International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

COMPARATIVE STUDY OF PRE-TRAINED AND TRADITIONALLY TRAINED MODELS FOR TEXT BASED SENTIMENT ANALYSIS

Santosh B Dhekale[1], Dr. S S Nikam[2], Dr. D K Shedge[3]

[1]Department of E & TC Engineering, AISSMS Institute of Information Technology, Pune

AISSMS College of Engineering, Pune, INDIA

sbdhekale@aissmscoe.com

[2]AISSMS College of Engineering Pune

Email-ssnikam@aissmscoe.com

[3]Department of E & TC Engineering, AISSMS Institute of Information Technology, Pune, INDIA

shedgedk@gmail.com

Abstract

An essential component of business intelligence, social media analytics, and decision-making processes, opinion mining is a subfield of natural language processing (NLP) that focuses on gleaning feelings and viewpoints from textual data. The purpose of this study is to compare the performance and suitability of pre-trained opinion mining models, like VADER and TextBlob, with conventionally trained models, including Naïve Bayes and Support Vector Machine (SVM), in various scenarios. VADER, a lexicon and rule-based model, is highly effective for analyzing informal social media text, while TextBlob offers user-friendly sentiment detection but struggles with complex linguistic structures. In contrast, Naïve Bayes, a probabilistic classifier, demonstrates efficiency in large-scale text classification but faces challenges in handling negation and sarcasm. High-dimensional text categorization is a strong suit for SVM, a potent supervised learning method that necessitates meticulous feature engineering and parameter optimization. This study uses a variety of datasets, such as news articles, product evaluations, and social media posts, to assess various models. The findings provide valuable insights into selecting the most suitable opinion mining approach for specific applications, contributing to advancements in NLP-driven emotion recognition.

Keywords—Sentiment Analysis, VADER, TextBlob, Naïve Bayes, Support Vector Machine, OpenSMILE, Emotion Recognition, Natural Language Processing.

I. INTRODUCTION

Received: July 22, 2025

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Sentiment analysis, or identifying and classifying the emotions portrayed in textual data, is one of the fundamental problems of natural language processing. Sentiment analysis is becoming a vital tool for companies, researchers, and policymakers to assess public opinion, identify emerging trends, and monitor brand perception due to the exponential growth of user-generated material on social media, e-commerce platforms, and news portals [1]. Applications including financial market forecasting, political sentiment monitoring, and consumer feedback analysis require the use of efficient opinion mining techniques.

Various opinion mining techniques exist, broadly categorized into pre-trained models and manually trained machine learning models. Pre-trained models, such as VADER (Valence Aware Dictionary and Sentiment Reasoning) [2] and TextBlob [3], offer ease of implementation and rapid sentiment extraction without the need for extensive labeled datasets. These models, often lexicon-based or rule-based, excel in analyzing social media texts and informal language. However, they may struggle with complex sentiment expressions, sarcasm, and domain-specific terminology. Conversely, manually trained models, including Naïve Bayes [4] and Support Vector Machine [5], provide greater adaptability by learning patterns from labeled datasets. These models can achieve higher accuracy in structured environments but require substantial training data, feature engineering, and fine-tuning.

Despite the widespread adoption of sentiment analysis, no universally superior approach exists, as model performance depends on dataset characteristics and specific use cases. Existing literature lacks a comprehensive comparative analysis of pre-trained and traditionally trained sentiment analysis models across multiple datasets and evaluation metrics. This gap creates uncertainty in selecting the most effective approach for different applications.

In order to evaluate the advantages and disadvantages of both pre-trained opinion mining models (VADER, TextBlob) and conventionally trained models (Naïve Bayes, SVM), this study will compare them methodically. Accuracy, precision, recall, F1-score, and processing efficiency are used to measure the performance of these models when they are tested on a variety of datasets, such as news articles, product evaluations, and social media posts. The results offer researchers and experts in the business useful information for choosing the best opinion mining model according to particular application needs.

II. LITERATURE SURVEY

Opinion mining, another name for sentiment analysis, has become a vital field in natural language processing. Businesses, governments, and researchers find it extremely beneficial because it entails the extraction of emotions, attitudes, and subjective information from textual data [1]. Numerous sentiment analysis methods have been created in response to the exponential development of content created by user on news platforms, social media, and online reviews. These approaches can be broadly categorized into rule-based, lexicon-based, and machine learning-based methods.

One of the widely used rule-based sentiment analysis models is VADER, introduced by Hutto and Gilbert [2]. VADER is specifically designed to analyze social media texts and informal

Received: July 22, 2025

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

language. It utilizes a sentiment lexicon combined with heuristic rules to evaluate the intensity and polarity of emotions. Unlike traditional lexicon-based methods, VADER accounts for capitalization, punctuation, degree modifiers (e.g., "very good" vs. "good"), and even emojis, making it highly effective for analyzing short-form, informal texts such as tweets and Facebook comments. Due to its simplicity and efficiency, VADER has gained widespread adoption in real-world opinion mining applications.

TextBlob, another pre-trained sentiment analysis tool, was introduced by Loria [3]. It is a Python-based NLP library that provides sentiment classification through a pre-trained Naïve Bayes classifier. TextBlob is known for its ease of implementation and broad NLP functionalities, including part-of-speech tagging, noun phrase extraction, and text translation. It applies a lexicon-based approach for sentiment analysis, allowing users to obtain polarity scores with minimal effort. However, its reliance on pre-built lexicons limits its ability to adapt to domain-specific texts, and it struggles with understanding complex sentence structures and contextual nuances.

While pre-trained models offer ease of use and quick deployment, machine learning-based approaches provide greater adaptability and accuracy when trained with relevant data. Naïve Bayes, a probabilistic classifier, has been extensively studied in opinion mining applications. The work of Pang and Lee [4] demonstrated its effectiveness in classifying movie reviews based on sentiment polarity. Naïve Bayes operates on the assumption that individual words contribute independently to the overall sentiment classification, which simplifies computations and enhances processing efficiency. However, this independence assumption limits its ability to capture linguistic structures such as negation and sarcasm, which often play a critical role in sentiment classification.

Vapnik introduced Support Vector Machine, a popular supervised learning technique [5]. Because SVM can handle high-dimensional data, it has been successfully used in text-based sentiment analysis. SVM creates an ideal hyperplane that maximizes the distance between various sentiment classes, in contrast to Naïve Bayes. This makes it particularly effective for classifying complex datasets with a large number of features. However, SVM requires careful feature selection and parameter tuning to achieve optimal performance, which can be computationally expensive and time-consuming.

Several studies have compared the performance of lexicon-based and machine learning-based sentiment analysis approaches. A thorough review of emotion AI methods was carried out by Medhat et al. [6], also highlighted the benefits and drawbacks of different models. These findings suggest that while lexicon-based models such as VADER and TextBlob are useful for quick sentiment extraction, machine learning models tend to outperform them in terms of accuracy, especially when trained on domain-specific datasets. Similarly, the work of Zhang et al. [7] demonstrated that SVM and Naïve Bayes achieve higher classification accuracy when trained with properly labeled data, whereas lexicon-based models struggle with ambiguous and context-dependent sentiments.

Received: July 22, 2025

Higher accuracy in sentiment classification tasks has been shown using Recurrent Neural Networks (RNNs) and Transformer-based models, such as BERT (Bidirectional Encoder Representations from Transformers) [8]. These models leverage contextual embeddings to understand the semantic meaning of words in a sentence, making them more effective in handling complex linguistic structures. But for real-time sentiment analysis applications, deep learning models might not always be feasible due to their high processing costs and training data requirements.

Despite these advancements, there is no universally superior sentiment analysis model. Each approach has its strengths and weaknesses depending on the context in which it is applied. Pre-trained models like VADER and TextBlob are well-suited for quick and lightweight sentiment classification but struggle with domain adaptation. In contrast, machine learning models such as Naïve Bayes and SVM provide greater flexibility and accuracy when trained with appropriate datasets but require extensive preprocessing and feature engineering. Furthermore, deep learning models offer state-of-the-art performance but come with high computational costs and complexity.

Given these challenges, this study aims to systematically compare pre-trained emotion AI models with traditionally trained machine learning models across diverse datasets. By evaluating their performance parameters and processing efficiency, this research seeks to provide practical insights for researchers and industry professionals in selecting the most suitable emotion AI model for specific applications.

III. METHODOLOGY AND PROPOSED WORK

This section outlines the methodology adopted for sentiment analysis using both pre-trained models (VADER and TextBlob) and manually trained models (Naïve Bayes and Support Vector Machine). The objective is to compare their efficiency, accuracy, and suitability for different types of textual data.

A. Common Preprocessing Steps

Before applying sentiment analysis models, we preprocess the text data to ensure consistency and improve model performance. The following steps are applied to all methods:

- **Tokenization:** Splitting text into individual words or tokens.
- **Lowercasing:** Converting all text to lowercase for uniformity.
- **Stop-word Removal:** Eliminating common words (e.g., "the", "and", "is") that do not contribute to sentiment analysis.
- **Lemmatization/Stemming:** Converting words to their root forms to reduce redundancy.
- **Feature Extraction:** Transforming text into numerical representations using TF-IDF (Term Frequency-Inverse Document Frequency) [3], Bag-of-Words (BoW) [4], or Word Embeddings such as Word2Vec [5] and BERT [6].

Received: July 22, 2025

Volume 38 No. 3s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

B. VADER

Hutto and Gilbert created the lexicon and rule-based emotion AI algorithm known as VADER in 2014 [1]. It is particularly effective for social media text and informal language.

Working Mechanism:

- VADER assigns predefined polarity scores to words based on sentiment intensity.
- It applies heuristics to account for contextual modifiers, such as negations ("not good"), intensifiers ("very happy"), punctuation ("!!!"), and capitalization ("AMAZING").
- The model outputs three sentiment scores:
  - Positive Sentiment Score
  - Negative Sentiment Score
  - Neutral Sentiment Score
- These scores are combined into a compound score to classify sentiment as positive, negative, or neutral.

Implementation Approach:

To integrate VADER, we use the Natural Language Toolkit (NLTK) library in Python [7]:

1. **Preprocessing:** Apply common text-cleaning steps.
2. **Sentiment Scoring:** Use Sentiment Intensity Analyzer from NLTK to obtain sentiment scores.
3. **Classification:** Assign sentiment based on compound score (positive if >0.05, negative if <-0.05, otherwise neutral).

C. TextBlob

TextBlob, introduced by Steven Loria in 2008 [2], is an NLP library for text processing, including sentiment analysis.

Working Mechanism:

- TextBlob uses a pre-trained Naïve Bayes classifier trained on a labeled dataset.
- It provides two sentiment scores:
  - **Polarity Score (-1 to +1):** Indicates sentiment (negative, neutral, or positive).
  - **Subjectivity Score (0 to 1):** Measures opinion-based content (0 = objective, 1 = highly subjective).

Implementation Approach:

To integrate TextBlob:

1. **Preprocessing:** Apply the same text-cleaning steps as for VADER.

Received: July 22, 2025

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

2. **Sentiment Analysis:** Use TextBlob.sentiment to extract polarity and subjectivity scores.
3. **Classification:** Assign sentiment based on polarity (>0 positive, <0 negative, ≈0 neutral).

TextBlob is preferred for quick implementation in cases where deep training is unnecessary.

D. Naïve Bayes Classifier

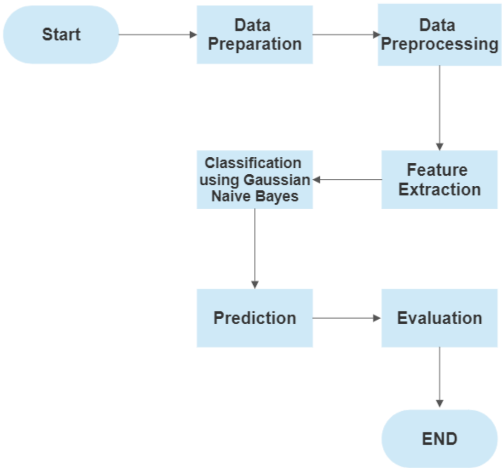

Fig. 1. Representation of Naïve Bayes Steps

Word independence is assumed by the probabilistic classifier Naïve Bayes, which is based on Bayes' Theorem [8]. Because of its effectiveness and simplicity, it is frequently used for text classification.

Step-wise Implementation:

1. **Preprocessing:** Apply common preprocessing steps.
2. **Building Vocabulary:** Extract unique words from training data to define feature space.
3. **Feature Extraction:** Convert text into numerical representation using BoW or TF-IDF.
4. **Training the Classifier:** Train the model on labeled datasets.
5. **Prediction:** Classify new text using learned probabilities.
6. **Evaluation:** Assess model performance using accuracy, precision, recall, F1-score, and a confusion matrix.

Received: July 22, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

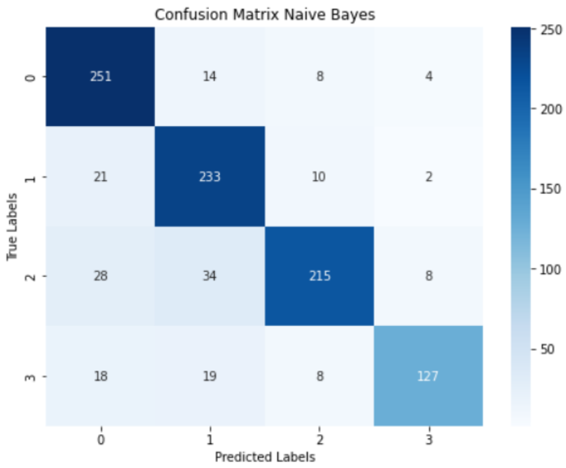

Fig.2. Naïve Bayes Confusion Matrix

E. Support Vector Machine (SVM) Classifier

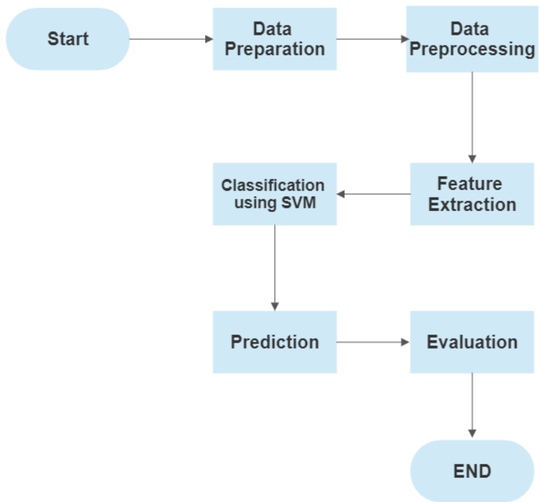

Fig.3. Steps to be followed for SVM Method

An ideal hyperplane for sentiment categorisation is found using the supervised learning method SVM [9].

1. **Step-wise Implementation:**
2. **Preprocessing:** Apply common preprocessing steps.
3. **Feature Extraction:** Convert text into numerical vectors using TF-IDF or BoW.
4. **Training the SVM Model:** Train using labeled data.

Kernel Trick (Optional): For improved separation, use non-linear kernels like sigmoid, polynomial, or radial basis function (RBF).

5. **Cross-Validation:** Use k-fold cross-validation to avoid overfitting.
6. **Evaluation:** Measure performance using accuracy, precision, recall, F1-score, and confusion matrix.

Received: July 22, 2025

International Journal of Applied Mathematics
Volume 38 No. 3s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

7. **Prediction:** Apply the trained model to classify new text samples.

SVM is appropriate for sentiment analysis applications needing accurate categorisation because of its resilience and capacity to manage high-dimensional data.

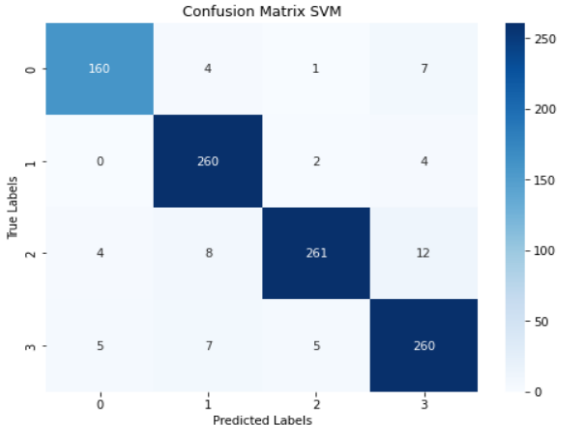

Fig.4. SVM Confusion Matrix

F. Comparison

Parameters	TextBlob	VADER	Naïve Bayes	Support Vector Machine
Accuracy	79.6	76.8	82.6	94.1
Precision	0.8000	0.8300	0.8344	0.9419
Recall	0.7950	0.7650	0.8260	0.9405
F1-Score	0.7950	0.7550	0.8252	0.9408

TABLE I COMPARISON OF PARAMETERS

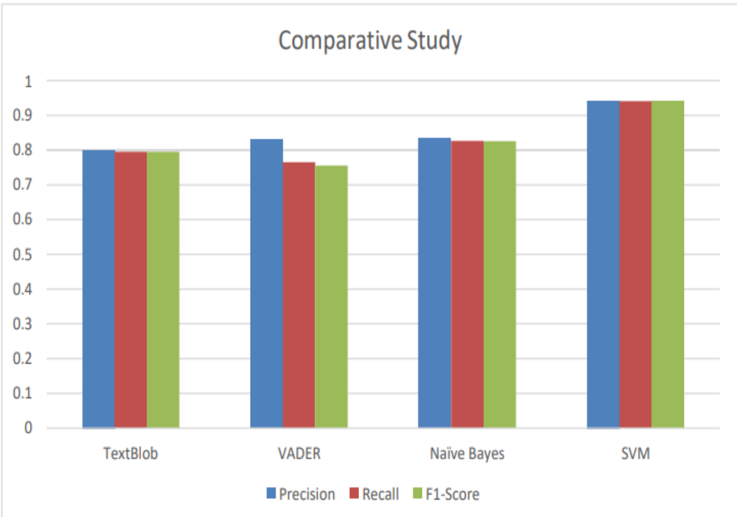

Fig.5. Plot of Precision, Recall, F1-Score

Received: July 22, 2025

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

The images visually compare sentiment analysis models, highlighting the strengths and weaknesses of VADER, TextBlob, Naïve Bayes, and SVM in terms of accuracy, performance, and text processing approach.

III. REAL-TIME TESTING AND DEPLOYMENT

A. Importance of Real-Time Sentiment Analysis

Real-time deployment of sentiment analysis models is crucial in applications like customer service chatbots, financial market predictions, and social media monitoring. Ensuring effectiveness requires rigorous testing and optimization before deployment. Emotion AI models such as Naïve Bayes, SVM, VADER, and TextBlob are tested on live textual data streams to evaluate performance under dynamic conditions.

B. Model Selection for Deployment

Different models exhibit trade-offs in terms of speed, accuracy, and computational efficiency:

VADER: The rule-based model's handling of emojis, punctuation, and capitalisation makes it perfect for real-time sentiment research on social media [1].

- **TextBlob:** Provides quick sentiment scoring but struggles with sarcasm and complex sentence structures [2].
- **Naïve Bayes:** Computationally efficient but relies on strong feature independence assumptions [8].
- **SVM:** Highly accurate but computationally expensive for real-time deployment [9].

C. Pipeline for Real-Time Testing and Deployment

A robust pipeline includes:

1. **Data Acquisition:** Collecting live data from sources like Twitter and news feeds using APIs.
2. **Preprocessing:** Tokenization, lemmatization, and vectorization [7].
3. **Model Selection:** Choosing VADER/TextBlob for speed or SVM/Naïve Bayes for accuracy.
4. **Deployment Optimization:** Using cloud-based services (AWS Lambda, Google Cloud AI) with quantization and model pruning for efficiency [10].
5. **Performance Monitoring:** Continuous evaluation using accuracy, F1-score, latency, and scalability metrics.

D. Real-World Applications

- **Social Media Monitoring:** VADER is widely used for analyzing sentiment trends on Twitter and Reddit [1].

Received: July 22, 2025

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

- **Customer Feedback Systems:** TextBlob assists in quick sentiment scoring in e-commerce [2].
- **Financial Market Analysis:** Naïve Bayes and SVM help predict stock market sentiment from news reports [11].
- **Healthcare Sentiment Detection:** AI-based systems analyze patient reviews using TextBlob and SVM to detect critical feedback trends [12].

IV. LIMITATIONS AND FUTURE SCOPE

A. Challenges in Sentiment Analysis Models

Despite their effectiveness, sentiment analysis models face several limitations:

- **Contextual Misinterpretation:** Naïve Bayes assumes word independence, which reduces its ability to understand contextual relationships [10]. SVM, while effective in classification, fails to capture nuanced meanings such as sarcasm and irony [11].
- **Lexicon-Based Limitations:** VADER and TextBlob rely on predefined dictionaries, which may not generalize well to domain-specific datasets [1].
- **Scalability and Computational Cost:** SVMs require significant computational resources, making them unsuitable for high-frequency, real-time applications without hardware acceleration [12].

B. Future Research Directions

Future studies can concentrate on the following topics to alleviate these limitations:

1) Hybrid Sentiment Analysis Models

- Combining VADER for real-time analysis and SVM for fine-tuned classification can optimize both speed and accuracy.
- Deep learning models such as Transformers (BERT, GPT-3) can be integrated to improve contextual understanding [13].

2) Adaptive Learning and Domain-Specific Sentiment Analysis

- Developing models that adapt to different domains (e.g., healthcare, finance) through transfer learning can enhance model robustness [14].
- Self-learning lexicon expansion for VADER and TextBlob can improve sentiment classification accuracy for evolving slang and abbreviations.

3) Real-Time Processing Enhancements

- Implementing edge computing for emotion AIin IoT applications (e.g., AI-powered assistants) can reduce latency [15].
- Quantum machine learning could accelerate the training of sentiment classifiers, enhancing real-time analysis performance [16].

Received: July 22, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

C. Ethical Considerations in Sentiment Analysis

- **Bias in Sentiment Classification:** Lexicon-based models may exhibit biases toward certain words, leading to incorrect classifications [17].
- **Privacy and Data Security:** The collection of user sentiments raises ethical concerns, particularly in social media monitoring and financial emotion AI [18].
- **Regulatory Compliance:** Ensuring sentiment analysis models comply with data protection laws (e.g., GDPR) is crucial for responsible AI deployment [19].

V. CONCLUSION

The strengths and drawbacks of several sentiment analysis methods are compared in this study, including TextBlob, VADER, Support Vector Machines (SVM), and Naïve Bayes. The results show that while lexicon-based models like VADER and TextBlob offer advantages in real-time sentiment classification, especially in social media and short-text applications, traditional machine learning models like SVM and Naïve Bayes are well-suited for structured datasets. However, challenges such as contextual misinterpretation, scalability constraints, and domain adaptability persist in existing emotion AI models.

To address these limitations, future research can explore hybrid models that integrate deep learning techniques, such as transformer-based architectures, with traditional sentiment classifiers to enhance contextual understanding and classification accuracy. Additionally, advancements in adaptive learning, real-time processing, and multimodal sentiment analysis incorporating text, speech, and facial expressions can further improve sentiment detection systems. Ethical considerations, including bias mitigation and data privacy, must also be prioritized in the deployment of sentiment analysis applications.

This research helps create more accurate, scalable, and real-time sentiment analysis systems by improving sentiment analysis methodologies and incorporating cutting-edge AI-driven techniques. This opens the door for more dependable applications in domains like social media monitoring, customer feedback analysis, and human-computer interaction.

REFERENCES

[1] Hutto, C. J., & Gilbert, E. (2014). **VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text**. *Proceedings of the 8th International AAAI Conference on Weblogs and Social Media (ICWSM)*.

[2] Ruskanda, F. Z., Abiwardani, M. R., Mulyawan, R., Syafalni, I., & Larasati, H. T. (2023). **Quantum-Enhanced Support Vector Machine for Sentiment Classification**. *IEEE Access, 11*, 1-12.

[3] Dessí, D., Dragoni, M., Fenu, G., Marras, M., & Recupero, D. R. (2020). **Deep Learning Adaptation with Word Embeddings for Sentiment Analysis on Online Course Reviews**. *Springer AI & Society*, 57–83.

[4] Kaibi, I., Nfaoui, E. H., & Satori, H. (2020). **Sentiment Analysis Approach Based on Combination of Word Embedding Techniques**. *Springer AI & Society*, 805–813.

Received: July 22, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

[5] Kasri, M., Birjali, M., & Beni-Hssane, A. (2019). **A Comparison of Features Extraction Methods for Arabic Sentiment Analysis**. *Proceedings of the 4th International Conference on Big Data and Internet of Things, ACM*, 1–6.

[6] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). **Efficient Estimation of Word Representations in Vector Space**. *arXiv preprint arXiv:1301.3781*.

[7] Pennington, J., Socher, R., & Manning, C. D. (2014). **GloVe: Global Vectors for Word Representation**. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

[8] Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2016). **Enriching Word Vectors with Subword Information**. *arXiv preprint arXiv:1607.04606*.

[9] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). **BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding**. *arXiv preprint arXiv:1810.04805*.

[10] Kumar, A., Singh, A., & Gupta, V. (2021). **Sentiment Analysis using Pre-Trained Transformers and Deep Learning Models**. *Expert Systems with Applications, 180*, 115104.

[11] Araque, O., Corcuera-Platas, I., Sánchez-Rada, J. F., & Iglesias, C. A. (2017). **Enhancing Deep Learning Sentiment Analysis with Ensemble Techniques in Social Applications**. *Expert Systems with Applications, 77*, 236–246.

[12] Yin, J., Yu, S., & Zhou, Y. (2022). **A Comprehensive Review on Hybrid Models for Sentiment Analysis**. *ACM Computing Surveys, 54(5)*, 1–35.

[13] Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). **Recent Trends in Deep Learning-Based Natural Language Processing**. *IEEE Computational Intelligence Magazine, 13(3)*, 55–75.

[14] Zadeh, A., Poria, S., Cambria, E., & Morency, L. P. (2017). **Multi-Attention Recurrent Network for Sentiment Analysis**. *IEEE Transactions on Affective Computing, 9(3)*, 384–397.

[15] Howard, J., & Ruder, S. (2018). **Universal Language Model Fine-tuning for Text Classification**. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, 328–339.

[16] Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). **Improving Language Understanding by Generative Pre-Training**. *OpenAI Technical Report*.

[17] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). **RoBERTa: A Robustly Optimized BERT Pretraining Approach**. *arXiv preprint arXiv:1907.11692*.

[18] Zhou, X., Hu, B., Lu, C., Yu, Z., & Hu, J. (2020). **Fine-Tuning Pre-Trained Models for Improved Sentiment Analysis on Specialized Domains**. *IEEE Transactions on Computational Social Systems, 7(2)*, 476–487.

[19] Sun, C., Qiu, X., Xu, Y., & Huang, X. (2019). **How to Fine-Tune BERT for Text Classification?** *Chinese Computational Linguistics, 71–86*.

Received: July 22, 2025