Vol.8 No.2 April 2025

Multimedia Research

Received 25 March, Revised 29 April, Accepted 6 May

Comparative Analysis on Artificial Intelligence Technique for Sentiment Analysis

Saranya Kamaraj

Department of Computer Science and Engineering
Bharathiyar Institute of Engineering for Women
Tamil Nadu, India
ksaranya@biew.ac.indiv

Abstract: Sentiment` analysis, a key task in natural language processing (NLP), involves recognizing and classifying emotions expressed in textual data. By analyzing online discussions, companies can gain insights into public sentiment toward their offerings. With the rise of digital platforms like blogs and social media, people now frequently share reviews and opinions related to daily activities. This study presents a comparative analysis of sentiment analysis techniques by reviewing 15 research papers encompassing machine learning (ML), deep learning (DL), and traditional NLP approaches. The analysis highlights the evolving landscape of sentiment classification, showcasing the performance of models such as Support Vector Machines (SVM), Naïve Bayes, Random Forest (RF), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and advanced architectures like Capsule Networks and Switch Transformers. The studies were evaluated using common metrics such as precision, recall, accuracy, and F1-score. DL methods, particularly those combining LSTM and CNN with optimization strategies, performed better across all metrics, including precision, recall, accuracy, and F1-score. These hybrid DL models showed strong contextual understanding and handled complex data well. Traditional ML models offered interpretability and ease of implementation, while NLP-based methods remained competitive for lightweight applications. The review also notes challenges like sarcasm, domain shifts, and imbalanced data that hinder generalization. This comparative review provides insights into the strengths, limitations, and suitable application contexts of each approach, offering a foundation for selecting appropriate sentiment analysis methods in future research and real-world applications.

Keywords: Sentiment Analysis, Machine Learning, Deep Learning, Natural Language Processing, Emotion Detection

Nomenclature

Abbreviation	Description
ABC	Artificial Bee Colony
ABO	African Buffalo Optimization
BoW	Bag-of-Words
CapsNet	Capsule Networks
CNN	Convolutional Neural Network
DL	Deep Learning
GRU	Gated Recurrent Unit
GSA	Gravitational Search Algorithm
GA	Genetic Algorithm
KNN	K-Nearest Neighbor
LR	Logistic Regression
LSTM	Long Short-Term Memory
ML	Machine Learning
NLP	Natural Language Processing
RBM	Restricted Boltzmann Machines
RF	Random Forest
RNN	Recurrent Neural Network
SA	Sentiment Analysis
SVM	Support Vector Machines
TF-IDF	Term Frequency-Inverse Document Frequency

https://doi.org/10.46253/j.mr.v8i2.a5

1. Introduction

With the rapid advancement of smart devices and mobile internet, user-generated content on social media has become increasingly diverse, extending beyond simple text formats. Modern platforms now enable users to share opinions through various online communities, providing valuable insights into user behavior. Analyzing these opinions through sentiment analysis plays a crucial role in capturing public sentiment and understanding emerging trends. In the context of e-commerce, online reviews and ratings have a strong influence on consumer purchasing decisions [1]. Consumers frequently depend on these reviews to assess products and interact with both retailers and fellow buyers. However, the absence of ratings or the presence of vague or overly simplistic reviews can hinder effective product evaluation. Furthermore, inconsistencies between written reviews and their corresponding ratings pose a significant challenge for accurately gauging product appeal [2]. These challenges highlight the need for advances sentiment analysis techniques capable of interpreting nuanced and context-rich user input. Leveraging more sophisticated models can help bridge the gap between subjective language and quantifiable sentiment, improving the reliability of automated opinion mining in online marketplaces.

Sentiment Analysis (SA) involves the identification, extraction, and interpretation of sentiments embedded within textual data, utilizing techniques from Natural Language Processing (NLP), Machine Learning (ML), and Deep Learning (DL). Various approaches have been applied to SA, particularly in extracting class-specific informative features and selecting effective classification models. Feature extraction plays a critical role in enhancing both model performance and interpretability by transforming unstructured input text into meaningful representations [3]. This transformation simplifies the data, highlights important patterns, and removes irrelevant information, making it easier to model sentiment accurately. Established methods such as Word Embeddings, Term Frequency-Inverse Document Frequency (TF-IDF), Bag-of-Words (BoW), and N-Gram models have proven effective in uncovering underlying semantic structures in textual data. Recent methods also leverage pre-trained embeddings and attention mechanisms to better capture semantic relationships.

Sentiment analysis and NLP face numerous challenges stemming from informal writing styles, sarcasm, irony, and language-specific complexities [4]. The interpretation of words can vary significantly depending on their contextual and domain-specific usage, and this variability is further complicated by the limited availability of tools and annotated resources for many languages. Additionally, capturing long-range dependencies and nuanced expressions like sarcasm, irony, or metaphor remains a major limitation. Large, complex models are also prone to overfitting, especially when trained on limited or noisy datasets [5]. Moreover, these models may struggle with out-of-vocabulary terms or uncommon linguistic patterns, which negatively affect their robustness and ability to generalize across diverse text inputs. Domain adaptation and data augmentation techniques are being explored to improve generalizability and reduce overfitting. The main contributions of this analysis are as follows:

- A thorough review of 15 recent research papers on sentiment analysis has been conducted, highlighting key methodologies, datasets, tools, and findings across diverse applications.
- The study systematically compares sentiment analysis techniques based on ML, DL, and NLP, examining their strengths, limitations, and suitability for various linguistic and contextual challenges.
- A detailed performance evaluation is presented, comparing the accuracy, robustness, and effectiveness of the reviewed approaches, thereby identifying the most efficient techniques under different conditions.
- The review also outlines ongoing research efforts aimed at addressing challenges like sarcasm, noisy data, and limited labeled resources.

The rest of the paper is organized as follows: Section reveals the classification of SA. Section 3 analyzes the performance of various approaches across various metrics. Finally, Section 4 summarizes the comparative analysis of SA.

2. Classification of Sentimental Analysis

A comparative analysis of Sentiment Analysis (SA) involves examining the different approaches, including ML, DL, and NLP, used for determining sentiment in textual data. Fig. 1 shows the classification of sentimental analysis. Below is a structured comparison that is given as follows: The classification aids in understanding how each method processes and interprets sentiment features, helping to evaluate their applicability in diverse contexts.

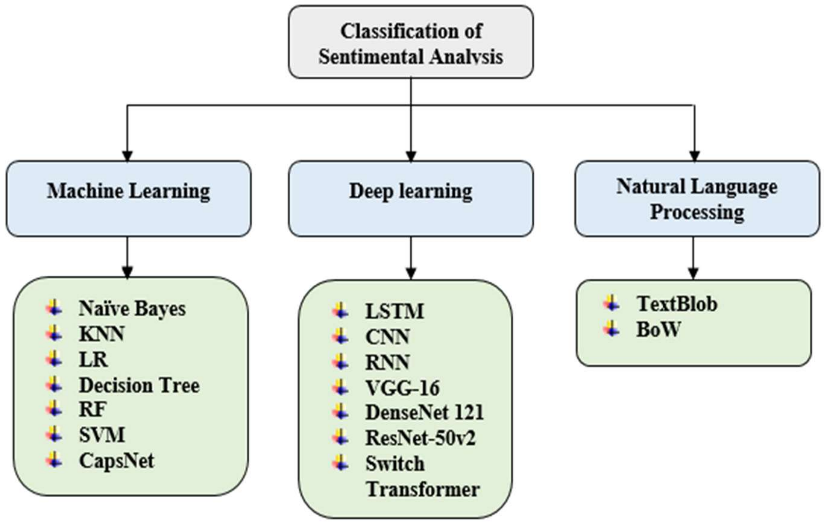

Fig.1. *Classification of sentimental analysis*

2.1 Machine Learning-based Sentimental Analysis

Machine learning-based sentiment analysis employs various algorithms to automatically identify and classify emotions or opinions expressed in textual data, which are discussed in detail as follows. These methods are favored for their simplicity, interpretability, and relatively low computational cost compared to deep models.

2.1.1 Naïve Bayes

Naïve Bayes is a simple yet powerful probabilistic classifier commonly employed for sentiment analysis due to its efficiency and effectiveness with text data. The architecture relies on Bayes' Theorem, assuming that the occurrence of one feature (e.g., a word) is independent of the occurrence of others, hence the term "naïve." Despite its simplicity and the independence assumption, Naïve Bayes performs remarkably well for large-scale text classification tasks like sentiment analysis, especially when the dataset is well-balanced and feature engineering is properly applied. In the reviewed analysis [8] used Naïve Bayes as a standalone classifier, while [7] included it as part of an ensemble approach that demonstrated its continued relevance and utility in sentiment classification tasks. Its fast training time and low memory requirements make it suitable for real-time sentiment monitoring systems.

2.1.2 k-Nearest Neighbor (KNN)

The K-Nearest Neighbor (KNN) algorithm is a non-parametric, simple, instance-based learning method that can be effectively used for sentiment analysis. In this approach, each text sample is first preprocessed and transformed into a numerical feature vector using techniques like word embeddings or TF-IDF. When a new input text is given, the algorithm estimates the distance between the input and all labeled training samples in the feature space. It then identifies the 'k' most similar texts and assigns the sentiment label that is most frequent among them. As noted in [7], KNN was used as part of an ensemble of machine learning classifiers, contributing to sentiment prediction by leveraging similarity-based classification in feature space. However, its high computational cost during prediction can be a limitation in large-scale applications.

2.1.3 Logistic Regression

Logistic Regression (LR) is a widely adopted linear classification algorithm that is particularly effective for binary sentiment analysis, such as classifying text as positive or negative. In this architecture, the input text is first preprocessed, which typically involves tokenization, stop-word removal, and transformation into numerical features using methods like Bag-of-Words or TF-IDF. These feature vectors are then input into the LR model, which estimates a weighted summation of the input features and passes the result through a sigmoid function to obtain the output. The model is trained by adjusting the weights using gradient descent to minimize the difference between actual and predicted sentiment labels. Despite its simplicity, logistic regression is known for its robustness and interpretability, and was included in the ensemble of models used in [7], showcasing its effectiveness in sentiment classification tasks. Its straightforward implementation and ability to handle sparse features make it ideal for quick baseline models.

2.1.4 Decision Tree

Decision Tree is a supervised learning algorithm that works well for sentiment analysis by learning a series of if-then-else decision rules from text features. In this architecture, the input text is first preprocessed and converted into numerical features that are typically obtained using TF-IDF, Bag-of-Words, or word embeddings. The model recursively splits the data according to feature values that provide the highest information gain or Gini index reduction, leading to a tree that classifies input texts by traversing from the root to a leaf node. Moreover, decision trees are easy to interpret and visualize, and they handle both non-linear and linear patterns in data. As referenced in [7], Decision Trees were part of the ensemble used for sentiment classification, contributing to the model's overall interpretability and predictive power. However, they are prone to overfitting on noisy datasets unless combined with pruning or ensemble methods.

2.1.5 Random Forest

Random Forest (RF) is an ensemble learning algorithm that forms multiple decision trees, each learned on an arbitrary subset of the dataset, and combines their predictions through majority voting or averaging. Such an ensemble method helps improve the accuracy and robustness of the model, making it ideal for sentiment analysis, where it can efficiently classify text data like positive, negative, or neutral by learning from complex relationships and patterns in the features. In the table, RF is used in [7], where it achieved an accuracy of 0.88, demonstrating its effectiveness in sentiment analysis tasks across various data sources. It also handles feature redundancy well and reduces the risk of overfitting common in single decision trees.

2.1.6 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a supervised learning algorithm used for classification tasks, including sentiment analysis. SVM works by finding the optimal hyperplane that best separates the data into different classes, such as negative, positive, or neutral sentiment, while maximizing the margin between the classes. In sentiment analysis, SVM is effective at handling high-dimensional data and can classify text based on learned patterns in feature space. It is particularly useful for tasks where the boundary between classes is not linear, as it can use various kernel functions to convert data into a higher-dimensional space. In [7], SVM achieved an accuracy of 0.88, demonstrating its strong performance in sentiment classification tasks. SVM is particularly effective when working with sparse textual data and small datasets.

2.1.7 Capsule Network

Capsule Networks (CapsNet) are an advanced deep learning architecture designed to capture and preserve the spatial hierarchies and relationships between features, which is especially useful in sentiment analysis. In CapsNet, groups of neurons called capsules work together to represent entities and their properties, enabling the model to understand complex relationships within the data. This approach is effective in sentiment analysis as it can better capture the context and nuances of sentiment expression in text. Capsule Networks are also robust to changes in the input data, enhancing the capability of the model to generalize. In [6], CapsNet, combined with the GSA, achieved an accuracy of 0.91, showcasing its strong performance in sentiment classification tasks. Its dynamic routing mechanism helps capture sentiment orientation from subtle linguistic patterns.

2.2 Deep Learning-based Sentimental Analysis

Deep learning-based sentiment analysis utilizes advanced neural network architectures to automatically learn from and classify emotions or opinions in textual data. The various approaches are outlined in detail below. These models can extract high-level features directly from raw text, offering superior performance in complex scenarios.

2.2.1 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) networks are a specialized type of RNN that excels at learning and remembering long-term dependencies in sequential data. This makes LSTMs particularly effective for SA, where accepting the context and sequence of words is essential to responsible for the sentiment (positive, negative, or neutral). LSTMs utilize gates to control the flow of information, allowing the model to retain significant characteristics over long sequences and discard irrelevant ones. In [10] [16], LSTM demonstrates its effectiveness in sentiment classification. Additionally, [17] uses LSTM and RNN, showcasing the model's strong performance in analyzing sentiment across different datasets. Also [18] combines LSTM with CNN and GRU, achieving an accuracy of 91%, further illustrating the versatility and power of LSTM in sentiment analysis tasks. Its architecture makes it resilient to vanishing gradients, enabling better performance on longer text.

2.2.2 Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNNs) are used in several papers for SA due to their ability to capture local patterns and hierarchical features in text. CNNs apply convolutional filters to extract essential features like n-grams, making them effective at detecting sentiment in smaller text segments. In [12], CNN is combined with GA optimization. The study [14] uses CNN with ABO and ABC optimization, reaching an impressive accuracy of 99.72%. Additionally, [15] integrates CNN with RBM. Finally, [18] combines CNN with LSTM and GRU. These results demonstrate CNN's strong performance in sentiment analysis tasks, especially when combined with other techniques for optimization or sequence learning. It is especially in short-text or phrase-level sentiment tasks.

2.2.3 Recurrent Neural Network

Recurrent Neural Networks (RNNs) are a type of neural network designed to handle sequential data, making them well-suited for SA tasks where the context and order of words matter. RNNs process input sequences step by step, maintaining a memory of prior words, which allows them to capture temporal dependencies and context over long sequences of text. In sentiment analysis, this capability helps RNNs understand the progression of sentiment across a sentence or document. In [17], an RNN combined with LSTM was used for sentiment analysis, demonstrating its ability to capture the nuances in text sequences. Similarly, in [18], RNNs combined with LSTM and GRU further enhance the model's capacity to process sequential data, illustrating the flexibility and strength of RNN-based architectures in sentiment classification tasks. Despite their strengths, RNNs can suffer from vanishing gradients if not paired with LSTM/GRU.

2.2.4 VGG-16

VGG-16 is a deep CNN architecture known for its simplicity and effectiveness, consisting of 16 layers with weights, including numerous convolutional layers followed by fully connected layers. VGG-16 is typically used for image classification but can also be adapted for text-based tasks like SA. In the context of sentiment analysis, VGG-16 extracts hierarchical features from text by treating sequences of words as spatial patterns, enabling it to capture intricate relationships and sentiment patterns within the data. In [13], VGG-16 was used for sentiment analysis, demonstrating its ability to effectively process and classify text by leveraging its deep convolutional layers to learn meaningful features from the input data. Its performance improves significantly when integrated with pre-trained embeddings.

2.2.5 DenseNet 121

DenseNet-121 is a deep CNN architecture that improves information flow between layers by using dense connections, where each layer is associated with every other layer in a feedforward fashion. This architecture enables more efficient characteristic reuse and helps to avoid the vanishing gradient issue, making it appropriate for tasks like SA. In sentiment analysis, DenseNet-121 can effectively capture complex relationships and contextual information within text by learning hierarchical features from word sequences. In [13], DenseNet-121 was applied to sentiment analysis, showcasing its capability to handle text classification by leveraging its dense connections to learn deeper, more intricate features from the data. Its compact architecture allows deep learning without excessive computational cost.

2.2.6 ResNet-50v2

ResNet-50v2 is a deep CNN architecture that utilizes residual connections to simplify the learning of very deep networks by enabling gradients to flow more easily through the network. This makes ResNet-50v2 particularly effective for complex tasks like sentiment analysis, as it can learn intricate features from large amounts of text data without suffering from issues like vanishing gradients. The residual connections help the model focus on important features while avoiding overfitting, making it robust in handling various sentiment classification tasks. In [13], ResNet-50v2 was used for sentiment analysis, demonstrating its ability to capture hierarchical patterns and contextual relationships in text to effectively classify sentiments. It is ideal for deep architectures that require stable gradient propagation.

2.2.7 Switch Transformer

The Switch Transformer is an advanced architecture that builds upon the Transformer model by incorporating a mixture of experts approach, which allows for more efficient computation while maintaining high performance. For sentiment analysis, the Switch Transformer is proficient in capturing complicated relationships and contextual dependencies in text sequences by leveraging the attention scheme inherent in transformers. In [11], the Switch Transformer was applied to sentiment analysis, where it demonstrated its ability to effectively classify sentiments by efficiently processing input sequences while utilizing the mixture of experts to balance performance and computational efficiency. This architecture shows promise in addressing the challenge of scaling transformer-based models while still achieving strong results in text classification tasks like sentiment analysis. Its efficiency and scalability make it suitable for real-time and large-scale sentiment systems.

2.3 Natural Language Processing-based Sentimental Analysis

Natural Language Processing (NLP)-based sentiment analysis applies various computational techniques to understand and interpret emotions or opinions expressed in text. The different methods are explained in detail as follows. These methods rely on linguistic rules and predefined lexicons to extract sentiment related features from textual content.

2.3.1 TextBlob

TextBlob is a simple and intuitive NLP library that is widely used for sentiment analysis due to its ease of use and effectiveness in analyzing text data. It uses predefined lexical resources, such as the PatternAnalyzer, to perform sentiment analysis by evaluating the polarity and subjectivity of text. This makes TextBlob particularly useful for sentiment classification in smaller or simpler datasets, where more complex models may not be necessary. In [19], TextBlob was applied to sentiment analysis, demonstrating its ability to quickly and effectively classify sentiment in text data without requiring extensive model training or computational resources. Moreover, TextBlob supports language translation, noun phrase extraction, and part-of-speech tagging, adding flexibility to its sentiment analysis capabilities. It is especially beneficial for early-stage prototypes or applications with limited computational capacity.

2.3.2 Bag of Words

The Bag of Words (BoW) method is a simple yet powerful technique for text representation in SA. It enables converting text data into a set of word tokens, where each word in the text is served as a unique feature, and the frequency of its occurrence is used to represent the document. This method disregards word order, grammar, and context but captures the frequency of individual words, making it effective for sentiment classification, especially when combined with ML models. In [20], the Bag of Words approach was applied to sentiment analysis, showcasing its ability to effectively represent text for sentiment classification tasks. Despite its simplicity, BoW can be a strong baseline method for sentiment analysis, especially when large datasets or complex relationships are not involved. It performs best when the vocabulary is controlled and stop words are removed during preprocessing. Although it lacks semantic understanding, BoW remains widely used due to its ease of implementation and strong compatibility with traditional classifiers.

3. Comparative Analysis of Performance

The performance analysis of sentiment analysis models across ML, DL, and NLP approaches highlights varying levels of metrics and adaptability to distinct datasets, as discussed in the following comparison.

3.1 Performance Analysis on Machine Learning

Performance analysis of ML approaches for sentiment analysis reveals distinct strengths and trade-offs among various models. In [7], a combination of rule-based algorithms and an ensemble of ML models, including Naïve Bayes, KNN, LR, Decision Tree, RF, SVM, AdaBoost, and Multilayer Perceptron, achieved an accuracy of 90%, precision of 98%, recall of 86%, and an F1-score of 89%, indicating a strong balance between precision and recall. The research in [8] employed a standalone Naïve Bayes classifier, recorded a lower accuracy of 85.4%, precision of 87.8%, and recall of 86.8%, showing that while effective, individual models may underperform compared to ensembles. In [6], a machine learning approach enhanced with a Capsule Network and Gravitational Search Algorithm achieved a superior accuracy of 91%, outperforming traditional models like Random Forest (0.88), Logistic Regression (0.83), KNN (0.79), and SVC (0.88) within the same study. This suggests that integrating optimization techniques or deep learning components with traditional ML models can significantly improve sentiment classification performance. Additionally, while standalone models like Naïve Bayes can still provide valuable insights, ensemble methods tend to offer more robustness in dealing with variations in the dataset and can outperform individual classifiers, especially in complex scenarios.

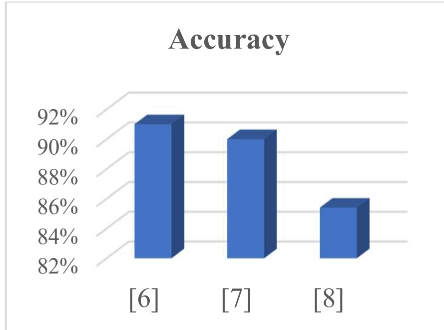

Fig. 2. *Performance analysis on ML in terms of Accuracy*

Fig. 2 shows the performance analysis on ML in terms of Accuracy by comparing three ML-based research studies, including [6], [7]. Among these, the study [6] achieved the highest accuracy at 91%, indicating the model performs best when identifying or classifying this category. The research on [7] follows closely with a slightly lower accuracy of 90%, suggesting similarly strong performance. However, the study [8] shows a noticeable drop in accuracy to 85.4%, implying that the model finds this category more challenging and is more prone to errors when handling it. This pattern may point to differences in the clarity, frequency, or distinctiveness of features associated with each label. Additionally, the varying performance across these studies may also reflect the impact of dataset size, feature engineering techniques, or the inherent difficulty of the classification task itself. The research in [6] and [7] likely benefitted from more refined models or better-preprocessed data, which could explain their higher accuracy rates.

3.2 Performance Analysis on Deep Learning

Performance analysis of DL approaches in sentiment analysis demonstrates the superiority of these models in capturing complicated patterns as well as contextual relationships in text data. For example, in [10], an LSTM model combined with RF-ML achieved a remarkably high accuracy of 0.9919, indicating excellent predictive performance. Similarly, [14] utilized CNN with ABO and ABC Optimization, achieving an outstanding accuracy of 99.72%. In [17], an LSTM-RNN architecture yielded strong performance with an accuracy of 95.8%, precision of 95.4%, recall of 95.6%, and F-measure of 95.2%, reflecting deep learning's strength in sequential data processing. Meanwhile, [9] using word embeddings with LSTM and CNNs, reported accuracy, precision, recall, and F1-score all around 95.2%, indicating robust performance. Even advanced architectures like the Switch Transformer in [11], despite showing a lower accuracy of 0.52, achieved a relatively high recall of 0.8076, highlighting its ability to detect positive instances, though with room for improvement in overall prediction. This variety in performance demonstrates the flexibility of deep learning models to adapt to different architectures and optimizations. While some models achieve near-perfect accuracy, others, like the Switch Transformer, showcase potential for high recall, underscoring deep learning's capacity to focus on specific tasks like identifying positive instances, even if this comes at the cost of overall accuracy.

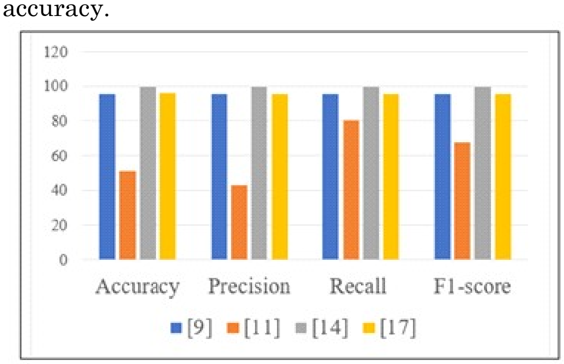

Fig. 3. *Performance analysis on DL in terms of Accuracy, Precision, Recall, and F1-score*

Fig. 3 shows the performance analysis on DL in terms of Accuracy, precision, recall, and F1-score, by comparing three DL-based research studies, including [9], [11], [14], and [17]. Especially [14], which approaches perfect accuracy and consistency. The studies on [9] and [17] also show strong and balanced results, with accuracies of 95.21% and 95.8%, and similarly high values for the other metrics, suggesting the model handles these categories very well. However, performance drops sharply for [11], with low precision (43%) and a lower F1-score (67.9%), indicating frequent misclassifications despite a high recall.

This disparity in performance may point to issues related to the model's optimization or its ability to generalize well across different categories. While the high recall suggests that the model is good at identifying relevant positive instances, its low precision indicates that it tends to classify too many instances as positive, leading to more false positives and a compromised F1-score.

3.3 Performance Analysis on NLP

Performance analysis of NLP techniques in sentiment analysis shows in Fig. 4 that even traditional NLP-based methods can yield competitive results when applied effectively. In [19], the TextBlob library was used for sentiment analysis, achieving a high accuracy of 98%, demonstrating the strength of lexicon-based approaches in capturing sentiment without the need for complex training or large datasets. This highlights the continued relevance and efficiency of rule-based systems, especially when dealing with well-defined, less ambiguous sentiment classifications. Although modern ML and DL models may offer greater flexibility and accuracy in more complex tasks, NLP methods like TextBlob can be highly effective in scenarios where simplicity and computational efficiency are prioritized, offering a strong baseline for sentiment analysis.

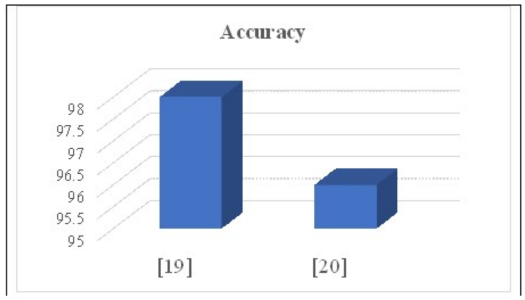

Fig. 4. *Performance analysis on NLP in terms of Accuracy*

Similarly, in [20], the BoW model achieved an accuracy of 96%, reflecting its reliability as a feature extraction method when combined with appropriate classifiers. Their strong performance highlights that with well-chosen preprocessing and modeling, NLP techniques remain valuable tools for sentiment analysis, especially in scenarios where interpretability, simplicity, and lower computational requirements are prioritized. These results underscore that despite the rise of complex deep learning models, traditional NLP methods still hold practical advantages in specific applications, such as real-time systems or environments with limited resources, where model transparency and speed are critical.

4. Conclusion

The comparative analysis of sentiment analysis techniques across 15 reviewed papers revealed the diverse landscape of methods applied to understand textual sentiment. Deep learning models, especially those integrating LSTM, CNN, and optimization algorithms, consistently outperformed traditional ML and basic NLP techniques across accuracy and robustness. However, machine learning models like RF and SVM still offer reliable performance with lower computational demands, and NLP methods such as TextBlob and Bag of Words remain effective for simpler tasks due to their ease of use and interpretability. The choice of method should ultimately depend on the complexity of the dataset, the desired level of accuracy, and the computational resources available. Notably, the trade-off between model complexity and efficiency remains a central consideration, especially in real-world applications where resource constraints and explainability are key concerns. As SA continues to evolve, incorporating hybrid models and leveraging advanced architectures will likely become increasingly essential for achieving high-performance sentiment classification across various domains.

Compliance with Ethical Standards

Conflicts of interest: Authors declared that they have no conflict of interest.

Human participants: The conducted research follows the ethical standards and the authors ensured that they have not conducted any studies with human participants or animals.

References

[1] Cai Guoyong, Lyu Guangrui, Lin Yuming, and Wen Yimin, "Multi-level Deep Correlative Networks for Multi-modal Sentiment Analysis," Chinese Journal of Electronics, vol. 28, no. 6, pp. 1025-1038, 2020.

[2] Snigdha Luthra, Rakesh Kumar, and Meenu Gupta, "A Novel Hybrid Model Based on Sentiment Analysis Using Community Guided Link Prediction for Connecting People With Similar Interest," IEEE Access, vol. 12, pp. 139435-139455, 2024.

[3] Aml Mostafa, Walaa Gad, Tamer Abdelkader, and Nagwa Badr, "Pre-HLSA: Predicting home location for Twitter users based on," Ain Shams Engineering Journal, vol. 13, 2022.

[4] Issa Atoum, "A novel framework for measuring software quality-in-use based on semantic similarity and sentiment analysis of software reviews," Journal of King Saud University – Computer and Information Sciences, vol. 12, pp. 113-125, 2020.

[5] Hongbin Zhang, Jinpeng Wu, Haowei Shi, Ziliang Jiang, Donghong Ji, Tian Yuan, and Guangli Li, "Multidimensional Extra Evidence Mining for Image Sentiment Analysis," IEEE Access, vol. 8, pp. 103619-103634, 2020.

[6] Nalini Chintalapudi, Gopi Battineni, Marzio Di Canio, Getu Gamo Sagaro and Francesco Amenta, "Text mining with sentiment analysis on seafarers' medical documents," International Journal of Information Management Data Insights, vol. 1, 2021.

[7] R. Strubytskyi and N. Shakhovska, "Method and models for sentiment analysis and hidden propaganda finding," Computers in Human Behavior Reports, vol. 12, 2023.

[8] V. Diviya Prabha and R. Rathipriya, "Sentimental Analysis using Capsule Network with Gravitational Search Algorithm," Journal of Web Engineering, vol. 19, no. 5, p. 775–794, 2020.

[9] Azhar Ahmed Bilal, O. Ayhan Erdem, and Sinan Toklu, "Children's Sentiment Analysis From Texts by Using Weight Updated Tuned With Random Forest Classification," IEEE Access, vol. 12, pp. 70089-70104, 2024.

[10] Wang Li, Chaozhu Hu and Youxi Luo, "A Deep Learning Approach with Extensive Sentiment Analysis for Quantitative Investment," Electronics, vol. 12, 2023.

[11] Sangeetha J and Kumaran U, "Sentiment analysis of Amazon user reviews using a hybrid approach," Measurement: Sensors, vol. 27, 2023.

[12] Noemí Merayo, Jesús Vegas, César Llamas, and Patricia Fernández, "Social Network Sentiment Analysis Using Hybrid Deep Learning Models," Applied Sciences, vol. 13, 2023.

[13] Adnan Ishaq, Sohail Asghar and Saira andleeb Gillani, "Aspect-Based Sentiment Analysis Using a Hybridized Approach Based on CNN and GA," IEEE Access, vol. 8, pp. 135499-135512, 2020.

[14] Meshrif Alruily, "Sentiment analysis for predicting stress among workers and classification utilizing CNN: Unveiling the mechanism," Alexandria Engineering Journal, vol. 81, pp. 360-370, 2023.

[15] Thilak Bellam and P. Lakshmi Prasanna, "Novel Deep-Learning based sentimental approach to identifying the fake news in social networking media based smart application," Measurement: Sensors, vol. 33, 2024.

[16] Ganesh Chandrasekaran, Naaji Antoanela, Gabor Andrei, Ciobanu Monica, and Jude Hemanth, "Visual Sentiment Analysis Using Deep Learning Models with Social Media Data," Applied Sciences, vol. 12, pp. 1-23, 2022.

[17] S. Muhan=mmad Ahmed Hassan Shah, Syed Faizan Hussain Shah, Asad Ullah, Atif Rizwan, Ghada Atteia, and Maali Alabdulhafith, "Arabic Sentiment Analysis and Sarcasm Detection Using Probabilistic Projections-Based Variational Switch Transformer," IEEE Access, vol. 11, pp. 67865-67881, 2023.

[18] Kakuthota Rakshitha, Ramalingam H M, M Pavithra, and Advi H D, Maithri Hegde, "Sentimental analysis of Indian regional languages on social media," Global Transitions Proceedings, vol. 2, pp. 414-420, 2021.

[19] Sangamesh Hosgurmath, Vishwanath Petli, and V.K. Jalihal, "An omicron variant tweeter sentiment analysis using NLP technique," Global Transitions Proceedings, vol. 3, pp. 215-219, 2022.

[20] Shaymaa E. Sorour, Abdulrahman Alojail, Amr El-Shora, Ahmed E. Amin and Amr A. Abohany, "A Hybrid Deep Learning Approach for Enhanced Sentiment Classification and Consistency Analysis in Customer Reviews," Mathematics, vol. 12, pp. 1-38, 2024.