International Conference on Monitoring, Modeling & Management of Emergent Economy (M3E2-MLPEED 2022)*, Aug. 2023, pp. 168–188, doi: 10.55056/ceur-ws.org/Vol-3465/paper18.pdf.

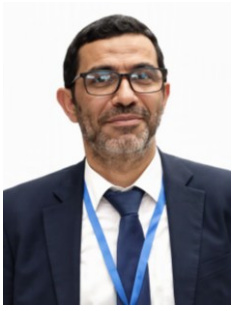
BIOGRAPHIES OF AUTHORS



Billal El Khattabi received a bachelor's degree in Mathematics and Applications from the University of Abdelmalek Essaadi (UAE) in Tétouan, Morocco. He then obtained a research master's degree in Signal Processing and Machine Learning from the National School of Applied Sciences at the same university in 2020. He is currently pursuing a Ph.D. in Computer Science at the Faculty of Sciences, University of Abdelmalek Essaadi, Tétouan, Morocco. His research focuses on developing models and architectures for NLP. He can be contacted at email: billal.elkhattabi@etu.uae.ac.ma or b.khattabi@gmail.com.



Hicham Omara is currently a professor of computer science in the Department of Computer Science at the Faculty of Polydisciplinary of Taza, Sidi Mohamed Ben Abdellah University, Morocco. He received his Ph.D. from the Faculty of Science at Abdelmalek Essaâdi University, Morocco. He has participated in numerous national and international scientific and organizational committees. His research interests include AI, image processing, big data, and other areas. He can be contacted at email: h.omara@uae.ac.ma.



Adnan Souri is currently a professor in Computer Science at Abdelmalek Essaâdi University, Faculty of Sciences, Tetouan, Morocco. He is a former teacher of computing at preparatory classes for engineering schools–Tangier. He received his Ph.D. degree in Computer Science from the Faculty of Sciences, Abdelmalek Essaâdi University, Tetouan, Morocco. His current research interests include AI, artificial neural networks, and algorithms. Their current project is "Arabic language processing". He can be contacted at email: a.souri@uae.ac.ma.