

# The impact of feature extraction techniques on the performance of text data classification models

Abdallah Maiti<sup>1</sup>, Abdallah Abarda<sup>2</sup>, Mohamed Hanini<sup>1</sup>

<sup>1</sup>Mobility and Modeling Laboratory: IR2M, Faculty of Sciences and Techniques, Computer, Networks, Hassan First University of Settat, Settat, Morocco

<sup>2</sup>Mathematical Modeling and Economic Calculation Laboratory: LM2CE, Faculty of Economic Sciences and Management, Hassan First University of Settat, Settat, Morocco

## Article Info

### Article history:

Received Jan 13, 2024

Revised Mar 24, 2024

Accepted Mar 30, 2024

### Keywords:

BERT

Classification

Feature extraction

LSTM

Sentiment analysis

Textual data

## ABSTRACT

Sentiment analysis is a crucial discipline that focuses on the interpretation of feelings and points of view in textual data. Our study aims to assess the impact of different feature extraction methods on the accuracy of opinion research models. Techniques such as bag-of-words (BoW), term frequency-inverse document frequency (TF-IDF), Word2Vec, global vectors (GloVe) and bidirectional encoder representations from transformers (BERT) were used with three machine learning algorithms and three deep learning networks as classifiers. The IMDB movie review dataset was used for evaluation. The results showed that combining BERT with LSTM, CNN and RNN improved performance, achieving an accuracy rate of 94%, precision of 94.14%, recall of 93.27% and an F1 score of 89.33%. These results highlight the significant contribution of BERT to model performance, outperforming other feature extraction techniques in text classification. The study concludes that the fusion of BERT and LSTM significantly improves model accuracy for opinion retrieval, recommending BERT as the main feature extraction method for optimizing performance in NLP tasks.

*This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.*



## Corresponding Author:

Abdallah Maiti

Mobility and Modeling Laboratory: IR2M, Faculty of Sciences and Techniques, Computer, Networks

Hassan First University of Settat

26000-Settat, Morocco

Email: sitizaiton@umk.edu.my

## 1. INTRODUCTION

The rise of social media and technological advancements have resulted in a significant increase in unstructured textual data. This is especially apparent on large platforms like Twitter, which has become a major information source with over 336 million active users per month, producing around 500 million tweets per day [1], [2]. This data provides a rich source of information, reflecting user opinions, emotions, and trends on a wide range of topics. However, converting this vast amount of data into useful information, particularly for text classification and sentiment analysis, remains a significant challenge.

Sentiment analysis in text has recently become a major focus for many data science researchers. The goal is to create models that can accurately determine the sentiment polarity in a given text, whether positive or negative. This capability is crucial in various areas such as business intelligence, customer service, and understanding public opinion on social networks [3]. Despite the progress, challenges persist in identifying the most effective feature extraction techniques for this task and their optimal combination with machine and deep learning models.

Numerous recent studies have explored different feature extraction and classification techniques for sentiment analysis, often combining traditional methods like bag-of-words (BoW) and term frequency-inverse document frequency (TF-IDF) with newer approaches like Word2Vec, global vectors (GloVe), and bidirectional encoder representations from transformers (BERT). For instance, a study by Demircan *et al.* [4] found that combining BoW with support vector machine (SVM) achieved 86.7% accuracy in classifying Turkish customer reviews on e-commerce platforms. However, further research by Cruz *et al.* [5] showed modest performance of BoW in detecting online hate speech. In 2023, Haque *et al.* [6] used BoW and TF-IDF to train a deep learning classifier on Bengali language comments on social networks, where the LSTM model outperformed others, achieving 85.8% accuracy and 86% F1 scores. Similarly, a study by Hassan *et al.* [7] used TF and TF-IDF feature extraction techniques to classify Twitter messages as credible or non-credible. This research showed significant improvements in results, including an accuracy of 84.9% and an F-1 score of 89% on an English dataset, and an accuracy of 73.2% and an F-1 score of 78.5% on an Arabic dataset.

The adoption of word embedding techniques in natural language processing (NLP) has seen a considerable rise in recent years. A study conducted by Mostafa [8] in 2021 led to the creation of a sentiment analysis model that utilized the Word2Vec feature extraction method in conjunction with machine learning algorithms to gauge the sentiments expressed by students during the challenging pandemic period. This research involved a comparative assessment of three distinct classifiers, namely Naïve Bayes (NB), SVM, and decision tree. The analysis conclusively determined the dominance of the SVM-based model, which achieved an important accuracy of 87.6%. In the biomedical sector, Sagnika *et al.* [9] compared the performance of GloVe with other word integration techniques, discovering high accuracy, especially with the SVM algorithm. Sentiment analysis, particularly on social networks, has gained increasing importance, with BERT emerging as a crucial tool and effective feature extraction technique. Research, such as that conducted by Kaliyar *et al.* [10], has demonstrated the effectiveness of the BERT technique when combined with classifiers. In particular, their model, which integrates the BERT technique with CNN and LSTM networks for fake news detection, has outperformed existing models, achieving an impressive accuracy of 98.90% for CNN and 97.55% for LSTM.

Existing literature reviews have observed a common practice of using feature extraction techniques without comprehensive justification, neglecting their impact on classification performance. Our study distinguishes itself by specifically evaluating the effects of these techniques on machine learning and deep learning models in text classification, unlike previous works. Using the IMDB dataset, we carried out extensive experiments, incorporating various algorithms (SVM, logistic regression (LR), NB) and neural networks (CNN, recurrent neural network (RNN), LSTM). We also integrated five text feature extraction methods (TF-IDF, BoW, Word2Vec, GloVe, BERT) to assess multiple combinations and identify optimal strategies. In contrast to previous studies, our approach aims to systematically identify the best feature extraction method, addressing a gap in the literature. By highlighting optimal classifiers, we provide valuable insights and recommendations for practitioners. In summary, our study makes a significant contribution by proposing a rigorous methodology to evaluate feature extraction methods, enhancing understanding of their impact on the performance of ML and DL models.

This study is driven by a critical review of existing literature on text-based sentiment analysis, with the aim to analyze how the techniques of feature extraction impact the performance of sentiment analysis models. The work's key contributions are as follows:

- Investigating the impact of feature extraction techniques: this involves illustrating how modifications to these techniques can influence the accuracy of sentiment prediction, covering both conventional ML methods and sophisticated DL strategies.
- Creating a sentiment polarity classification model: the aim is to build a robust model capable of accurately identifying sentiment polarity in text, categorizing it as either positive or negative, with significant real-world applications.
- Determining the optimal combination of techniques: the goal is to identify the most efficient combination of methods for extracting features from text data and choosing the best classifier among various ML and DL options, providing practical advice for professionals engaged in similar sentiment analysis tasks.
- Experimenting with IMDB movie review data: using the IMDB dataset, which classifies movie reviews based on sentiment, to assess the effectiveness of both models and feature extraction techniques.

The remainder of the article is organized as follows: section 2, titled "Method", discusses the proposed operational methodology and outlines the main steps of the process to be followed, including data collection and the analytical tools employed. Section 3, titled "Results and Discussion", provides an account of the experiments conducted and presents the results achieved, while facilitating a comprehensive discussion

of these findings. Finally, section 4, titled “Conclusion”, analyzes and interprets the results obtained, while proposing directions for future research.

## 2. METHOD

In this research, our objective was to evaluate the effect of diverse feature extraction techniques on the efficiency of sentiment classification models, using the IMDB film review dataset for assessment. Our methodological approach includes the following stages (refer to Figure 1):

- Data gathering: we assembled a dataset of IMDB film reviews from the Kaggle website, consisting of a total of 50,000 reviews, with an equilibrium between positive and negative sentiments.
- Cleaning and pre-processing: We cleaned and pre-processed the text data, eliminating undesired elements such as spaces, numbers, HTML tags, URLs, and special characters. We employed standard text pre-processing techniques, including punctuation removal, lower-casing, removal of empty words, URLs, and unwanted symbols, as well as truncation and lemmatization.
- Feature extraction: we employed five distinct feature extraction methods: TF-IDF, BoW, Word2Vec, GloVe, and BERT. These techniques transform text into numerical vectors, enabling their use in classification models.
- Sentiment classification: we utilized three machine learning algorithms (SVM, LR, and NB) and three deep learning networks (CNN, RNN, and LSTM) to classify the sentiments of film reviews. These models were trained and evaluated on pre-processed data to ascertain their sentiment classification performance.
- Evaluation and analysis: we assessed the performance of the sentiment classification models using a set of metrics (recall, precision, accuracy, and F1 score). We analyzed the results to determine the best classification algorithms and feature extraction techniques in our context.

This thorough methodology has allowed us to compare and assess the efficiency of various feature extraction techniques in enhancing the performance of models based on machine learning and deep learning approaches in sentiment classification tasks based on textual data. Figure 1 depicts the overall structure of the proposed model.

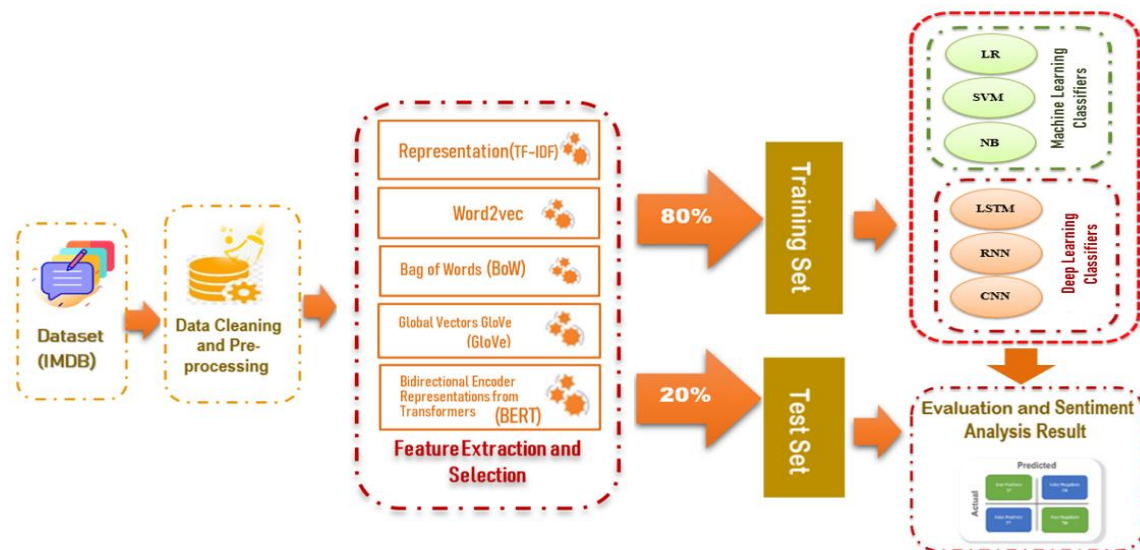


Figure 1. Diagram of the proposed model

In our article, we carried out a sequence of tests to assess the efficiency of different feature extraction techniques and learning models. We employed three frequently used machine learning models: SVM, LR, and NB, as well as three deep learning structures: CNN, LSTM, and RNN. These models were paired with five unique vectorization techniques: TF-IDF, BoW, Word2Vec, GloVe, and BERT. Textual data, sourced from English-language film reviews in the IMDB dataset, were preprocessed and cleaned using

Sklearn libraries in Python. We then performed five separate tests for each model, utilizing each feature extraction technique. For each model based on a machine learning algorithm, trained using the Sklearn library, we applied specific hyperparameters. For the SVM-based model, the “LinearSVC” class was used with a loss function of “Square Hinge”, a tolerance for stopping criteria of “1e-4”, and a maximum number of iterations of “1000”. For the LR-based model, the LR class was used with an “LBFGS” solver and a maximum number of iterations for the solvers to converge of “100”. For the NB-based model, the “MultinomialNB” class was used with a smoothing parameter of “1.0”, “fit\_prior” equal to “True”, and “class\_prior” equal to “None”. Additionally, we employed three deep learning architectures-CNN, RNN, and LSTM-with the Keras library. These models, with a shared set of hyperparameters, were trained in ten iterations with the “Adam” optimizer. To prevent overfitting, we implemented a dropout rate of 0.5. After applying the “rectified linear unit (ReLU)” activation function and maximum pooling, a dropout layer was added before data input to a fully connected layer. The output of this layer was passed through the sigmoid activation function to generate a probability score for sentiment classification. Models were trained and tested using the IMDB movie dataset, which was split into separate subsets for training and testing as described in Table 1.

Table 1. IMDB dataset information

Data	Positive	Negative	Total
Total	25,000	25,000	50,000
Training data	20,000	20,000	40,000
Test data	5,000	5,000	10,000

In summary, our research utilized a mix of five feature extraction methods and six classification models to assess their efficiency in classifying sentiment on the IMDB movie dataset. Indeed, our methodological approach is justified by several factors. Firstly, we chose a diverse set of techniques of feature extraction and classification algorithms to explore a broad range of models and identify those that yield the best results in our specific context. In addition, each technique and algorithm was selected based on previous research and their proven effectiveness in similar sentiment analysis tasks. For instance, TF-IDF is commonly used to identify important terms in a document, while deep learning networks like BERT are known for their ability to grasp the subtleties of natural language [11]. Finally, by using the IMDB Movie Reviews dataset, we ensure that we have a standardized and widely used dataset, facilitating the comparison of our results with other studies and reinforcing the validity of our approach.

## 2.1. IMDB dataset

The IMDB movie review dataset, sourced from the Kaggle website, consists of a total of 50,000 movie reviews, with an equal distribution of positive and negative sentiments [12]. On average, these reviews comprise approximately 234.76 words, with a standard deviation of 172.91 words. This dataset includes a diverse vocabulary, featuring 88,585 unique words. It is widely recognized as a valuable resource for gleaning insights about movies and serves as a standard benchmark in sentiment analysis for film critique.

To evaluate the influence of various feature extraction techniques on the model’s performance, we designated 80% of these reviews for training purposes, while setting aside the remaining 20% (equivalent to 10,000 reviews) for assessing the model’s capability to classify movie reviews as positive or negative. For additional details, please refer to Table 1.

## 2.2. Movie review preprocessing

In the field of NLP, text preprocessing is a crucial component, especially in the context of sentiment analysis tasks. Many NLP datasets consist of unstructured or semi-structured data, often filled with irrelevant or redundant information for analysis. Research by Li *et al.* [13] underscores the significant influence of preprocessing on model performance. Therefore, incorporating preprocessing steps to convert textual data into a suitable format is essential. Efficient preprocessing can lead to a reduction in the size of extracted features, typically between 30% and 50%, thereby preserving only vital features pertinent to the target value. This method not only saves computational resources but also boosts model training efficiency and prediction accuracy, as demonstrated by the study conducted by Samir and Labbib in 2018 [14].

The initial step involves cleaning and preprocessing the text data, similar to the procedure applied in the IMDB movie review dataset, by removing spaces, numbers, unwanted tags, URLs, and brackets. This process involves utilizing operations within the natural language Toolkit (NLTK) in Python. Subsequently, the texts are converted to lowercase to discard unnecessary information that does not enhance the model’s

effectiveness. Preprocessing operations for cleaning textual data, such as reviews, typically include punctuation removal, lowercasing, elimination of empty words, removal of URLs, links, and brackets, stemming, and lemmatization. In conclusion, text preprocessing is a vital step necessary for ensuring the precision and effectiveness of sentiment analysis model predictions.

### 2.3. Feature extraction

In the field of NLP, the process of feature extraction is crucial as it transforms text data into a numerical format that can be processed by machine learning classifiers. This step includes vectorization, a process where text data is converted into numerical vectors, with each component of the vector representing a specific aspect of the text. Given that most machine learning algorithms can only process numerical data, this step is of utmost importance.

The extraction of relevant terms from the text is a critical part of this transformation, as these terms become the base elements for the components of the document's vector, acting as input features for the classification models. Feature extraction holds a central position in NLP, with vectorization serving as a primary method for making text data compatible with machine learning algorithms. This procedure allows for effective analysis and understanding of text data, enabling operations such as sentiment analysis and document classification.

#### 2.3.1. Technical TF-IDF

TF-IDF, a statistical technique in NLP, allocates weights to words based on their occurrence in a document and their inverse occurrence across the collection of documents. It creates a weighted matrix for machine learning algorithms, assisting in tasks related to text categorization. The process of TF-IDF involves the multiplication of the TF matrix, which signifies term frequency, with the IDF matrix, which denotes inverse document frequency. The resulting equation assigns weights to words, thereby improving the efficiency of feature extraction. As pointed out by [15], TF-IDF is a frequently used method in text classification tasks, aiding in the detection of key words within a document. This method is recognized for its proficiency and effectiveness in feature extraction.

$$w_{i,j} = tf_{ij} * \log\left(\frac{N}{df_i}\right)$$

In the analysis of text documents, where  $N$  signifies the total number of documents in the collection, the weight of a term can be denoted by  $w_{i,j}$ , with 'i' standing for the term and  $j$  representing the document. The weight of a term is determined by taking into account its term frequency,  $tf_{ij}$ , and its frequency across the entire collection of documents,  $df_i$ . However, it is important to note that not all terms carry the same level of importance, making it essential to choose only the most relevant ones for further examination.

#### 2.3.2. BOW method

The BoW method is a critical tool in text data analysis and NLP, transforming text into a digital format. It counts the frequency of words in a document and represents it as a vector [16]. The implementation of BoW involves key steps such as tokenization, vocabulary creation, counting, and vectorization. Initially, the text is divided into words or tokens, and a vocabulary is established, with each word assigned a unique index. Subsequently, the frequency of each word in the document is calculated. Ultimately, the text data is converted into a numerical vector, where each component corresponds to word counts in the vocabulary. BoW vectors play a significant role in NLP tasks, particularly in document classification. However, they do have limitations as they fail to capture the meanings or context of words, potentially overlooking complex relationships between words and documents [17].

#### 2.3.3. Word2Vec methods

Word2Vec, a significant advancement in NLP, revolutionizes the transformation of textual data into high-dimensional digital vectors. This pioneering technique, pioneered by Google in 2013, delves into the intricate semantic and syntactic relationships among words through the utilization of a neural network architecture, where a hidden layer serves as the repository for word vectors. Employing both the continuous bag of words (CBOW) and skip-gram methodologies for training, Word2Vec operates with flexibility in its hyperparameter settings, with word vector dimensionality ranging from hundreds to thousands. The training process entails the application of stochastic gradient descent or similar optimization algorithms [18]. The resultant word vectors effectively encapsulate word meanings and associations, thereby proving invaluable in various tasks such as text classification and sentiment analysis. However, despite its prowess in handling extensive datasets, Word2Vec's vectors exhibit a limitation in capturing the nuanced context of entire phrases or sentences [19].

### 2.3.4. GloVe method

GloVe, an influential word embedding method, distinguishes itself from Word2Vec through its unique approach to capturing word relationships. While Word2Vec primarily focuses on local context, GloVe integrates global co-occurrence statistics, allowing it to more effectively grasp both synonymy and antonymy [20]. By constructing a word co-occurrence matrix and subsequently compressing it into a lower-dimensional space, GloVe calculates co-occurrence probability ratios among words with similar meanings [21]. In contrast to Word2Vec, GloVe relies on aggregate word statistics for predictions, providing exceptional speed and scalability in handling extensive datasets. Notably, even with smaller datasets, GloVe demonstrates impressive performance, positioning it as a robust option for various sentiment analysis tasks, including text classification, sentiment analysis, emotion analysis, and text translation.

### 2.3.5. The BERT method

BERT, celebrated for its adeptness in encoding textual data, stands as a deep learning architecture crafted specifically for pre-training text representations, addressing the challenge posed by the scarcity of labeled data in NLP endeavors [22]. In contrast to Word2Vec, BERT operates bidirectionally, taking into account word contexts in both forward and backward directions, thereby yielding more precise word-context representations [23]. Enjoying widespread adoption in NLP applications, BERT proves adaptable for fine-tuning across various tasks such as sentiment analysis, question answering, and named entity recognition, bolstering its efficacy through task-specific data training and parameter optimization [24]. Its distinguished reputation stems from its outstanding performance across diverse NLP tasks and its ability to generate high-quality responses in natural language. Additionally, the availability of pre-trained BERT models to the public renders it a favored option among NLP researchers and practitioners.

## 2.4. Classification models or classifiers

Our research aimed to evaluate the influence of various feature extraction methods on model performance, with a specific focus on sentiment analysis. To do this, we utilized a broad array of artificial intelligence techniques. These techniques encompassed both conventional machine learning algorithms and sophisticated deep learning architectures, forming a comprehensive basis for our investigation.

The subsequent section provides an in-depth overview of the machine learning and deep learning models that were employed in our evaluation. These models were meticulously chosen to ensure a wide-ranging and thorough examination. Their application was pivotal in the successful implementation of our research.

### 2.4.1. Support vector machine

The SVM holds significant prominence in the field of machine learning, particularly for the purpose of data classification. It achieves this by establishing a decision boundary or separating hyperplane to effectively partition data into two distinct sets [25]. In scenarios where the data is linearly separable, SVM functions by creating a hyperplane aimed at maximizing the margin between the two sets. This process involves working with a dataset formatted as  $(x_1, y_1), (x_2, y_2), \dots, (x_i, y_i), \dots, (x_n, y_n)$ , where "y" denotes the target variable and "x<sub>i</sub>" represents the independent variable or feature. Mathematically, the representation of the hyperplane function is expressed as  $f(X) = w_0 + w_1x_1 + w_2x_2 + \dots + w_px_p$ . The determination of the decision boundary is based on solving the equation  $0 = w_0 + w_1x_1 + w_2x_2 + \dots + w_px_p$ . The optimal positioning of the hyperplane, aimed at maximizing the margin between the sets, is visually illustrated in the accompanying graph as shown in Figure 2.

The visual depiction portrays the hyperplane as a solid red line, with dashed lines delineating the margin. Support vectors, which are critical points positioned closest to the margin, hold significant importance in accurately determining the precise position of the hyperplane. The versatility of the SVM algorithm enables it to effectively handle both linearly and non-linearly separable data, rendering it suitable for analyzing high-dimensional datasets as well as smaller ones. These characteristics enhance the applicability of SVM across various domains, encompassing text classification, image classification, and bioinformatics.

### 2.4.2. Logistic regression

LR, frequently utilized in supervised classification tasks, demonstrates proficiency in analyzing binary data and estimating probabilities associated with class associations [26]. By leveraging independent variables, it calculates probabilities associated with binary outcomes or categorical dependent variables. Especially adept at handling categorical target classes, logistic regression fits a sigmoid curve or logistic function to the data to predict class membership probabilities. The sigmoid curve of the logistic function is expressed by the:

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

where  $\sigma(X)$  is the output within the range of 0 and 1,  $x$  is the input.

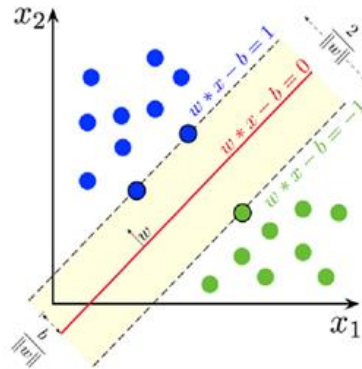


Figure 2. Support vecteur maching, decision boundary

**2.4.3. Probabilistic method (Naive Bayes)**

The NB classifier [27], extensively employed in classification tasks, particularly in the analysis of text data, has achieved success across various applications such as spam filtering and emotion recognition. Built upon Bayes' theorem, it forecasts event probabilities through conditional probabilities. Its "naive" assumption presupposes independence among features within a class, rendering it straightforward and practical to implement, thus well-suited for large datasets [27]. Bayes' theorem is typically articulated mathematically, often depicted by a pair of equations.

$$f(c | x) = f(x | c) f(c) / f(x)$$

$$f(x | c) = f(x_1 | c) * f(x_2 | c) * ... * f(x_n | c) * f(c)$$

In this context, we define various key probabilities essential for statistical inference and machine learning.  $f(c|x)$  represents the posterior probability of class  $c$ , while  $f(c)$  signifies the prior probability of class  $c$ . Additionally,  $f(x|c)$  denotes the probability of the predictor given the class, and  $f(x)$  expresses the prior probability of the predictor.

Prior and posterior probabilities play a vital role in prediction and decision-making. The former embodies the probability of an event before data collection, often grounded in existing knowledge. In contrast, posterior probability undergoes an update following the consideration of available data, portraying a conditional probability informed by dataset information.

**2.4.4. Convolutional neural network**

The CNN is predominantly utilized in image processing, comprising a feature extractor and a classification block [28]. Despite its prevalent usage in image-related tasks, CNNs can be repurposed for text classification, such as sentiment analysis or spam detection [28]. However, to apply CNNs to text-related tasks, textual data must undergo conversion into a numerical matrix, employing techniques like TF-IDF, Word2Vec, BoW, GloVe, or BERT.

Subsequently, text features are fed into the CNN architecture, where filters capture diverse patterns within the text. The resulting feature maps undergo size reduction layers to enhance computational efficiency. These condensed feature maps are then processed through fully connected layers for classification. Each text feature extraction technique is integrated with a CNN, serving as a classifier to categorize data into positive and negative sentiments.

**2.4.5. Recurrent neural network**

The RNN finds extensive application in various domains such as machine translation, speech recognition, text summarization, and music composition [29]. Unlike feedforward networks, RNNs feature a feedback loop mechanism, enabling them to retain memory of past inputs to influence current output [29]. Particularly in NLP, RNNs demonstrate proficiency in tasks like sentiment analysis and text translation by

capturing contextual information embedded within word sequences. Moreover, they are utilized in diverse areas including voice recognition, time series prediction, and image annotation [29]. Despite their effectiveness, RNNs encounter challenges such as the Gradient Vanishing problem, which has led to the development of alternatives like LSTM networks [30].

#### 2.4.6. Long short term memory

LSTM, a variant of RNNs, addresses the issue of gradient vanishing often encountered in conventional RNNs [30]. This challenge emerges when gradients weaken during back-propagation across extended sequences, impeding effective training of deep RNNs. LSTM networks feature a sophisticated architecture comprising memory cells, input gates, output gates, and forget gates, empowering them to preserve long-term memory and regulate the flow of information. Consequently, LSTMs excel in capturing contextual dependencies within sequences, proving particularly valuable for NLP tasks like sentiment analysis and speech recognition. Furthermore, LSTMs have exhibited success in tasks such as time series prediction, image captioning, and various other sequential data processing endeavors [30]. Leveraging these extracted features from input text, LSTM classifiers are trained to classify opinions into positive or negative categories.

### 3. RESULTS AND DISCUSSION

This section presents the results of our study, which evaluated the impact of various feature extraction methods on model performance. We used a range of algorithms on the IMDB dataset for this evaluation. The results provided valuable insights into the effectiveness of different feature extraction methods in improving model performance. Our study particularly highlighted the superiority of the model based on BERT and LSTM in the classification of feelings. This model demonstrated exceptional performance, outperforming other models in this task.

#### 3.1. Results presentation

In this section, we elaborate on the performance of various machine learning and deep learning-based models, along with feature extraction techniques, assessed using metrics such as accuracy, precision, recall, and F1 score as shown in Table 2.

- Models utilizing the SVM algorithm: the SVM-based model demonstrates noteworthy performance, especially when coupled with the BERT technique, achieving superior overall scores with an accuracy of 89.98%, precision of 89.93%, recall of 90.95%, and F1-score of 90.02. The GloVe technique closely follows with an accuracy of 88.97%, precision of 88.90%, recall of 89.97%, and F1-score of 88.25%. Both Word2Vec and TF-IDF also exhibit commendable performance, with F1 scores exceeding 85%.
- Models employing the NB algorithm: the performance of the NB algorithm-based model is equally remarkable, surpassing 86% across all metrics when utilizing the GloVe and BERT extraction techniques. Particularly, the TF-IDF technique shines in terms of accuracy, scoring 88.2%.
- Models based on the LR algorithm: regarding the LR-based model, BERT remains highly effective, with precision, recall, and F1 score nearing 90%. The TF-IDF and Word2Vec techniques also maintain robust performance, achieving F1 scores around 88%.
- CNN-based models: in the realm of CNN-based models, the BERT extraction method emerges as the frontrunner, boasting an impressive accuracy of 93.07% and an F1 score of 88.04%. Overall, other techniques exhibit relatively similar performance, although GloVe stands slightly above with an F1 score of 86.05%.
- Models utilizing the LSTM neural network: the LSTM neural network-based model once again underscores BERT's superiority, attaining an accuracy of 94.00%, precision of 94.14%, and an F1 score of 89.33%. GloVe and Word2Vec techniques also deliver commendable performance, with F1 scores hovering around 86%.
- RNN-based models: in the context of RNN-based models, although BERT maintains dominance with an Accuracy of 92.03%, Precision of 90.13%, Recall of 89.29%, and an F1 score of 88.98%, GloVe and Word2Vec remain competitive with F1 scores near 86%.

This varied performance is contingent on the feature extraction techniques employed, each capturing distinct facets of the text. BERT's supremacy can be attributed to its capacity to capture bidirectional context, while classifier efficacy is also impacted by the complexity of the classification algorithm. Indeed, LSTM networks excel in processing sequential data like text due to their ability to retain and utilize long-term information, enabling them to capture intricate dependencies and contextual relationships in text sequences.



Table 2. The model's performance regarding accuracy, recall, precision, and F1 score

Classification algorithms	Feature extraction techniques	Accuracy	Precision	Recall	F1-Score
SVM	BoW	81.07%	81.17%	83.17%	82.08%
	TF-IDF	83.37%	84.31%	84.34%	84.21%
	Word2Vec	84.13%	84.23%	86.00%	85.4%
	GloVe	88.97%	87.90%	89.97%	88.25%
	<b>BERT</b>	<b>89.98%</b>	<b>89.93%</b>	<b>90.95%</b>	<b>90.02%</b>
NB	BoW	80.96%	82.26%	81.07%	81.06%
	TF-IDF	82.28%	88.2%	81.74%	82.35%
	Word2Vec	83.37%	84.25%	85.07%	84.23%
	GloVe	86.18%	86.10%	86.23%	86.05%
	<b>BERT</b>	<b>86.89%</b>	<b>85.97%</b>	<b>87.01%</b>	<b>88.07%</b>
LR	BoW	86.45%	87.08%	87.77%	87.42%
	TF-IDF	89.62%	89.62%	89.63%	89.62%
	Word2Vec	87.66%	87.67%	87.64%	87.64%
	GloVe	86.3%	86.78%	87.47%	87.27%
	<b>BERT</b>	<b>89.96%</b>	<b>90.23%</b>	<b>90.11%</b>	<b>90.08%</b>
CNN	BoW	84.12%	82.42%	81.37%	81.22%
	TF-IDF	85.26%	82.36%	83.29%	82.26%
	Word2Vec	87.07%	84.13%	85.07%	84.28%
	GloVe	87.89%	85.99%	86.37%	86.05%
	<b>BERT</b>	<b>93.07%</b>	<b>86.42%</b>	<b>87.08%</b>	<b>88.04%</b>
LSTM	BoW	86.14%	82.44%	81.12%	81.04%
	TF-IDF	85.97%	81.99%	83.00%	82.31%
	Word2Vec	89.23%	84.33%	85.4%	84.38%
	GloVe	90.98%	90.98%	86.18%	86.43%
	<b>BERT</b>	<b>94.00%</b>	<b>94.14%</b>	<b>93.27%</b>	<b>89.33%</b>
RNN	BoW	85.07%	82.27%	81.01%	81.21%
	TF-IDF	85.11%	82.11%	83.27%	82.37%
	Word2Vec	88.18%	84.31%	85.41%	84.40%
	GloVe	89.97%	86.24%	86.38%	86.34%
	<b>BERT</b>	<b>92.03%</b>	<b>90.13%</b>	<b>89.29%</b>	<b>88.98%</b>

### 3.2. Comparison with other state-of-the-art research

Previous research on the impact of feature extraction methods in text processing is limited [31]. Table 3 displays the outcomes of various feature extraction techniques and classification models utilized in text classification tasks. Each row in the table corresponds to a specific pairing of a feature extraction method and a classification algorithm, along with the key findings derived from the study.

Our investigation stands out for its thorough exploration of an often overlooked area, delving into various feature extraction techniques from textual data such as BoW, TF-IDF, Word2Vec, GloVe, and BERT. This broad approach allows for a comprehensive evaluation of available methods, unlike previous studies that often focus on a limited set of classification algorithms. We encompass a range of classifiers including SVM, NB, LR, CNN, LSTM, and RNN, facilitating a detailed comparison of performance among different techniques.

Specifically, our study assesses how these techniques influence the performance of machine learning and deep learning-based models, offering valuable insights into their relative advantages concerning accuracy, precision, recall, and F1 score. This analysis provides essential guidance for researchers in selecting appropriate feature extraction methods and classifiers, crucial for guiding future research in machine learning and text analytics. Our findings highlight the effectiveness of various feature extraction techniques, particularly emphasizing the positive impact of BERT on model performance. However, it is important to note that our analysis focused on specific techniques and classifiers with results potentially varying with different methods or datasets. Hence, future research could explore these techniques across diverse textual data and employ a wider range of classification models for a more comprehensive understanding.

In conclusion, our research contributes significant insights into the influence of feature extraction techniques on text classification model performance. BERT, in particular, demonstrates notable enhancements, outperforming other techniques. Additionally, we find that the combination of LSTM and BERT is particularly effective for sentiment classification, emphasizing the importance of selecting suitable techniques and models for accurate classification. Thus, our study makes a substantial contribution to the field of machine learning and text analysis.

Table 3. Comparison with previous point studies on the impact of feature extraction techniques

Study	Feature extraction techniques	Classification models	Dataset used	Principle results
Aljuhani and Saleh (2019) [32]	BoW, TF-IDF, GloVe, Word2Vec	LR, NB, CNN	Mobile phone Reviews	The model using Word2Vec and CNN gave good performance compared to other models.
Ahuja <i>et al.</i> (2019) [33]	BoW, TF-IDF, N-Gram	DT, SVM, KNN, RF, LR, NB	SS-tweet	Models combining TF-IDF and classifiers (DT, SVM, KNN, RF, LR, and NB) improved sentiment analysis performance by 3-4% compared to N-Gram features.
Guda [34]	BOW, N-gram, TF-IDF, Word2Vec	SVM, LR, RF	Comments on the prosperity party	The model using TF-IDF and SVM achieved an accuracy of 0.82.
Mazumder <i>et al.</i> [35]	GloVe, TF-IDF	LR, LSTM	-	<ul style="list-style-type: none"> <li>- The model combining TF-IDF and LR achieved an Accuracy of 87.75%.</li> <li>- The model combining TF-IDF and LSM achieved an Accuracy of 87.89%.</li> </ul>
AlSurayyi <i>et al.</i> [36]	Word2Vec, GloVe	LSTM, Bi-LSTM, CNN	Yelp restaurant reviews	<ul style="list-style-type: none"> <li>- The RNN and Bi-LSTM models performed well for sentiment classification, while SVM achieved an F1 score of 91%.</li> <li>- The CNN and GloVe combination produced the best results of all techniques.</li> <li>- Models based on BoW and BERT achieved impressive F1 scores of 92% and 91.9% respectively.</li> </ul>
Ifitikhar <i>et al.</i> [37]	TF-IDF, BoW, GloVe, Word2Vec	CNN	Amazon	CNN provided the best accuracy of 97% when combined with Word2Vec.
Our study	BoW, TF-IDF, GloVe, Word2Vec and BERT	SVM, NB, LR, CNN, RNN, and LSTM	IMDB movie reviews	<ul style="list-style-type: none"> <li>- Combining BERT with LSTM resulted in the highest performance, achieving an accuracy of 94.00%, precision of 94.14%, recall of 93.27%, and F1 score of 89.33%.</li> <li>- BERT showcased exceptional classification accuracy, surpassing other feature extraction methods across all models.</li> <li>- Integrating BERT with deep learning architectures like LSTM, CNN, and RNN notably enhanced model accuracy.</li> </ul>

#### 4. CONCLUSION

Through our research, we have concluded that feature extraction techniques play a crucial role in determining model performance. Among the various methods investigated, including BoW, TF-IDF, Word2Vec, GloVe, and BERT, BERT stands out as the most effective for enhancing text classification. Its capacity to comprehend contextual nuances and semantics provides it with a distinct advantage over other techniques, resulting in outstanding performance.

Moreover, our experiments have shown that combining BERT with LSTM, CNN, and RNN further enhances performance, with the BERT-LSTM combination proving to be the most efficient model. It achieved an accuracy rate of 94%, precision of 94.14%, recall of 93.27%, and F1 score of 89.33% in text classification tasks. These results underscore the significant contribution of BERT to model performance, surpassing other feature extraction techniques in text classification. Therefore, we recommend utilizing BERT as a feature extraction method, combined with other models, to achieve optimal performance in NLP tasks.




Our study not only highlights the effectiveness of BERT but also prompts further exploration of its application in various text classification contexts. Additionally, it encourages the investigation of synergies between BERT and different deep learning architectures to advance text understanding and analysis. Ultimately, our research provides valuable insights for enhancing NLP systems, enabling more accurate applications in fields such as information retrieval, sentiment analysis, and automatic translation.

#### REFERENCES




- [1] S. Kumar, V. Koolwal, and K. K. Mohbey, "Sentiment analysis of electronic product tweets using big data framework," *Jordanian Journal of Computers and Information Technology*, vol. 5, no. 1, pp. 43–59, 2019, doi: 10.5455/jcit.71-1546924503.
- [2] V. Ashwin, "Twitter tweet classifier," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 5, no. 1, pp. 41–44, 2016, doi: 10.11591/ijai.v5.i1.pp41-44.
- [3] B. Liu, *Sentiment analysis and opinion mining*. Cham: Springer International Publishing, 2012.
- [4] M. Demircan, A. Seller, F. Abut, and M. F. Akay, "Developing Turkish sentiment analysis models using machine learning and e-commerce data," *International Journal of Cognitive Computing in Engineering*, vol. 2, pp. 202–207, Jun. 2021, doi: 10.1016/j.ijcce.2021.11.003.

- [5] R. M. O. Cruz, W. V. de Sousa, and G. D. C. Cavalcanti, "Selecting and combining complementary feature representations and classifiers for hate speech detection," *Online Social Networks and Media*, vol. 28, p. 100194, 2022, doi: 10.1016/j.osnem.2021.100194.
- [6] R. Haque, N. Islam, M. Tasneem, and A. K. Das, "Multi-class sentiment classification on Bengali social media comments using machine learning," *International Journal of Cognitive Computing in Engineering*, vol. 4, pp. 21–35, 2023, doi: 10.1016/j.ijcce.2023.01.001.
- [7] N. Hassan, W. Gomaa, G. Khoriba, and M. Haggag, "Credibility detection in Twitter using word N-gram analysis and supervised machine learning techniques," *International Journal of Intelligent Engineering and Systems*, vol. 13, no. 1, pp. 291–300, Feb. 2020, doi: 10.22266/ijies2020.0229.27.
- [8] L. Mostafa, "Egyptian student sentiment analysis using Word2vec during the coronavirus (COVID-19) pandemic," *Advances in Intelligent Systems and Computing*, vol. 1261 AISC, pp. 195–203, 2021, doi: 10.1007/978-3-030-58669-0\_18.
- [9] S. Sagnika, B. S. P. Mishra, and S. K. Meher, "Improved method of word embedding for efficient analysis of human sentiments," *Multimedia Tools and Applications*, vol. 79, no. 43–44, pp. 32389–32413, 2020, doi: 10.1007/s11042-020-09632-9.
- [10] R. K. Kaliyar, A. Goswami, and P. Narang, "FakeBERT: Fake news detection in social media with a BERT-based deep learning approach," *Multimedia Tools and Applications*, vol. 80, pp. 11765–11788, doi: 10.1007/s11042-020-10183-2.
- [11] F. Saeed, F. Mohammed, and A. Al-Nahari, *Innovative systems for intelligent health informatics: data science, health informatics, intelligent systems, smart computing*, vol. 72. Cham: Springer International Publishing, 2021.
- [12] T. K. Koc, "IMDB movie review-sentiment analysis," *Kaggle.Com*, 2022. <https://www.kaggle.com/code/tarkkaanko/imdb-movie-review-sentiment-analysis>.
- [13] P. Li, K. Mao, Y. Xu, Q. Li, and J. Zhang, "Bag-of-concepts representation for document classification based on automatic knowledge acquisition from probabilistic knowledge base," *Knowledge-Based Systems*, vol. 193, p. 105436, Apr. 2020, doi: 10.1016/j.knsys.2019.105436.
- [14] A. Samir and Z. Lahbib, "Stemming and lemmatization for information retrieval systems in amazigh language," in *Big Data, Cloud and Applications: Third International Conference, BDCA 2018*, 2018, pp. 222–233, doi: 10.1007/978-3-319-96292-4\_18.
- [15] W. N. Ibrahim Al-Obaydy, H. A. Hashim, Y. AbdulKhalq Najm, and A. A. Jalal, "Document classification using term frequency-inverse document frequency and K-means clustering," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 27, no. 3, pp. 1517–1524, 2022, doi: 10.11591/ijeecs.v27.i3.pp1517-1524.
- [16] R. Egger, "Text representations and word embeddings: vectorizing textual data," in *Tourism on the Verge*, vol. Part F1051, 2022, pp. 335–361.
- [17] M. S. Sayeed, V. Mohan, and K. S. Muthu, "BERT: a review of applications in sentiment analysis," *HighTech and Innovation Journal*, vol. 4, no. 2, pp. 453–462, Jun. 2023, doi: 10.28991/HIJ-2023-04-02-015.
- [18] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, p. 12, 2013.
- [19] M. Bryan, B. James, X. Caiming, and S. Richard, "Learned in translation: contextualized word vectors," *Advances in Neural Information Processing Systems*, vol. 30, pp. 6294–6305, 2017.
- [20] J. Pennington, R. Socher, and C. Manning, "Glove: global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543, doi: 10.3115/v1/D14-1162.
- [21] M. Zhang, V. Palade, Y. Wang, and Z. Ji, "Word representation using refined contexts," *Applied Intelligence*, vol. 52, no. 11, pp. 12347–12368, Sep. 2022, doi: 10.1007/s10489-021-02898-y.
- [22] K. Purwandari, T. W. Cenggoro, J. W. Chanlyn Sigalingging, and B. Pardamean, "Twitter-based classification for integrated source data of weather observations," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 12, no. 1, pp. 271–283, Mar. 2023, doi: 10.11591/ijai.v12.i1.pp271-283.
- [23] K. Imamura and E. Sumita, "Recycling a pre-trained BERT encoder for neural machine translation," in *Proceedings of the 3rd Workshop on Neural Generation and Translation*, 2019, pp. 23–31, doi: 10.18653/v1/D19-5603.
- [24] C. Cortes and V. Vladimir, "Support-vector networks," *Machine Learning*, vol. 297, no. 20, pp. 273–297, 1995.
- [25] D. Muchlinski, D. Siroky, J. He, and M. Kocher, "Comparing random forest with logistic regression for predicting class-imbalanced civil war onset data," *Political Analysis*, vol. 24, no. 1, pp. 87–103, 2016, doi: 10.1093/pan/mpv024.
- [26] P. Supsermpol, V. N. Huynh, S. Thajchayapong, and N. Chiadamrong, "Predicting financial performance for listed companies in Thailand during the transition period: A class-based approach using logistic regression and random forest algorithm," *Journal of Open Innovation: Technology, Market, and Complexity*, vol. 9, no. 3, p. 100130, 2023, doi: 10.1016/j.joitmc.2023.100130.
- [27] A. Maiti, A. Abarida, M. Hanini, and A. Oussous, "An optimal model combining SqueezeNet and machine learning methods for lung disease diagnosis," *Current Medical Imaging Reviews*, vol. 20, 2023, doi: 10.2174/0115734056258742230920062315.
- [28] Y. Luan and S. Lin, "Research on text classification based on CNN and LSTM," in *2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, Mar. 2019, pp. 352–355, doi: 10.1109/ICAICA.2019.8873454.
- [29] X. Li and X. Wu, "Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2015, pp. 4520–4524, doi: 10.1109/ICASSP.2015.7178826.
- [30] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, Mar. 2020, doi: 10.1016/j.physd.2019.132306.
- [31] A. Maiti, A. Abarida, and M. Hanini, "A new hybrid artificial intelligence model for diseases identification," *Lecture Notes in Networks and Systems*, vol. 629 LNNS, pp. 825–836, 2023, doi: 10.1007/978-3-031-26852-6\_76.
- [32] S. A. Aljuhani and N. Saleh, "A comparison of sentiment analysis methods on Amazon reviews of mobile phones," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 6, 2019, doi: 10.14569/IJACSA.2019.0100678.
- [33] R. Ahuja, A. Chug, S. Kohli, S. Gupta, and P. Ahuja, "The impact of features extraction on the sentiment analysis," *Procedia Computer Science*, vol. 152, pp. 341–348, 2019, doi: 10.1016/j.procs.2019.05.008.
- [34] S. T. Guda, "Political stance detection on amharic text using machine learning," St. Mary's University, 2023.
- [35] T. Mazumder, S. Das, M. Hasibur Rahman, T. Helaly, and T. Sarkar Pias, "Performance evaluation of different word embedding techniques across machine learning and deep learning models," in *2022 25th International Conference on Computer and Information Technology (ICCIT)*, Dec. 2022, pp. 932–937, doi: 10.1109/ICCIT57492.2022.10055572.
- [36] W. I. AlSurayyi, N. S. Alghamdi, and A. Abraham, "Deep learning with word embedding modeling for a sentiment analysis of online reviews," *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 11, pp. 227–241, 2019.
- [37] S. Iftikhar, B. Alluhaybi, M. Suliman, A. Saeed, and K. Fatima, "Amazon products reviews classification based on machine learning, deep learning methods and BERT," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 21, no. 5, pp. 1084–1101, Oct. 2023, doi: 10.12928/telkonnika.v21i5.24046.




**BIOGRAPHIES OF AUTHORS**

**Abdallah Maiti**    a Moroccan statistical engineer, is a graduate of the National Institute of Statistics and Applied Economics in Rabat. He is currently a doctoral student at the FST in Settat, specializing in artificial intelligence. During his academic and research career, he has contributed to this field by publishing a number of articles in indexed journals, covering machine learning, and computer vision. He can be contacted at email : maiabdel@gmail.com.



**Abdallah Abarda**    a professor of statistic and data analysis in Hassan First University of Settat, with over 31 publications in indexed journals, his research has significantly contributed to the field. Additionally, he's known for organizing international workshops on statistical methods and AI, showcasing his leadership in academia and research. As a member of various conference committees, his expertise extends to shaping the discourse on these subjects. He can be contacted at email: abdallah.abarda@uhp.ac.ma.



**Mohamed Hanini**    a professor at FST Settat Mathematics and Computer Science Department, obtained his Ph.D. in 2013, making significant contributions to the scientific community through approximately 35 indexed articles covering diverse topics. His active participation in international conferences, serving on technical and organizing committees, highlights his leadership in academia. Additionally, he offers his expertise as a reviewer for international journals, further enhancing his scholarly impact. He can be contacted at email: haninimohamed@gmail.com.