Turkish Journal of Engineering
https://dergipark.org.tr/en/pub/tuje
e-ISSN 2587-1366

Analysis of feature extraction techniques for sentiment analysis of tweets

Satyendra Singh*[1], Krishan Kumar [1], Brajesh Kumar [2]

[1]*Gurukula Kangri (Deemed to be University), Department of Computer Science, Faculty of Science, Uttarakhand India, 19523009@gkv.ac.in; krishan.kumar@gkv.ac.in*
[2]*IET Mahatma Jyotiba Phule Rohilkhand University, Department of CSIT, Uttar Pradesh India, bkumar@mjpru.ac.in*

Cite this study: Singh, S., & Kumar, K., & Kumar, B. (2024). Analysis of feature extraction techniques for sentiment analysis of tweets. Turkish Journal of Engineering, 8 (4), 741-753.

https://doi.org/10.31127/tuje.1477502

Keywords
Sentiment Analysis
Twitter
BOW
TF-IDF
Word2Vec
SVM

Research/Review Article

Received: 02.05.2024
Revised: 05.06.2024
Accepted: 06.06.2024
Published: 31.10.2024

Abstract

Over the past few years, sentiment analysis has moved from social networking services like LinkedIn, Facebook, YouTube, Twitter, and online product-based reviews to determine public opinion or emotion using social media textual contents. The methodology includes data selection, text pre-processing, feature extraction, classification model, and result analysis. Text pre-processing is an important stage in structuring data for improved performance of our methodology. The feature extraction technique (FET) is a crucial step in sentiment analysis as it is difficult to obtain effective and useful information from highly unstructured social media data. A number of feature extraction techniques are available to extract useful features. In this work, popular feature extraction techniques including bag of words (BOW), term frequency and inverse document frequency (TF-IDF), and Word2vec are compared and analyzed for the sentiment analysis of social media contents. A method is proposed for processing text data from social media networks for sentiment analysis that uses support vector machine as a classifier. The experiments are carried on three datasets of different context namely US Airline, Movie Review, and News from Twitter. The results show that TF-IDF consistently outperformed other techniques with best accuracy of 82.33%, 92.31%, and 99.10% for Airline, Movie Review, and News datasets respectively. It is also found that the proposed method performed better than some existing methods.

1. Introduction

Natural Language Processing (NLP) is a computational linguistics technique that helps automating the analysis of human language. Sentiment analysis or is an influential application of NLP leading to automatic decision-making based on people's opinions, feelings, attitudes, and perceptions, etc. using textual messages or speech [1]. It classifies the human views as positive, negative, neutral, etc., which are categories of human sentiments. It has applications in various areas including social media monitoring, healthcare systems, politicians, business intelligence, sports, etc. In recent years, social media have become enormously popular to share information, thoughts, knowledge, and opinion, etc. Currently, a lot of users consider online reviews to make various kind of decision on shopping, travelling, hotel booking, movies, healthcare and many other services or activities. Social media platforms such as Facebook, Instagram, LinkedIn, ResearchGate, Twitter, and YouTube, etc. have become important part of modern life, where people share their opinion, ideas, and emotions. Twitter is a widely popular social media platform used by people to interact and share information. According to a report, there are about 400 million Twitter users at present, sending more than 500 million tweets every day [2]. Therefore, Twitter is an effective source of information suitable for sentiment analysis. However, Twitter text data are highly unstructured. Normally, tweets are written in unofficial languages and users often use abbreviations, emojis, and symbols in their tweets [3]. Therefore, extraction of useful information from tweets is a challenging task leading to the need of effective feature extraction techniques. Feature extraction techniques transform text data into an appropriate format and extract useful information discarding the unnecessary contents in the data. Over the years a number of feature extraction techniques have been developed for sentiment analysis of text data. The major techniques include bag-of-words (BOW), term frequency and inverse document frequency (TF-IDF), doc2vec, fastText, and word2vec, etc.

The contributions of research work are as follows:

- To analyze the human opinions and experiences of online review systems for social media information in various domains.
- Extract the sentiments for different classes from the text document in online social media data of natural language.
- A novel scheme based on different pre-processing techniques and feature extraction techniques (text-to-numeric vector representation) is used before using model training for sentiment analysis.
- The experimental results working on three distinct benchmark datasets are presented, showing that our proposed methodology significantly outperforms gains in all taken evaluation parameters.

The paper is structured as follows: section 2 includes a detailed discussion of the proposed methodology for different stages, using three popular benchmark datasets that enhanced the performance for sentiment analysis. In section 3, results and discussions are presented based on various feature extraction techniques and ML classifier. The last section 4, contains the conclusion and outlines future research directions.

Chen X et al. [4] used order preserving sub-matrix (OPSM) and word vector to improve TF-IDF for sentiment analysis of Chinese reviews. OPSM helps to reduce the sparsity. They also proposed a frequent, pseudo-consecutive phrase feature with high discriminative ability (FPCD) to limit the frequent phrase patterns. This approach helped to achieve great results on short text classification. Sohrabi and Hemmatian [5] applied Word2vec technique for processing Twitter data and obtained better results than traditional TF-IDF. Rustam F et al. [6] investigated TF, TF-IDF, and Word2vec feature extraction techniques on the accuracy of tweet classification using ensemble classifier. The results demonstrated that TF-IDF achieved higher accuracy. Li J et al. [7] proposed a model that uses Word2vec for handling semantic gaps and implemented weighted TF-IDF for mapping HTTP traffic to detect anomalies. Umer M et al. [8] compared Word2vec and TF-IDF feature extraction techniques for sentiment analysis of tweets and reported better results for TF-IDF.

Zhao H et al. [9] used log term frequency-based modified inverse class frequency (LTF-MICF) for sentiment analysis of online product reviews. The authors reported that LTF-MICF provided better results than various other techniques including TF-IDF, Word2vec, TF-DFS. Gaye B et al. [10] combined TF-IDF and BOW for tweet sentiment classification. They found that union of two techniques outperformed both techniques if used individually. Kamyab M et al. [11] presented a novel feature extraction method based on TF-IDF feature weighting and pretrained Glove word embedding for sentiment analysis and obtained accuracy up to 94.54% on Tweeter data. Raj C et al. [12] observed that TF-IDF gives high accuracies with conventional machine learning techniques, while Glove vectors consistently perform better with neural networks. Subba and Kumari [13] used a combination of three-word embedding techniques including Bi-directional Encoder Representation from Transformers (BERT), Glove, and Word2vec to obtain high accuracy with an ensemble-based classifier on four different datasets. Tabinda Kokab S et al. [14] employed a pretrained BERT model for extracting semantics and contextual features. The obtained features are processed with a neural network for sentiment analysis of social media data.

2. Proposed methodology

This portion describes the steps of the proposed methodology as shown in Figure 1, which comprises the identification and use of text datasets, the generation of pre-processing techniques, and classification tasks using feature extraction techniques (FET). The main aspects of the proposed approach for sentiment analysis on benchmark datasets are as follows:

- Selection of dataset relates to social media text information.
- Normalize a dataset with the help of NLP pre-processing techniques.
- Extract a useful feature for text documents to convert into a numerical vector representation.
- Design a model (training and testing) using ML classifiers.
- Finally, the result from the evaluation.

2.1. Analysis of datasets

The purpose of this study is to collect textual information about sentiment from various social media sources. In this article, we have studied three benchmark datasets described in Table 1.

- Text data set 1 (TDS1): It is a US airline sentiment Twitter dataset. This data set by February 2015, it is categorized into three classes/labels C/L ϵ *{positive-2363, negative-9178, neutral-3099}*. The focus of this dataset is the tweets of users and their classes for sentiment classification.
- Text data set 2 (TDS2): IMDB dataset stands for the Internet Movie Database (Movie Reviews) base, which consists of two classes/labels C/L ϵ *{positive-25000, negative-25000}* for (binary) sentiment.
- Text data set 3 (TDS3): BBC News is a multi-label topic text classification data set for classifying natural language text based on content. The classifying classes/labels C/L ϵ *{sports-511, business-510, politics-417, tech-401, entertainment-386}* for sentiment headlines.

2.2. Pre-processing techniques

Pre-processing techniques are performed on text-based data for data cleaning, data normalizing, etc. [15]. The pre-processing process may vary depending on the input text data, and these techniques are detailed descriptions given in Figure 2 below. By carefully considering data preprocessing parameters before training a model, we ensure that the data is normalised, clean, transformed, and properly formatted, for the effectiveness of our adopted methodology performance.

Table 1. Description of benchmark datasets.

Dataset Domain	#Total Sample	#Training Sample	#Testing Sample	#Attributes	#Classes/Labels	Language
TDS1 (Airline)	14640	11712	2928	15	3	English
TDS2 (Movie)	50000	40000	10000	2	2	English
TDS3 (News)	2225	1780	445	2	5	English

Detailed descriptions of these three datasets are shown above in Table 1.

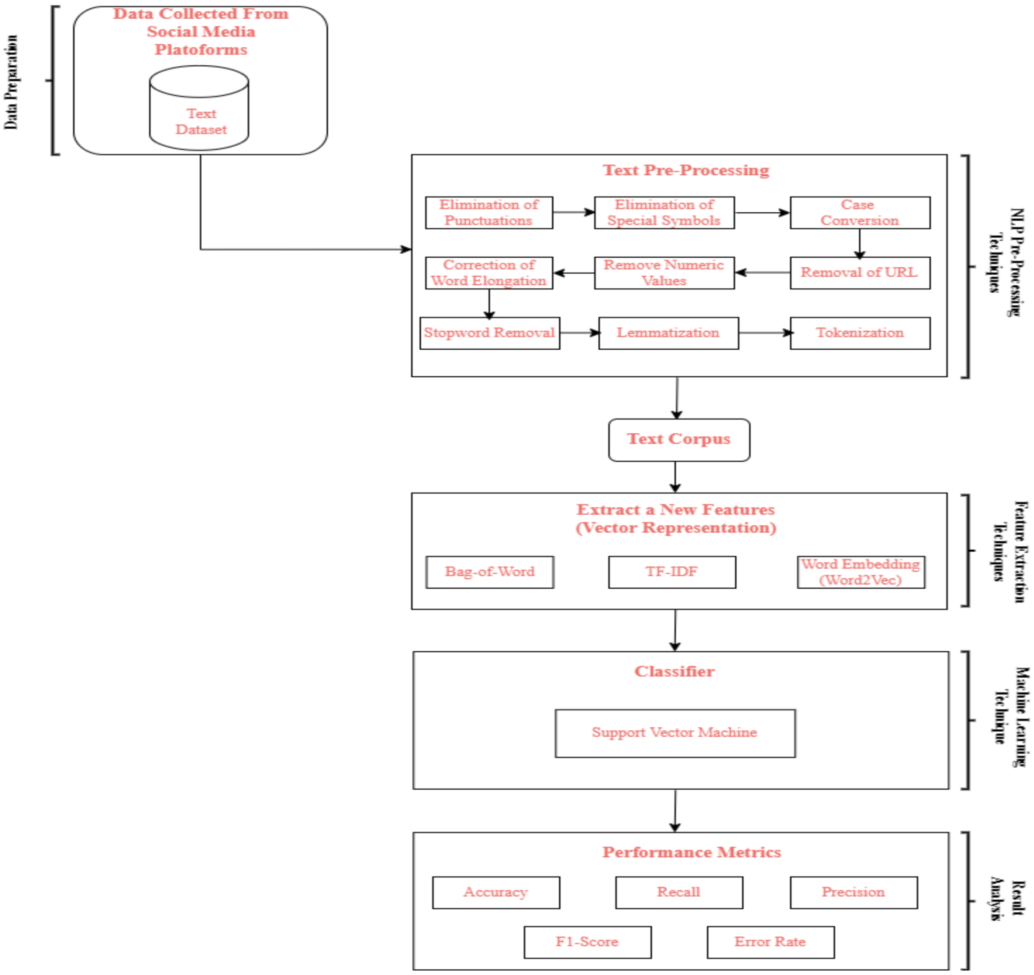


Figure 1. Flow the steps of the proposed working methodology for sentiment analysis.

Removal of Punctuation: All irrelevant symbols/characters are removed in punctuation because machines easily understand text documents.

Case Conversion: All uppercase characters are converted into lowercase in the whole document.

Tokenization: Tokenization is the process of breaking an entire sentence into individual words.

Correction of Word Elongation: The term 'elongation' increases the variety of words where a letter is repeated more than twice.

Stemming: The term stemming is the process of extracting root words from suffix words.

Stop Word Removal: Stop words are those words that are generally not used for analysis, so these words, are pre-eliminated.

Lemmatization: Lemmatization techniques are used in the text pre-processing stage, which is the most common application of NLP models. It recognizes the base form of words and constructs the meaningful of contextual words.

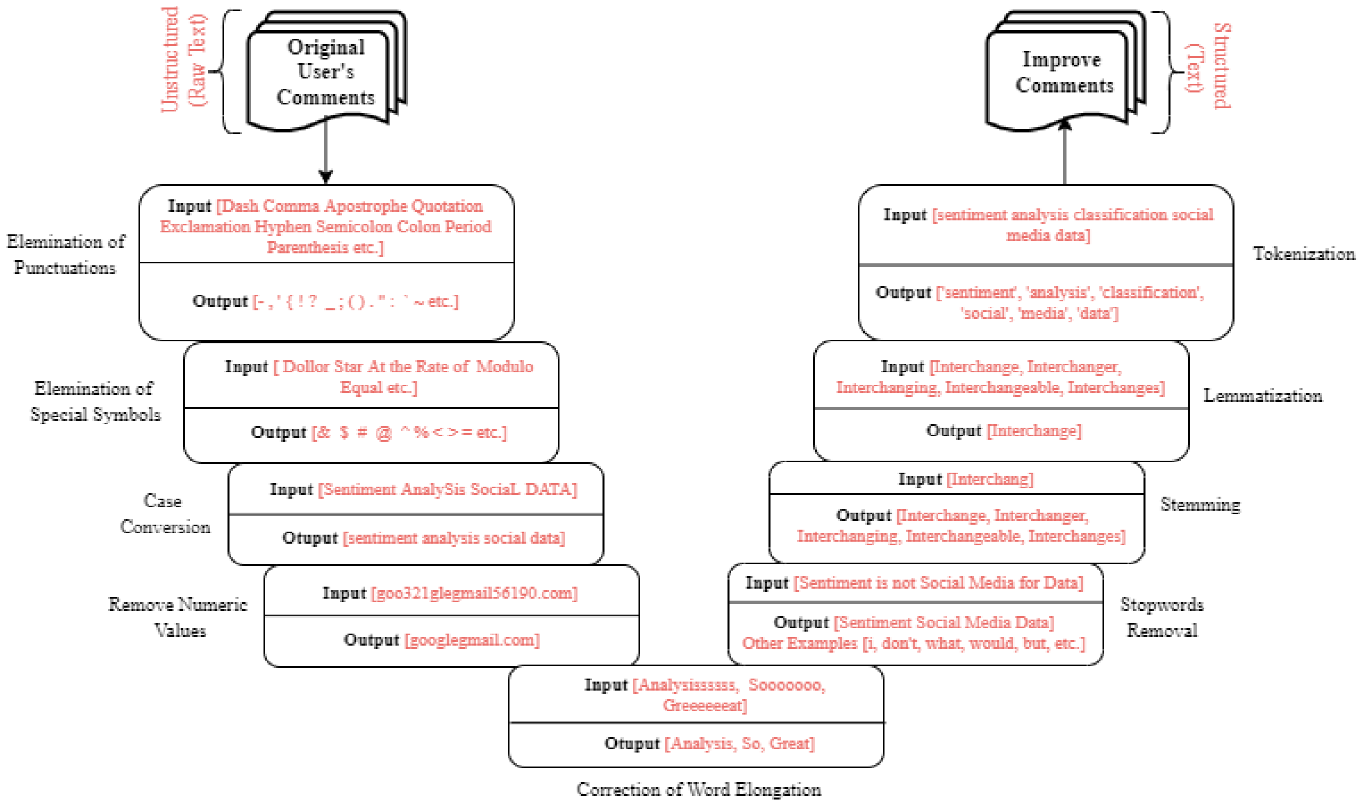


Figure 2. Pre-processing techniques with input/output step-by-step process for improving the user's comments.

2.3. Feature extraction techniques

Feature extraction techniques (FET) are the most crucial step in sentiment analysis; because human opinions are usually identified based on post information, comments, opinion, attitudes, behaviours, etc. Many significant features have been extracted from the selected text dataset. It originated from social media information for sentiment analysis in the word-based features. The extracted features are involved in the four most efficient techniques for sentiment analysis, like BOW, TF-IDF, and word embedding Word2Vec [16]. These techniques extract the beneficial features from feature extraction techniques for text datasets, which are implemented using Python programming.

2.3.1. Bag-of-word

Bag-of-word techniques are the most straightforward way to represent text documents in natural language processing (NLP). The BOW technique is a simple way to extract features from text documents or sentences [17]. The BOW technique represents the text in each document or sentence. It creates a vocabulary that counts how often each document occurs in all unique words [18].

Algorithm 1: Bag-of-Word

Input: Textual contents (dataset)
Output: Calculate the frequency (length) of each document
Notations: words [w_1, w_2....w_n], documents [d_1, d_2....d_n]

Begin:
- BOW model converts from text to vector representation (number)

Step-1 For the given text dataset
Step-2 Initialize build a vocabulary from all unique words [w_1, w_2.... w_n] in the corpus documents (d_1, d_2.... d_n) for each set of word (w)
Step-3 Count the unique occurrence words for each document
Step-4 Calculate the total frequency of each document in a vector representation
End

2.3.2. Term frequency-inverse document frequency

TF-IDF is an acronym that stands for Term Frequency Inverse Document Frequency. TF-IDF is a feature extraction method that extracts weighted features [19]. It assigns weights to each phrase in the text documents to better reform the performance of the trained model [20]. The complete step-by-step process is shown below in algorithm 2.

2.3.3. Word embedding

In NLP, the word embedding technique is converted into a text-to-vector representation. This technique is mapped into word representation in the form of numerical vector representation, which consists of words with the same sense that have similar representations [21].

Algorithm 2: Term Frequency-Inverse Document Frequency

Input: Text-based contents
Output: Compute *tf, idf,* and *tf-idf*

Notations: number (#), term frequency (tf), inverse document frequency (idf), word (w), corpus document (Doc)

Begin:

- This is a mathematical statistic approach that reflects the significance of a word (w) in a textual corpus document (Doc)

Step-1 For the given text-based contents

Step-2 Initialize term frequency-inverse document frequency is split into two stages term frequency *(tf)* and inverse document frequency *(idf)*

Step-3 Compute the value of *tf*

$$tf\,(\mathrm{tm},\mathrm{de}) = \frac{n_{tm,\mathrm{de}}}{\left|\{n_{tm`,\mathrm{de}} : tm` \in de\}\right|}$$

where,

n_{tm} = # of the term (tm);

de = Appears in a document (de);

$n_{tm'}$ = Presence the # of any term (tm');

$\left|\{n_{tm`,\mathrm{de}} : tm` \in de\}\right|$ = Total # of terms in a documents

Step-4 Compute the value of *idf*

$$idf\,(\mathrm{tm},\mathrm{Doc}) = \left[\log \frac{|Doc|}{\left|\{de` \in Doc : tm \in de`\}\right|}\right]$$

where,

Doc = Corpus document;

$|Doc|$ = Total # of documents given in corpus for a term (tm);

$\left|\{de` \in Doc : tm \in de`\}\right|$ = Give the total # of the corpus documents (Doc) in a term (tm)

Step-5 Compute the value of the TF-IDF weight matrix

$$tf-idf\,(\mathrm{tm},\mathrm{de},\mathrm{Doc}) = tf\,(\mathrm{tm},\mathrm{de}) \times idf\,(\mathrm{tm},\mathrm{Doc})$$

End

The idea of word embeddings is helpful for vector representations for text categorization, which can be integrated with DL architectures as well as ML algorithm. The word2vec method is a version of the word embedding technique. The method is based on neural language models, it was developed by Google research team Tomas Mikolov et al. in 2013 [22]. The word2vec technique simple and efficient way to create vector-based representations of words from unlabelled text documents [23]. The model learns from word embeddings using shallow neural network concepts. It is used in a three-layer neural network based on input, hidden, and output layers. These include both surrounding words (context words) and targeted words. The main goal of this model is to transform the high-dimensional feature (HDF) space of words into low-dimensional feature (LDF) vectors while maintaining text similarity out of the corpus.

Algorithm 3: Word2Vec

Input: Textual corpus
Output: Weight matrix (numeric vector representation)

Notations: Left Context Window (LCW), Right Context Window (RCW), High Dimensional Features (HDF), Low Dimensional Features (LDF)

Begin:

- Word2Vec model is a concept of Word Embedding technique based on a text corpus

- This technique converts the HDF space of words into LDF vectors by conserving the similarities of context in the corpus document

Step-1 For the given text corpus

Step-2 Build a vocabulary using the genism python library

Step-3 Build a Word2Vec

- Tuning the parameters
  - (a) Minimum word count (min_count)
  - (b) Vocabulary size ()
  - (c) Window size ()
  - (d) Context word surrounding
    - Left context window (LCW)
    - Right context window (RCW)
  - (e) Hidden layer

Step-4 Generate a frequency of co-occurrence words

Step-5 Computed the output weight matrix (numeric vector representation)

End

The word2vec approach refers to two main architectures, the Continuous bag of words (CBOW) and Skip-gram (SG) models [24]. Predict the target word (center word) from the given context word (surround word) in the CBOW model [25]. The SG model just reverses the CBOW, which predicts the context words (surrounding words) given the target word (the center word). In CBOW, the maximum probability is the word co-occurrence distance based on (d), but the SG model is (-d, +d) comes from the target word. Figure 3. shows the frameworks of CBOW and SK models [22]. The Word2vec model sets values for the hyper-parameters, such as window size, vocabulary size, minimum count frequency, negative, and workers in the large text corpus [26]. These hyper-parameters are used in the Python Gensim library. Many researchers use word2vec techniques to better reform their performance.

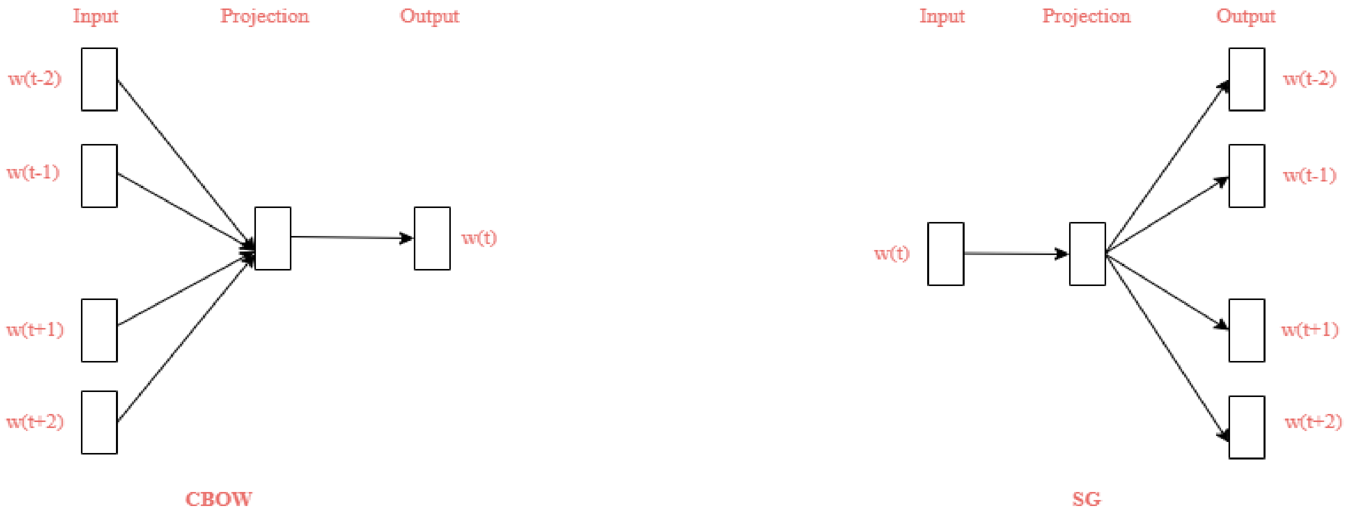


Figure 3. CBOW and SG frameworks.

2.3.4. Classifier

In this section, we discuss machine-learning technique for sentiment analysis. Here, ML technique is analyzed to detect the sentiment from text review data. In general, building a machine learning technique typically involves two phases of data preparation: a training phase and a testing phase. ML technique learned from the training phase. The authors appraise the trained ML technique using testing data. An assessment of testing data is to ensure that we can trust the trained model to predict upcoming inconspicuous data. It provides a brief overview of the most commonly used machine learning technique and their perspective scope of applications.

Support Vector Machine (SVM): SVM is the most common machine learning (ML) algorithm that can be used for regression or classification tasks [27]. It is a statistical-based method that can be divided into two distinct classes in the hyperplane [28]. SVM are linearly separable and plot data points into N-dimensional space. In cases where the data is not linearly separable, the SVM classifier can use kernel functions to transform the input space into a higher-dimensional space where linear separation becomes possible. Many researchers enforce the use of SVM in various applications such as face detection, speech recognition, pattern recognition, text classification, etc.

2.3.5. Performance metrics

The performance of our methodology for measuring the sentiment prediction of binary and multi-class systems is evaluated in this study [29]. We have employed five well-known evaluation metrics, such as precision, error rate, recall, F1-score, and accuracy [30]. Generally, the standardized binary class problem is classified into four classes, which reflect the actual values (horizontal manner) and model prediction (vertical manner), see the confusion metrics in Figure 4.

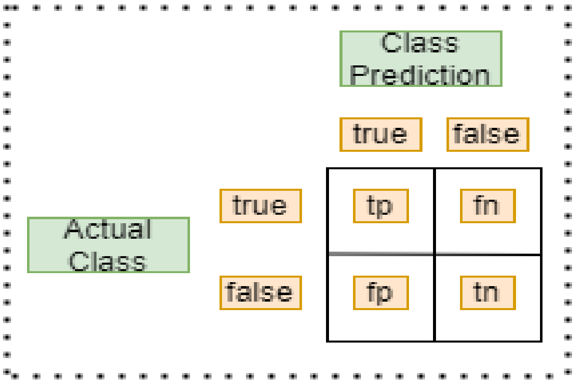


Figure 4. Confusion metrics.

Accuracy (Ac): Accuracy is the proportion of correctly classified samples over all cases of samples, it computes as:

$$Ac = \frac{tp + tn}{tp + tn + fp + fn}$$

Here, *tp* is the true positive rate prediction for the correctly true class, *fp* is the false positive rate prediction for the incorrectly false class similarly, *tn* is the true negative rate prediction for the correctly true class, and *fn* is the false negative rate prediction for the incorrectly false class, respectively.

Precision (Pr): Precision is the proportion of the correct true class among all positive samples, it is calculated as:

$$\Pr = \frac{tp}{tp + fp}$$

Recall (Re): A recall is calculated by dividing a sample of the true positive class over the sum of the true positive class and false negative class samples.

$$\mathrm{Re} = \frac{tp}{tp + fn}$$

F1-Score (F1-S): The F1 score is determined by the harmonic mean of precision and recall.

$$F1 - S = 2 \times \frac{\Pr \times \mathrm{Re}}{\Pr + \mathrm{Re}}$$

Error Rate (Er): The error rate is the ratio of incorrectly classified samples over all cases of samples, it is calculated as:

$$Er = \frac{fp + fn}{tp + tn + fp + fn}$$

or

$$Er = 100 - Ac$$

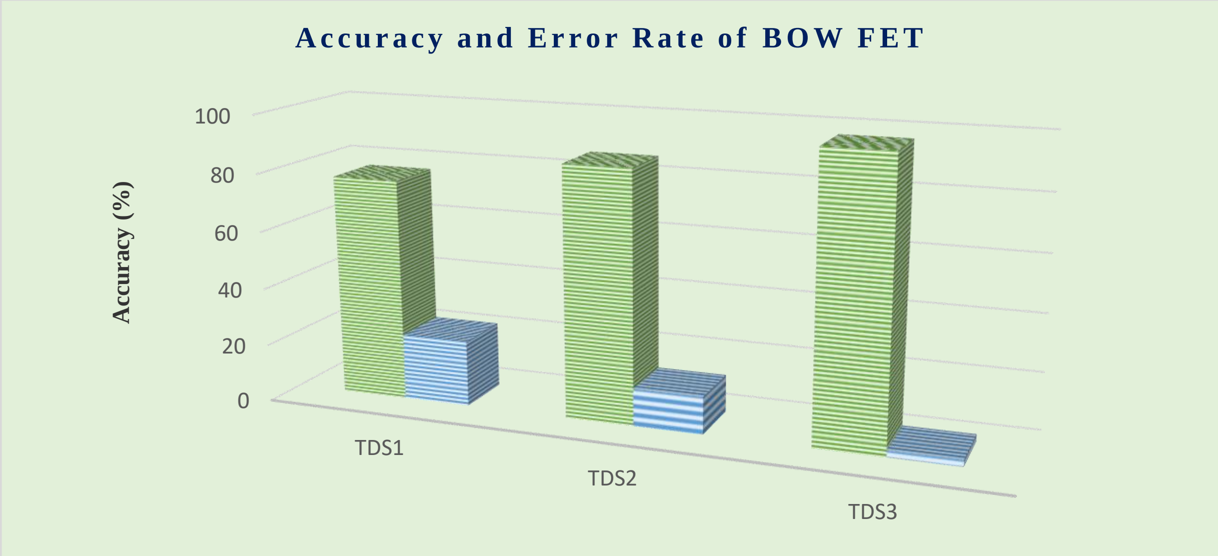
3. Result and discussion

In the present section, we will use three different domain benchmark sentiment analysis datasets to evaluate the effectiveness of our proposed strategy. Moreover, we conducted an experimental assessment to calibrate two facets involved in social media content for sentiment analysis. In the first facet, our purpose is to analyze the impact of the FET model, considering benchmark datasets. In the second facet, we appraise the efficacy of the SVM classification algorithm for sentiment analysis. As can be seen in, Table 2 presents the sentiment analysis results for the SVM classifier with three distinct domain datasets respectively. In addition, these values are computed corresponding to sentiment tweets to maximize accuracy and minimize error for better performance.

Table 2. Performance of the TDS1, TDS2, and TDS3 datasets for SVM classifier

Benchmark Datasets	BOW		TF-IDF		Word2Vec	
	Accuracy (%)	Error Rate (%)	Accuracy (%)	Error Rate (%)	Accuracy (%)	Error Rate (%)
TDS1	77.06	22.94	82.33	17.67	77.77	22.23
TDS2	86.89	13.11	92.31	7.69	87.31	12.69
TDS3	97.53	2.47	99.10	0.90	96.17	3.83

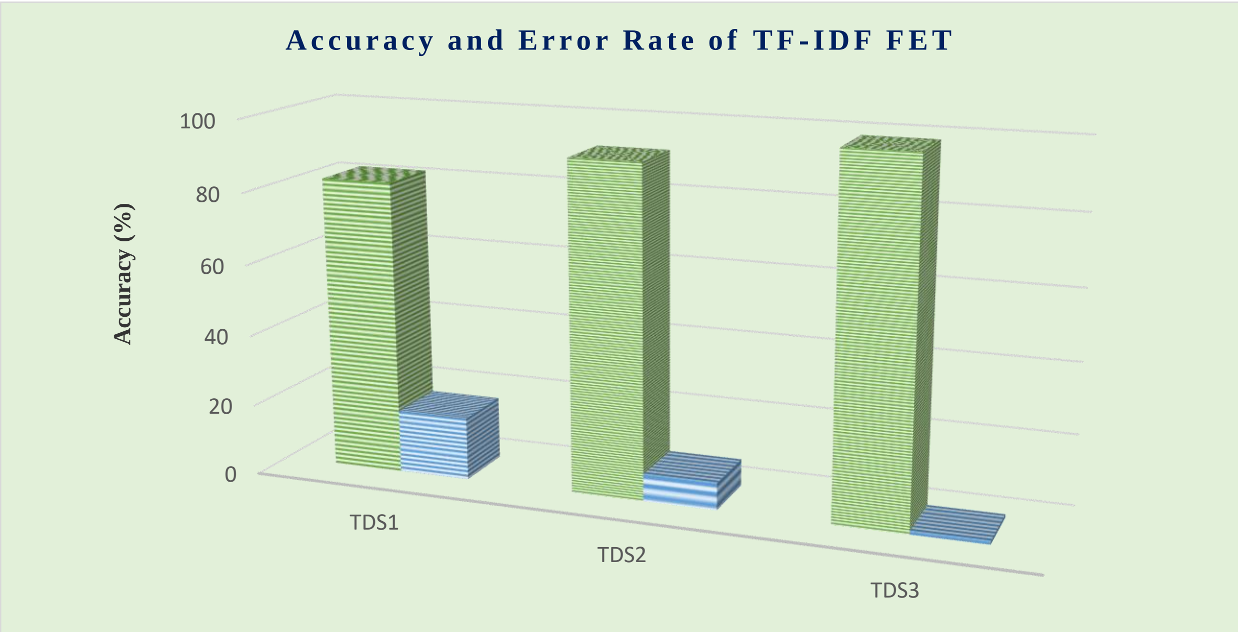
Likewise, using the TF-IDF technique, we find that the TDS3 dataset has 99.10% accuracy and a minimum 0.90% error rate. Which is better performed than BOW and Word2Vec techniques. As shown in Figure 5 we noted that TDS1 gives a minimum accuracy of 77.06% and a maximum error rate of 22.94% as compared to the TDS2 and TDS3 datasets when we use in BOW technique. In other words, we also observe that in Figure 5 TDS3 dataset achieves high accuracy and low error rate which is better performed than the TDS2 dataset.



	TDS1	TDS2	TDS3
Accuracy	77.06	86.89	97.53
Error Rate	22.94	13.11	2.47

Figure 5. Comparision of accuracy for BOW technique.

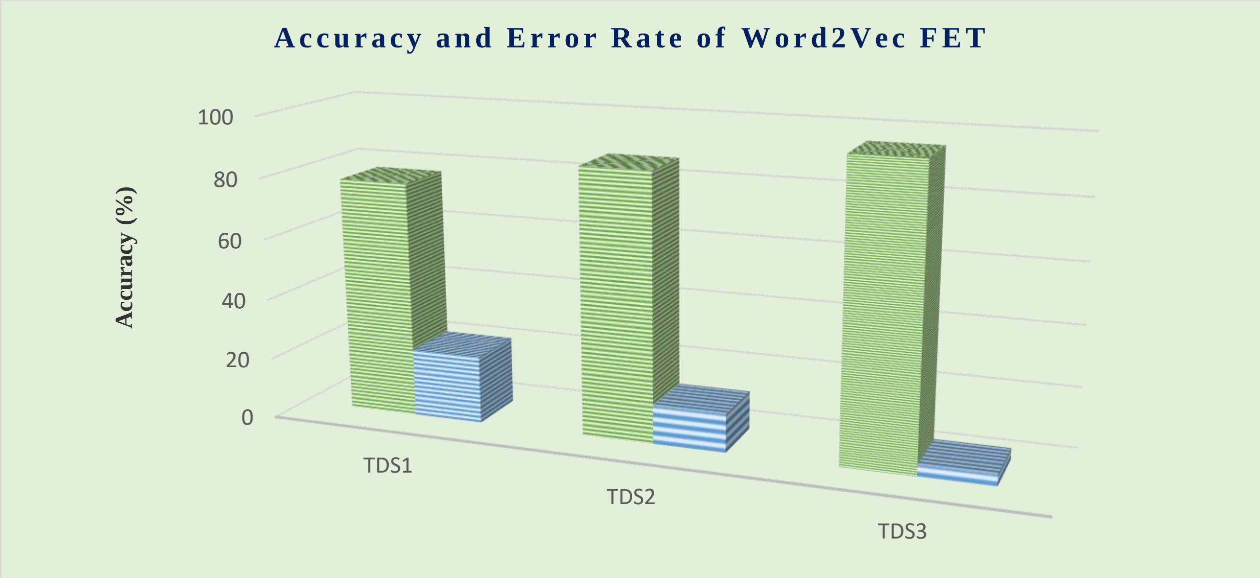
Figure 6 shows that, even when using the TF-IDF technique, the TDS3 dataset achieves the best results as compared to the other two datasets in terms of reduced error rate and improved accuracy. The accuracy and error rates of the TDS2 dataset differ by 10.64% and 6.79%.



	TDS1	TDS2	TDS3
Accuracy	82.33	92.31	99.1
Error Rate	17.67	7.69	0.9

Figure 6. Comparison of accuracy for TF-IDF technique.

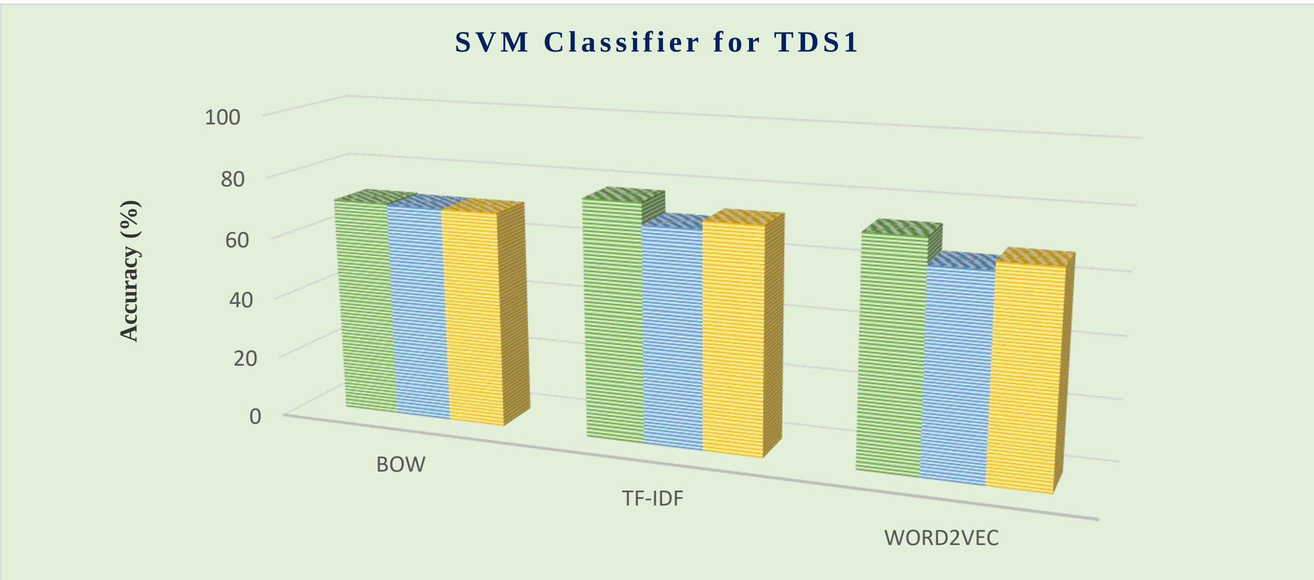
In Figure 7, we reveal the analysis of the accuracy of TDS1, TDS2, and TDS3 datasets. Despite the employ of Word2Vec technology, we found that TDS1 accomplished the lowest accuracy, and TDS3 accomplished the highest accuracy. Likewise, the accuracy using the TDS2 dataset is 87.31%, and the lowest error rate is 12.69%.



	TDS1	TDS2	TDS3
Accuracy	77.77	87.31	96.17
Error Rate	22.23	12.69	3.83

Figure 7. Comparision of accuracy for Word2Vec technique.

Table 3 shows the best classification performance achieved for each FET, with three different domain datasets, and the parameterizations leading to the corresponding results. As one would hope, training feature extraction techniques on their respective corpora improve overall performance in most cases. The performance evaluation metric results of precision, recall, and F1-score are graphically represented as Figures. 8, 9, and 10 the precision value of the TDS3 dataset is 99.27% higher than other datasets when using the TF-IDF technique. Likewise, Figures 8 and 9 show the precision, recall, and F-1 score for each dataset tested individually. The results show that when feature extraction switches from TF-IDF to BOW and Word2Vec, the performance of the TDS1 dataset drops significantly compared to the TDS2 dataset.



	BOW	TF-IDF	WORD2VEC
Pr(%)	71.09	77.68	73.7
Re(%)	70.69	70.66	65.31
F1-S(%)	70.66	73.34	68

Figure 8. Performance evaluation metrics for TDS1.

Table 3. Performance of the TDS1, TDS2, and TDS3 datasets with feature extraction techniques.

Datasets with feature extraction techniques			SVM classifier
TDS1	BOW	Pr (%)	71.09
		Re (%)	70.69
		F1-S (%)	70.66
	TF-IDF	Pr (%)	77.68
		Re (%)	70.66
		F1-S (%)	73.34
	Word2Vec	Pr (%)	73.70
		Re (%)	65.31
		F1-S (%)	68.00
TDS2	BOW	Pr (%)	88.72
		Re (%)	89.88
		F1-S (%)	89.30
	TF-IDF	Pr (%)	90.72
		Re (%)	89.88
		F1-S (%)	89.30
	Word2Vec	Pr (%)	86.93
		Re (%)	88.05
		F1-S (%)	87.48
TDS3	BOW	Pr (%)	97.64
		Re (%)	97.28
		F1-S (%)	97.40
	TF-IDF	Pr (%)	99.27
		Re (%)	99.10
		F1-S (%)	99.00
	Word2Vec	Pr (%)	96.20
		Re (%)	96.28
		F1-S (%)	96.20

The comparison of our proposed work and existing work with three multi-domain datasets is shown in Table 4. This work considers a systematic FET approach that leads to significant improvements compared to those obtained using existing work.

Table 4. Accuracy comparison of the proposed methodology with existing work using text datasets.

Paper's	Approach/Techniques	Dataset	Accuracy (%)
Chen J et al. [31]	BERT	IMDB	75.8
Rustam F et al. [6]	TF-IDF	UST	79.2
Umer M et al. [8]	CNN-LSTM	UST	82.0
Araque O et al. [32]	Ensemble- Based Meta Learning Classifier	IMDB	90.93
Alsayat A [33]	Ensemble Deep Learning	BBC	35.8
Subba B et al. [13]	Word Embedding (Word2Vec, Glove, BERT)	IMDB	92
Our work	FET	TDS1, TDS2, TDS3	82.33, 92.31, 99.10

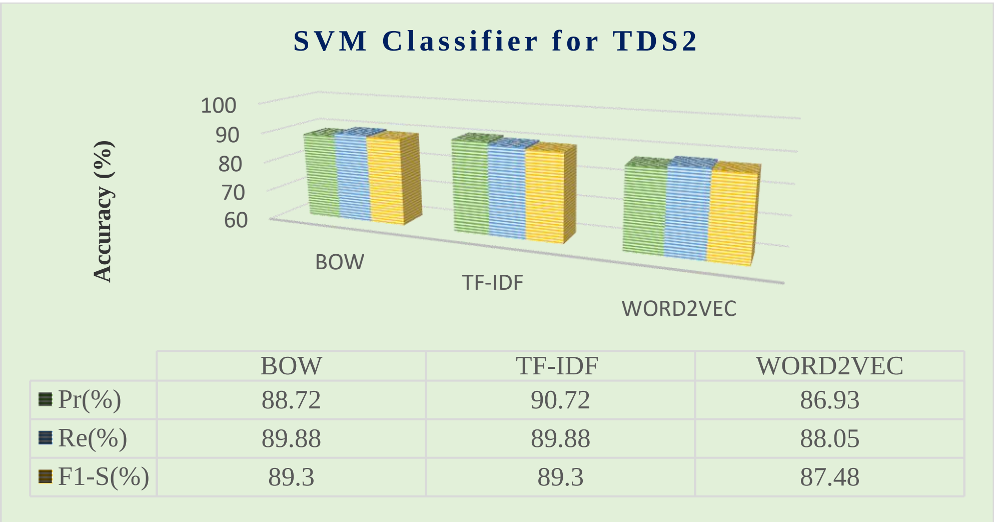
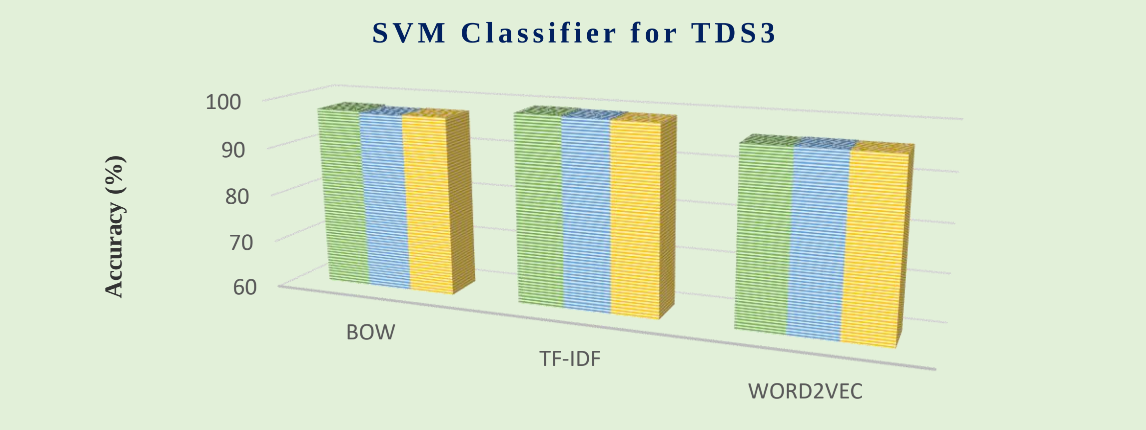


Figure 9. Performance evaluation metrics for TDS2.



	BOW	TF-IDF	WORD2VEC
Pr(%)	97.64	99.27	96.2
Re(%)	97.28	99.1	96.28
F1-S(%)	97.4	99	96.2

Figure 10. Performance evaluation metrics for TDS3.

4. Conclusion and future research

This work intends to propose a FET approach to social media content from various social networking sources that allows the analysis of users' sentiments and attributes. This has been demonstrated to improve the sentiment classification performance of the SVM algorithm on multi-domain datasets. The approach shows that in most cases, it is more dominant and accomplished in terms of classification accuracy with the SVM machine learning algorithm. The methodology achieves prominent reformation in accuracy compared to existing results, and it ensures that our model will indeed be helpful for further exploration. Moreover, the results of three dataset experiments show that by using the SVM algorithm, the classification accuracy of our result outperforms existing works, in multiclass sentiment analysis. As far as future work is concerned, the work done here in this paper can be elaborated to analyze the pre-processing techniques mentioned in the collection of multilingual textual information for a better and more affluent assessment of these techniques; and to provide the pre-trained Bidirectional Encoder Representations from Transformers (BERT)-based language models for multilingual textual content, which could be applicable for various tasks related to opinion mining, sentiment analysis, and utility prediction. It also ensures that the proposed methodology outperforms other methods in the dataset.

Author contributions

Satyendra Singh: Conceptualization, Design Methodology, Implementation work **Krishan Kumar:** Data preparation, Draft preparation, Prove reading. **Brajesh Kumar:** Editing and Modifying research work, Reviewing.

Conflicts of interest

The authors declare no conflicts of interest.

References

1. Lauriola, I., Lavelli, A., & Aiolli, F. (2021). An introduction to deep learning in natural language processing: Models, techniques, and tools. *Neurocomputing, 470*, 443–456. https://doi.org/10.1016/j.neucom.2021.05.103
2. Shayaa, S., Jaafar, N. I., Bahri, S., Sulaiman, A., Seuk Wai, P., Wai Chung, Y., Piprani, A. Z., & Al-Garadi, M. A. (2018). Sentiment analysis of big data: Methods, applications, and open challenges. *IEEE Access, 6*, 37807–37827. https://doi.org/10.1109/access.2018.2851311
3. Naseem, U., Razzak, I., & Eklund, P. W. (2020). A survey of pre-processing techniques to improve short-text quality: A case study on hate speech detection on Twitter. *Multimedia Tools and Applications*. https://doi.org/10.1007/s11042-020-10082-6
4. Chen, X., Xue, Y., Zhao, H., Lu, X., Hu, X., & Ma, Z. (2018). A novel feature extraction methodology for sentiment analysis of product reviews. *Neural Computing and Applications, 31*(10), 6625–6642. https://doi.org/10.1007/s00521-018-3477-2
5. Sohrabi, M. K., & Hemmatian, F. (2019). An efficient preprocessing method for supervised sentiment analysis by converting sentences to numerical vectors: A Twitter case study. *Multimedia Tools and Applications*. https://doi.org/10.1007/s11042-019-7586-4
6. Rustam, F., Ashraf, I., Mehmood, A., Ullah, S., & Choi, G. (2019). Tweets classification on the base of

sentiments for US airline companies. *Entropy, 21*(11), 1078. https://doi.org/10.3390/e21111078
7. Li, J., Zhang, H., & Wei, Z. (2020). The weighted Word2vec paragraph vectors for anomaly detection over HTTP traffic. *IEEE Access, 8*, 141787–141798. https://doi.org/10.1109/ACCESS.2020.301384
8. Umer, M., Ashraf, I., Mehmood, A., Kumari, S., Ullah, S., & Sang Choi, G. (2020). Sentiment analysis of tweets using a unified convolutional neural network-long short-term memory network model. *Computational Intelligence*. https://doi.org/10.1111/coin.12415
9. Zhao, H., Liu, Z., Yao, X., & Yang, Q. (2021). A machine learning-based sentiment analysis of online product reviews with a novel term weighting and feature selection approach. *Information Processing & Management, 58*(5), 102656. https://doi.org/10.1016/j.ipm.2021.102656
10. Gaye, B., Zhang, D., & Wulamu, A. (2021). A tweet sentiment classification approach using a hybrid stacked ensemble technique. *Information, 12*(9), 374. https://doi.org/10.3390/info12090374
11. Kamyab, M., Liu, G., & Adjeisah, M. (2021). Attention-based CNN and Bi-LSTM model based on TF-IDF and GloVe word embedding for sentiment analysis. *Applied Sciences, 11*(23), 11255. https://doi.org/10.3390/app112311255
12. Raj, C., Agarwal, A., Bharathy, G., Narayan, B., & Prasad, M. (2021). Cyberbullying detection: Hybrid models based on machine learning and natural language processing techniques. *Electronics, 10*(22), 2810. https://doi.org/10.3390/electronics10222810
13. Subba, B., & Kumari, S. (2021). A heterogeneous stacking ensemble based sentiment analysis framework using multiple word embeddings. *Computational Intelligence*. https://doi.org/10.1111/coin.12478
14. Tabinda Kokab, S., Asghar, S., & Naz, S. (2022). Transformer-based deep learning models for the sentiment analysis of social media data. *Array*, 100157. https://doi.org/10.1016/j.array.2022.100157
15. Jain, P. K., Pamula, R., & Srivastava, G. (2021). A systematic literature review on machine learning applications for consumer sentiment analysis using online reviews. *Computer Science Review, 41*, 100413. https://doi.org/10.1016/j.cosrev.2021.100413
16. Alamoudi, E. S., & Alghamdi, N. S. (2021). Sentiment classification and aspect-based sentiment analysis on Yelp reviews using deep learning and word embeddings. *Journal of Decision Systems*, 1–23. https://doi.org/10.1080/12460125.2020.1864106
17. Ankit, & Saleena, N. (2018). An ensemble classification system for Twitter sentiment analysis. *Procedia Computer Science, 132*, 937–946. https://doi.org/10.1016/j.procs.2018.05.109
18. Mujahid, M., Lee, E., Rustam, F., Washington, P. B., Ullah, S., Reshi, A. A., & Ashraf, I. (2021). Sentiment analysis and topic modeling on tweets about online education during COVID-19. *Applied Sciences, 11*(18), 8438. https://doi.org/10.3390/app11188438
19. Thakkar, A., & Chaudhari, K. (2020). Predicting stock trend using an integrated term frequency-inverse document frequency-based feature weight matrix with neural networks. *Applied Soft Computing, 96*, 106684. https://doi.org/10.1016/j.asoc.2020.106684
20. Pathak, A. R., Pandey, M., & Rautaray, S. (2021). Topic-level sentiment analysis of social media data using deep learning. *Applied Soft Computing*, 107440. https://doi.org/10.1016/j.asoc.2021.107440
21. Alhumoud, S. O., & Al Wazrah, A. A. (2021). Arabic sentiment analysis using recurrent neural networks: A review. *Artificial Intelligence Review*. https://doi.org/10.1007/s10462-021-09989-9
22. Kalyan, K. S., & Sangeetha, S. (2020). SECNLP: A survey of embeddings in clinical natural language processing. *Journal of Biomedical Informatics, 101*, 103323. https://doi.org/10.1016/j.jbi.2019.103323
23. Alorini, G., Rawat, D. B., & Alorini, D. (2021). LSTM-RNN based sentiment analysis to monitor COVID-19 opinions using social media data. *IEEE Xplore*. https://doi.org/10.1109/ICC42927.2021.9500897
24. Ji, S., Satish, N., Li, S., & Dubey, P. K. (2019). Parallelizing Word2Vec in shared and distributed memory. *IEEE Transactions on Parallel and Distributed Systems, 30*(9), 2090–2100. https://doi.org/10.1109/tpds.2019.2904058
25. Jang, B., Kim, I., & Kim, J. W. (2019). Word2Vec convolutional neural networks for classification of news articles and tweets. *PLOS ONE, 14*(8), e0220976. https://doi.org/10.1371/journal.pone.0220976
26. Borg, A., Boldt, M., Rosander, O., & Ahlstrand, J. (2020). E-mail classification with machine learning and word embeddings for improved customer support. *Neural Computing and Applications*. https://doi.org/10.1007/s00521-020-05058-4
27. Tripathy, A., Agrawal, A., & Rath, S. K. (2016). Classification of sentiment reviews using n-gram machine learning approach. *Expert Systems with Applications, 57*, 117–126. https://doi.org/10.1016/j.eswa.2016.03.028
28. Elhag, M., Idris, N., Mahmud, R., Qazi, A., Ibrahim, Jaafar Zubairu Maitama, Naseem, U., Shah Alam Khan, & Yang, S. (2021). A multi-criteria approach for Arabic dialect sentiment analysis for online reviews: Exploiting optimal machine learning algorithm selection. *Sustainability, 13*(18), 10018. https://doi.org/10.3390/su131810018
29. Birjali, M., Kasri, M., & Beni-Hssane, A. (2021). A comprehensive survey on sentiment analysis: Approaches, challenges and trends. *Knowledge-Based Systems, 226*, 107134. https://doi.org/10.1016/j.knosys.2021.107134
30. Sohrabi, M. K., & Hemmatian, F. (2019). An efficient preprocessing method for supervised sentiment analysis by converting sentences to numerical vectors: A Twitter case study. *Multimedia Tools and*

Applications. https://doi.org/10.1007/s11042-019-7586-4
31. Chen, J., Chen, Y., He, Y., Xu, Y., Zhao, S., & Zhang, Y. (2021). A classified feature representation three-way decision model for sentiment analysis. *Applied Intelligence, 52*(7), 7995–8007. https://doi.org/10.1007/s10489-021-02809-1
32. Araque, O., Corcuera-Platas, I., Sánchez-Rada, J. F., & Iglesias, C. A. (2017). Enhancing deep learning sentiment analysis with ensemble techniques in social applications. *Expert Systems with Applications, 77*, 236–246. https://doi.org/10.1016/j.eswa.2017.02.002
33. Alsayat, A. (2021). Improving sentiment analysis for social media applications using an ensemble deep learning language model. *Arabian Journal for Science and Engineering*. https://doi.org/10.1007/s13369-021-06227-w