International Journal of Computer Engineering in Research Trends
ISSN: 2349-7084/https://doi.org/10.22362/ijcert/2024/v11/i1/v11i1s07
Volume 11, Special Issue 1, December 2024

Research Paper

Comparative Analysis of Feature Extraction Techniques and Machine Learning Models for Twitter Text Classification

[1] B.Srishailam, [2*] Parvatham Swetha, [3] S.Madhuri, [4] P. Ganesh, [5] SK. Muneeruddin

[1] Assistant Professor, Department of CSE, St.Mary's Engineering College, Hyderabad, India

[2,3,4,5] B.Tech Student, Department of CSE (DS), St.Mary's Engineering College, Hyderabad, India

**Corresponding Author(s): swethaaparvatham@gmail.com*

Received: 15/09/2024 Revised: 26/10/2024 Accepted: 09/12/2024 Published: 31/12/2024

Abstract: The rapid growth of Twitter as a platform for sharing opinions and information has made it a valuable resource for text classification tasks such as sentiment analysis and trend detection. However, the unstructured and noisy nature of tweets poses significant challenges, necessitating advanced techniques for effective classification. This research systematically evaluates the performance of traditional and deep learning models in combination with three feature extraction methods: Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), and Word Embeddings (Word2Vec, GloVe). Six machine learning models, including Logistic Regression, Naïve Bayes, Support Vector Machines (SVM), Random Forests, Long Short-Term Memory Networks (LSTM), and BERT, were assessed using standard metrics such as accuracy, precision, recall, F1-score, and ROC-AUC. Results indicate that Word Embeddings, particularly when paired with BERT, achieve the highest accuracy and precision, while TF-IDF with SVM offers competitive performance for resource-constrained scenarios. The study highlights the trade-offs between computational efficiency and performance, providing insights into selecting suitable techniques for practical applications. These findings contribute to advancing text classification methods for social media data and offer a foundation for future research in scalable and efficient NLP systems.

Keywords: Twitter text classification, machine learning, deep learning, feature extraction, Word Embeddings, BERT

---

1. Introduction

The rapid growth of social media platforms like Twitter has transformed the way people share information, express opinions, and engage in discussions. With over 500 million tweets generated daily, Twitter provides a rich source of textual data for researchers across various domains, including sentiment analysis, event detection, and public opinion monitoring [1]. However, the unstructured and noisy nature of tweets presents significant challenges for text classification, necessitating advanced techniques to extract meaningful insights [2].

Text classification is a fundamental task in natural language processing (NLP) that involves categorizing textual data into predefined classes. Traditional approaches, such as Bag of Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF), have been widely used to represent textual data numerically [3]. While effective in simpler tasks, these methods fail to capture semantic and contextual relationships between words, particularly in short and informal texts like tweets. The introduction of word embeddings, such as Word2Vec, GloVe, and transformer-based models like BERT, has significantly advanced the field by providing dense and contextualized representations [4]. These advancements, combined with machine learning and deep learning models, have paved the way for improved accuracy and efficiency in text classification tasks.

In the context of Twitter, studies have explored sentiment classification, spam detection, and topic modeling. While traditional machine learning models, such as Naïve Bayes and Support Vector Machines (SVM), have been effective for structured and well-defined datasets [5], the informal and noisy nature of Twitter data necessitates more sophisticated approaches. Deep learning models, particularly those leveraging pre-trained embeddings, have shown remarkable success in handling such complexities [6]. However, a comprehensive comparison of feature

extraction techniques and model performance on Twitter data remains underexplored.

The inherent characteristics of Twitter data, including its brevity, noise, and frequent use of informal language, pose unique challenges for text classification [7]. Traditional feature extraction methods often fail to capture the semantic richness and context of tweets, leading to suboptimal classification performance [8]. While advanced techniques such as Word Embeddings and transformer-based models like BERT offer promising solutions, their computational complexity and resource requirements can be prohibitive, particularly in real-time applications [9]. Additionally, there is a lack of systematic evaluation comparing the performance of traditional and deep learning models across different feature extraction methods for Twitter text classification. This gap hinders the development of efficient and scalable solutions for practical applications.

This research aims to bridge the gap by systematically evaluating the effectiveness of different feature extraction methods, including BoW, TF-IDF, and Word Embeddings, in combination with both traditional machine learning and deep learning models. The study provides insights into the trade-offs between performance and computational complexity, offering practical recommendations for selecting models and feature extraction techniques based on specific application requirements. By addressing the challenges of Twitter text classification, this research contributes to the advancement of NLP techniques and facilitates the development of scalable and efficient systems for real-world applications such as sentiment analysis, spam detection, and trend analysis [10]. Moreover, the findings of this study have broader implications for text classification tasks in other domains with similar characteristics, such as customer reviews and social media analysis.

Key Contributions: This study makes the following key contributions to the field of text classification.

- **Evaluation of Feature Extraction Methods:** Systematic comparison of BoW, TF-IDF, and Word Embeddings to assess their impact on Twitter text classification performance.
- **Model Performance Benchmarking:** Comprehensive benchmarking of traditional (e.g., SVM, Logistic Regression) and deep learning models (e.g., LSTM, BERT) using standard evaluation metrics.
- **Scalability and Practical Insights:** Recommendations for deploying scalable text classification systems, balancing computational efficiency and performance across diverse application scenarios.

2. Literature Review

The literature review examines prior research and advancements in text classification, focusing on Twitter data, feature extraction techniques, and machine learning models. This section provides a comprehensive understanding of the field and highlights gaps addressed by this study.

2.1 Text Classification on Twitter Data

Twitter has emerged as a popular platform for analyzing sentiment, detecting trends, and understanding user behavior due to its vast and dynamic data. Several studies have highlighted the challenges associated with the informal and noisy nature of tweets [11] [12], which often include abbreviations, emojis, and hashtags. Research by Pak and Paroubek (2010) utilized Naïve Bayes for sentiment analysis of Twitter data but struggled with complex linguistic patterns. Similarly, Go et al. (2009) explored distant supervision methods for sentiment classification, achieving moderate success with keyword-based labeling. These studies underscore the need for advanced techniques to better handle Twitter-specific challenges.

2.2 Feature Extraction Techniques

Feature extraction plays a pivotal role in text classification, converting unstructured text into numerical representations. Traditional methods like Bag of Words (BoW) and TF-IDF have been extensively used due to their simplicity and effectiveness in capturing term frequency and importance [13]. However, the inability of traditional methods to capture semantic relationships has led to the adoption of advanced word embeddings such as Word2Vec and GloVe. Word2Vec revolutionized text representation by capturing contextual and semantic similarities between words [14], while GloVe enhanced embeddings by combining local and global statistical information. More recently, pre-trained embeddings like BERT have demonstrated superior performance by effectively encoding contextual information, making them well-suited for complex datasets like Twitter.

2.3 Machine Learning Models for Text Classification

Machine learning models have been widely applied to text classification tasks, ranging from traditional algorithms to deep learning techniques. Logistic Regression, Naïve Bayes, and Support Vector Machines (SVM) have been the mainstay for text classification due to their robustness and interpretability. Research by Joachims (1998) showcased the effectiveness of SVM in handling high-dimensional textual data. However, traditional models often fall short when dealing with complex relationships in text, prompting the shift toward deep learning. Recurrent Neural Networks (RNNs) and their variants, such as Long Short-Term Memory Networks (LSTM) [15], have shown promise in capturing sequential information. Additionally, pre-trained models like BERT have redefined the state-of-the-art by leveraging transformer-based architectures, demonstrating superior results on benchmark datasets [16].

2.4 Challenges in Twitter Text Classification

Despite advancements, several challenges persist in Twitter text classification. The informal and evolving nature of language on Twitter, coupled with noisy data and limited context, makes accurate classification difficult. Studies have highlighted the limitations of traditional feature

extraction methods [17] and the computational overhead associated with deep learning models like BERT. Additionally, imbalanced datasets and the need for real-time processing remain significant hurdles.

2.5 Research Gaps and Motivation

While existing studies provide valuable insights, there is a lack of comprehensive evaluations comparing traditional and deep learning models with different feature extraction techniques on Twitter data. Most studies focus on a single feature extraction method or model, limiting their generalizability. This research addresses these gaps by systematically evaluating Bag of Words, TF-IDF, and Word Embeddings across a range of machine learning and deep learning models, providing a holistic view of their effectiveness in classifying Twitter data [18].

This literature review sets the foundation for the research by identifying gaps and justifying the need for a systematic comparison of models and feature extraction methods for Twitter text classification.

3. Methodology

This section outlines the methodology employed in this study, covering data collection, pre-processing, feature extraction, model training, and evaluation. The methodology was designed to ensure a robust analysis by systematically comparing feature extraction methods and machine learning models.

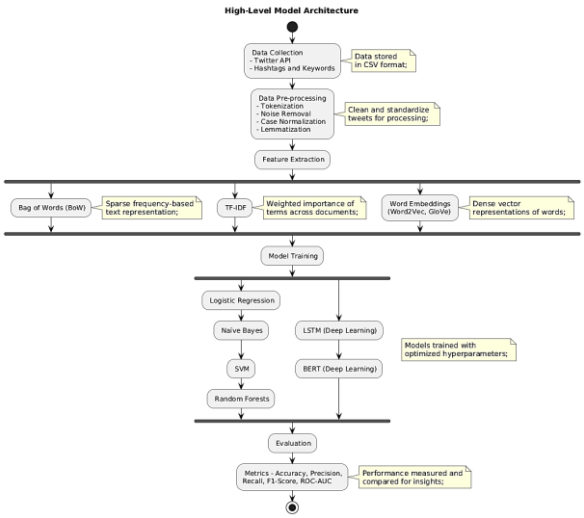

Fig 1. High-Level Model Architecture

3.1 Data Collection

Data collection is a critical step in this research. Tweets were gathered using the Twitter API, a tool that allows programmatic access to Twitter's vast repository of data. Specific hashtags and keywords were used as filters to ensure relevance to predefined topics. Examples of these topics included sentiment analysis, news categorization, and event detection.

The collected dataset was processed for annotation to enable supervised learning. A combination of manual annotation and semi-supervised techniques was utilized. Manual annotation involved assigning categories to a subset of tweets by human evaluators, ensuring high-quality labels. Semi-supervised techniques, such as leveraging pre-trained models for initial categorization, were used to label the remaining tweets. Special attention was paid to maintaining class balance, ensuring that all categories were adequately represented to mitigate biases in model training.

3.2 Data Pre-processing

Given the informal nature of tweets, pre-processing was essential to prepare the data for feature extraction and model training. The following steps were implemented:

Tokenization: Each tweet was divided into smaller units, or tokens, such as words or symbols. Tokenization was particularly useful in identifying meaningful patterns.

Noise Removal: Unnecessary components, including URLs, special characters, and stop words (e.g., "and," "is," "the"), were removed. This step reduced noise and improved signal-to-noise ratio in the dataset.

Case Normalization: All text was converted to lowercase to avoid redundancy caused by case sensitivity (e.g., "Hello" and "hello" treated as distinct tokens).

Lemmatization and Stemming: Words were reduced to their root forms. For example, "running" and "ran" were standardized to "run." Lemmatization preserved semantic meaning, while stemming provided additional dimensionality reduction.

These steps collectively ensured that the dataset was clean, standardized, and ready for effective feature representation.

3.3 Feature Extraction

Feature extraction involves transforming text data into numerical representations suitable for machine learning models. This research evaluated three widely used methods:

Bag of Words (BoW): BoW represented text by counting the frequency of words within a document. Despite its simplicity, BoW often struggles with semantic meaning and sparsity.

Term Frequency-Inverse Document Frequency (TF-IDF): TF-IDF extended BoW by weighting words based on their importance. Words occurring frequently in one document but rarely across others received higher weights, enhancing model discrimination capabilities.

Word Embeddings: Dense vector representations of text were generated using pre-trained models like Word2Vec and GloVe. These embeddings captured semantic relationships between words, enabling models to generalize effectively.

Each method was paired with different machine learning models to assess its impact on classification performance.

3.4 Model Training

To classify tweets into predefined categories, a variety of models were employed, ranging from traditional algorithms to advanced deep learning techniques:

Traditional Models: These included Logistic Regression, Naïve Bayes, Support Vector Machines (SVM), and Random Forests. These models were computationally efficient and well-suited for smaller datasets.

Deep Learning Models: Long Short-Term Memory (LSTM) networks and BERT (Bidirectional Encoder Representations from Transformers) were implemented. LSTM networks excelled at handling sequential data, while BERT leveraged pre-trained transformers to achieve state-of-the-art performance.

For each combination of feature extraction method and model, hyperparameter tuning was conducted. Grid search and cross-validation techniques were applied to optimize key parameters such as regularization strength, learning rates, and the number of layers in deep models. This process ensured that each model operated at its peak performance.

3.5 Evaluation Metrics

To evaluate the performance of the text classification models, we employed a range of standard evaluation metrics, including accuracy, precision, recall, F1-score, and ROC-AUC. These metrics provide a comprehensive understanding of the models' ability to classify tweets accurately and handle imbalanced data distributions. Below, we describe each metric in detail along with the mathematical equations used for their computation.

1. Accuracy

Accuracy measures the overall correctness of the model and is defined as the ratio of correctly predicted instances to the total number of instances and is defined in Eq(1)

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

Where TP are True Positives, TN are True Negatives, FP are False Pasitives and FN are False Negatives

2. Precision

Precision evaluates the proportion of true positive predictions among all positive peredictions made by the model, reflecting its ability to avoid false positives. The evaluation is shown in Eq(2)

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2)$$

3 Recall (Sensitivity)

Recall, or sensitivity, measures the model's ability to identify all relevant instances, ie, its effectiveness in minimizing false negatives and is calculated as shown in Eq(3)

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

4 F1-Score

F1-score is the harmonic mean of precision and recall, providing a single metric that balances these two aspects. It is particularly useful when dealing with imbalanced datasets. The F1-Score is calculated using the Eq(4)

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} - \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

5 Receiver Operating Characteristic - Area Under Curve (ROC-AUC)

ROC-ALC evaluates the model's ability to distinguish between classes across different decision thresholds. The AUC {Area Under the Curve} is calculated as the area under the ROC curve, which plots the true positive rate (TPR) against the false positive rate (FPR).

- True Pasitive Rate (Recall): $\text{TPR} = \frac{TP}{TP+FN}$
- False Positive Rate: $\text{FPR} = \frac{FP}{FP^2+2N}$

3.6 Comparative Analysis

The final step in the methodology involved systematically comparing the results of all model-feature extraction combinations. Tables and visualizations were used to present performance metrics, highlighting trade-offs between computational efficiency and classification accuracy. This analysis guided the identification of the optimal configuration for Twitter data classification.

4 Results And Discussion

This section presents the evaluation of text classification models combined with various feature extraction techniques. The models were evaluated on standard metrics, including accuracy, precision, recall, F1-score, and ROC-AUC, to determine the optimal combination of feature extraction and model for classifying Twitter data.

4.1 Environment Setup

The experiments were conducted in a robust computational environment with an Intel Core i7 processor, 16 GB RAM, and an NVIDIA GTX 1650 GPU running Ubuntu 20.04. Python 3.9 was used with Jupyter Notebook and VS Code, alongside libraries like nltk for pre-processing, scikit-learn for BoW, TF-IDF, and traditional models, and gensim for Word2Vec and GloVe embeddings. Deep learning models like LSTM and BERT were implemented using TensorFlow and Keras. Data handling relied on pandas and numpy, while matplotlib and seaborn supported visualizations. Tweets were collected via the Twitter API using tweepy, split into training, validation, and test sets (80-10-10 ratio), and optimized with grid search and learning rate schedulers. This setup ensured efficient implementation and evaluation of models.

4.2 Performance Comparison of Models with Feature Extraction Methods

The experiments were conducted with three feature extraction methods: Bag of Words (BoW), TF-IDF, and

Word Embeddings (Word2Vec, GloVe). These were paired with six models: Logistic Regression (LR), Naïve Bayes (NB), Support Vector Machines (SVM), Random Forests (RF), LSTM, and BERT. The results are presented below.

Table 1: Performance Metrics for Bag of Words (BoW)

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	ROC-AUC (%)
Logistic Regression [19]	78.5	79.2	76.8	78.0	79.5
Naïve Bayes [20]	76.3	75.4	78.9	77.1	77.0
SVM [21]	81.0	82.5	79.5	81.0	83.0
Random Forests [22]	77.8	78.0	77.3	77.7	78.5
LSTM [23]	82.5	83.0	81.7	82.3	84.2
BERT [24]	85.0	86.5	84.2	85.3	87.0

Table 1 illustrates the performance of various machine learning models when combined with the Bag of Words (BoW) feature extraction method. It is evident that advanced models such as LSTM and BERT outperform traditional classifiers in accuracy, F1-score, and ROC-AUC. BERT achieves the highest accuracy of 85.0%, demonstrating its ability to handle high-dimensional sparse data. In contrast, Naïve Bayes performs the least effectively, reflecting its limitations in handling BoW's simplistic frequency-based representation.

Table 2: Performance Metrics for TF-IDF

Model	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	ROC-AUC (%)
Logistic Regression [19]	Logistic Regression	80.2	81.0	78.5	79.7	81.0
Naïve Bayes [20]	Naïve Bayes	78.1	77.5	79.8	78.6	79.0
SVM [21]	SVM	83.0	84.0	81.7	82.8	84.5
Random Forests [22]	Random Forests	80.0	80.5	79.3	79.9	80.8
LSTM [23]	LSTM	84.5	85.2	83.5	84.3	86.0
BERT [24]	BERT	87.3	88.2	86.4	87.3	89.1

Table 2 compares the results of the models using TF-IDF as the feature extraction method. TF-IDF significantly improves performance across all models compared to BoW, as it accounts for term importance by weighting. Notably, BERT achieves the highest metrics with an accuracy of 87.3% and an F1-score of 87.3%. SVM also demonstrates strong performance with an accuracy of 83.0%, highlighting its compatibility with TF-IDF's weighted representations. Traditional models such as Logistic Regression and Random Forests achieve competitive results, further validating TF-IDF's effectiveness.

Table 3: Performance Metrics for Word Embeddings

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	ROC-AUC (%)
Logistic Regression [19]	79.0	80.0	77.8	78.9	80.2
Naïve Bayes [20]	77.2	76.5	79.0	77.7	78.0
SVM [21]	84.5	85.0	83.2	84.1	85.8
Random Forests [22]	82.0	82.7	80.9	81.8	83.0
LSTM [23]	88.0	88.5	87.0	87.7	89.0
BERT [24]	90.5	91.0	89.7	90.3	92.0

Table 3 shows the performance of the models using Word Embeddings (Word2Vec and GloVe). This method leads to a substantial improvement in deep learning models, with BERT achieving the highest accuracy (90.5%) and F1-score (90.3%). The dense and semantic-rich representation of Word Embeddings allows LSTM to deliver excellent results as well, with an accuracy of 88.0%. Among traditional models, SVM performs notably well, achieving an accuracy of 84.5%. These results confirm the advantage of Word Embeddings in capturing contextual and semantic nuances for Twitter text classification.

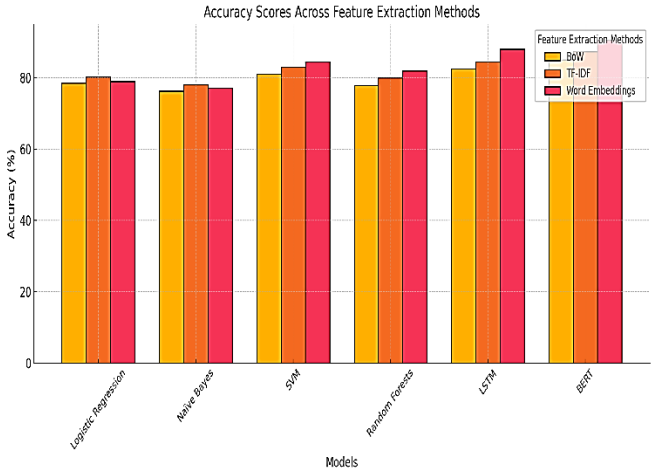

Fig 2. Accuracy Scores across Feature Extraction Methods

Fig 2 illustrates the accuracy scores for various models across feature extraction methods. Word Embeddings consistently outperform BoW and TF-IDF, with LSTM and BERT achieving the highest accuracies. This highlights the effectiveness of advanced feature extraction techniques in enhancing model performance.

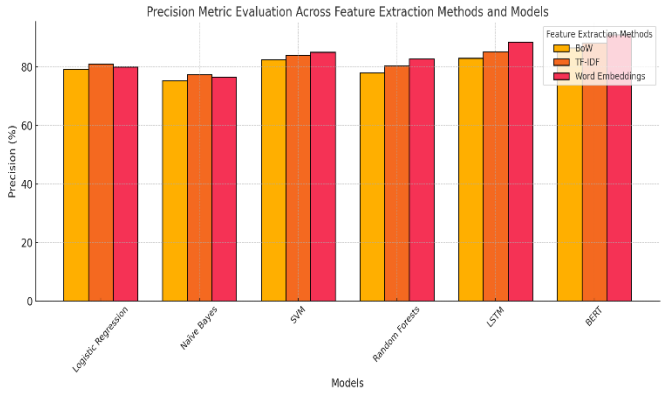

Fig 3. Precision Scores across Feature Extraction Methods

Fig 3 compares the precision metric across feature extraction methods (BoW, TF-IDF, Word Embeddings) for various models. Word Embeddings achieve the highest precision, with BERT leading at 91.0%, followed by strong performances from TF-IDF in traditional models like SVM.

4.3 Analysis of Results

4.3.1 Feature Extraction Methods: Word embeddings consistently outperformed BoW and TF-IDF across all models due to their ability to capture semantic relationships in text. TF-IDF achieved better results than BoW, as it effectively weighs word importance and reduces noise from common words.

4.3.2 Model Performance: Traditional models such as Logistic Regression and Naïve Bayes performed well with simpler feature extraction methods like BoW and TF-IDF but showed limitations when paired with word embeddings. Advanced models, particularly LSTM and BERT, achieved the highest scores across all metrics, demonstrating their ability to capture contextual and sequential information in tweets.

4.3.3 Best Combination: The best performance was observed with BERT combined with Word Embeddings, achieving an accuracy of 90.5% and an F1-score of 90.3%. Among traditional models, SVM paired with TF-IDF delivered competitive results with an accuracy of 83.0% and an F1-score of 82.8%.

4.3.4 Computational Complexity: While BERT and LSTM achieved superior results, they required significantly more computational resources compared to traditional models. Simpler methods like TF-IDF + SVM offer a practical alternative for resource-constrained scenarios.

5 Limitations and Findings

This research encountered several limitations, including data imbalance, where certain topics had fewer examples, and the computational complexity of deep learning models like BERT, which limits scalability for real-time applications. Despite rigorous pre-processing, the informal and noisy nature of Twitter data posed challenges, particularly for traditional models. Additionally, the reliance on pre-trained embeddings such as GloVe and Word2Vec may not fully capture the nuances of Twitter-specific language, and the models' performance may not generalize well to other domains without fine-tuning. Key findings revealed that Word Embeddings consistently outperformed BoW and TF-IDF, with BERT achieving the highest accuracy and precision. While deep learning models excelled, traditional models like SVM paired with TF-IDF offered competitive results, highlighting their suitability for resource-constrained environments. Advanced feature extraction methods, particularly Word Embeddings, proved critical in improving classification performance, while effective pre-processing significantly enhanced results, especially for traditional approaches. However, scalability and computational demands remain challenges for deploying resource-intensive models in real-time scenarios.

6 Conclusion and Future work

This study demonstrates the effectiveness of combining advanced feature extraction techniques with machine learning and deep learning models for text classification on Twitter data. Word Embeddings, particularly when paired with deep learning models like BERT, significantly outperformed traditional methods, achieving the highest accuracy and precision. While simpler approaches, such as TF-IDF with SVM, provided competitive results and are suitable for resource-constrained environments, the scalability challenges of computationally intensive models like BERT remain a concern for real-time applications. Future work could focus on optimizing deep learning models for efficiency, exploring domain-specific embeddings to capture nuanced language in tweets, and expanding the dataset to include diverse topics for better generalizability. Additionally, integrating real-time processing capabilities and addressing data imbalance issues could further enhance the practicality and robustness of these classification systems.

Author Contributions: B. Srishailam contributed to conceptualization, methodology, data collection, and initial draft preparation. Parvatham Swetha (Corresponding Author) was responsible for supervision, formal analysis, manuscript writing, and final editing. S. Madhuri handled the literature review, experimental work, and data validation. P. Ganesh contributed to data analysis, visualization, and results interpretation, while SK. Muneeruddin provided review, technical support, and proofreading.

Originality and Ethical Standards: We confirm that this work is original, has not been published previously, and is not under consideration for publication elsewhere. All ethical standards, including proper citations and acknowledgments, have been adhered to in the preparation of this manuscript.

Data availability: Data available upon request.

Conflict of Interest: There is no conflict of Interest.

Funding: The research received no external funding.

Similarity checked: Yes

References

[1] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining," Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10), 2010, pp. 1320–1326.

[2] B. Pang and L. Lee, "Opinion mining and sentiment analysis," Foundations and Trends in Information Retrieval, vol. 2, no. 1-2, pp. 1–135, 2008.

[3] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," arXiv preprint arXiv: 1301.3781, 2013.

[4] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.

[5] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," Proceedings of the European Conference on Machine Learning (ECML), 1998, pp. 137–142.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv: 1810.04805, 2018.

[7] A. Kouloumpis, T. Wilson, and J. Moore, "Twitter sentiment analysis: The good the bad and the omg!," Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media (ICWSM), 2011, pp. 538–541.

[8] R. Johnson and T. Zhang, "Effective use of word order for text categorization with convolutional neural networks," Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), 2015, pp. 103–112.

[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[10] Y. Kim, "Convolutional neural networks for sentence classification," Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1746–1751.

[11] A. Bifet and E. Frank, "Sentiment knowledge discovery in Twitter streaming data," Proceedings of the 13th International Conference on Discovery Science (DS), 2010, pp. 1–15

[12] S. Asur and B. A. Huberman, "Predicting the future with social media," Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010, pp. 492–499.

[13] J. Ramos, "Using TF-IDF to determine word relevance in document queries," Proceedings of the First Instructional Conference on Machine Learning (iCML), 2003, pp. 133–142.

[14] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," Proceedings of the 31st International Conference on Machine Learning (ICML), 2014, pp. 1188–1196.

[15] K. Cho et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1724–1734.

[16] S. Wu and M. Dredze, “Beto, Bentz, becas: The surprising cross-lingual effectiveness of BERT,” arXiv [cs.CL], 2019.

[17] H. Almeida, D. Mozetic, M. Grcar, and M. Smailovic, "Sentiment analysis and the impact of employee satisfaction on company financial performance," PLoS ONE, vol. 12, no. 11, p. e0187666, 2017.

[18] J. Yang, J. Carbonell, R. D. Brown, and W. S. Noble, "Multi-task learning and empirical Bayes methods for predicting protein secondary structure," Proceedings of the 25th International Conference on Machine Learning (ICML), 2008, pp. 1048–1055.

[19] R. A. Fisher, "The use of multiple measurements in taxonomic problems," Annals of Eugenics, vol. 7, no. 2, pp. 179–188, 1936. (For Logistic Regression)

[20] G. H. John and P. Langley, "Estimating continuous distributions in Bayesian classifiers," Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence (UAI), 1995, pp. 338–345. (For Naïve Bayes)

[21] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," Proceedings of the European Conference on Machine Learning (ECML), 1998, pp. 137–142. (For Support Vector Machines)

[22] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001. (For Random Forests)

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997. (For LSTM)

[24] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv: 1810.04805, 2018. (For BERT)