HYBRID FEATURE SELECTION FRAMEWORK FOR SENTIMENT ANALYSIS ON LARGE CORPORA

*Kayode S. Adewole[1], Abdullateef O. Balogun[1], Muiz O. Raheem[1], Muhammed K. Jimoh[2], Rasheed G. Jimoh[1], Modinat A. Mabayoje[1], Fatima E. Usman-Hamza[1], Abimbola G. Akintola[1] and Ayisat W. Asaju-Gbolagade[1]

(Received: 5-Jan.-2021, Revised: 22-Feb.-2021, Accepted: 17-Mar.-2021)

ABSTRACT

Sentiment analysis has recently drawn considerable research attention in recent years owing to its applicability in determining users' opinions, sentiments and emotions from large collections of textual data. The goal of sentiment analysis centred on improving users' experience by deploying robust techniques that mine opinions and emotions from large corpora. There are several studies on sentiment analysis and opinion mining from textual information; however, the existence of domain-specific words, such as slang, abbreviations and grammatical mistakes further posed serious challenges to existing sentiment analysis methods. In this paper, we focus on the identification of an effective discriminative subset of features that can aid classification of users' opinions from large corpora. This study proposes a hybrid feature-selection framework that is based on the hybridization of filter- and wrapper-based feature selection methods. Correlation feature selection (CFS) is hybridized with Boruta and Recursive Feature Elimination (RFE) to identify the most discriminative feature subsets for sentiment analysis. Four publicly available datasets for sentiment analysis: Amazon, Yelp, IMDB and Kaggle are considered to evaluate the performance of the proposed hybrid feature selection framework. This study evaluates the performance of three classification algorithms: Support Vector Machine (SVM), Naïve Bayes and Random Forest to ascertain the superiority of the proposed approach. Experimental results across different contexts as depicted by the datasets considered in this study clearly show that CFS combined with Boruta produced promising results, especially when the features selected are passed to Random Forest classifier. Indeed, the proposed hybrid framework provides an effective way of predicting users' opinions and emotions while giving substantial consideration to predictive accuracy. The computing time of the resulting model is shorter as a result of the proposed hybrid feature selection framework.

KEYWORDS

Sentiment analysis, Opinion mining, Hybrid feature selection, Boruta, Recursive feature elimination.

1. INTRODUCTION

Nowadays, the content on the World Wide Web (WWW) has witnessed exponential growth with the advent of e-commerce, blogs, microblogs and social media websites. Availability of large textual data, usually referred to as corpora, has created a massive opportunity to mine users' opinions from such data for business analytics and for decision-making to expand businesses, products and brands and improve customers'/users' experience [1]-[2]. Although large corpora are now available for sentiment analysis to extract opinions and sentiments, processing this text data to extract such information has sparked recent attention from researchers in the last few years. Sentiment analysis is a significant stakeholder in the decision-making process and it enables individuals and groups to make sense of other people's opinions, which can be in textual form [3]. Natural Language Processing (NLP) and text classification are used in Sentiment Analysis (SA), which is a fast-growing area of computing that deals with the challenges of interpreting the text (usually human feelings or opinions) using lexicon-based approach or machine learning approach or hybrid approaches [1], [3]-[5]. Other approaches can be through knowledge-based analysis and statistical analysis or a hybrid of the two [5]. Written text can contain lots of expressions and feelings that may not be easily interpreted by the system. Unique forms of expression -called Emojis- have also been introduced into communication at various user-owned contents at large, be it personal blogs, e-commerce websites or social networks [4], [6].

A lexicon-based method is an approach that utilizes collections of sentiment words and phrases which

1. K. S. Adewole*, A. O. Balogun, M. O. Raheem, R. G. Jimoh, M. A. Mabayoje, F. E. Usman-Hamza, A. G. Akintola and A. W. Asaju-Gbolagade are with Department of Computer Science, University of Ilorin, Ilorin, Nigeria. Email: adewole.ks@unilorin.edu.ng
2. M. K. Jimoh is with Department of Education Technology, University of Ilorin, Ilorin, Nigeria. Email: jmklarularu@gmail.com

are predefined for classifying documents in the corpora to positive, negative or neutral. Lexicon-based approaches can be visualized as methods for clustering documents into clusters of positive, negative or neutral based on sentiment words and phrases already predefined. Recent development and research advancement in the domain of lexicon-based sentiment analysis have been published [7]-[8]. Conversely, machine learning approaches can be broadly classified as supervised and unsupervised learning. This method has been employed to classify or group documents based on some extracted features from documents for sentiment analysis. Supervised machine learning relies on already classified documents as training and a testing dataset which can be used to develop a predictive model for classifying emotions of new unseen documents. On the other hand, unsupervised learning groups documents into some clusters based on the similarities that exist between the documents in the corpora. The literature is vast in the application of machine learning methods for sentiment analysis [1], [6], [9]-[10]. The use of machine learning approaches for sentiment analysis relied majorly on the extraction and selection of highly discriminative features to build effective predictive models.

Feature selection involves the process of selecting a subset of features from the originally extracted features, which are considered as the best features for predicting the class of the documents in the corpora [11]. The purpose of searching for the best subset of features is to reduce the model training time and maintain the accuracy of the predictive model, if not higher than the performance with the original features. Feature selection helps in reducing the dimensionality of the feature subsets by removing irrelevant and noisy features that may hamper the performance of the predictive model for sentiment analysis. In most cases, these irrelevant and noisy features negatively affect the model generalization ability and predictive performance when used for sentiment analysis. Feature selection methods can be classified as filter-, wrapper- and embedded-based techniques [1], [11]-[14]. These three categories of feature selection pertain to the realm of classification algorithms.

At present, the size of user-owned information on the Internet is large and, on the increase, daily. The complex nature of information increase has raised the need for sentiment analysis as a tool used to understand or extract human emotions [6]. Furthermore, researchers are using machine learning algorithms to extract features from the available extensive collection of high-dimensional feature space that identifies and picks relevant features, leaving the noisy and irrelevant features behind [1]. This has made machine learning-based sentiment analysis popular in the field [15]. Some of the popularly used feature selection techniques in the literature are: Recursive feature elimination (RFE), Fast Rank-based method, Relief-F, Gain Ratio, Information Gain, Chi-Square and Boruta [1], [9]. Several supervised algorithms are also available in the literature for classification using the extracted relevant features for sentiment analysis, such as Support Vector Machine (SVM), Decision Trees, Naïve Bayes, Logistic Regression, K-Nearest Neighbour (KNN) and Multilayer Perceptron [4], [6], [9], [16].

There exist several machine learning approaches for extracting sentiment from corpora [1], [15]; however, this paper focuses specifically on corpora from different domains, which are publicly available for sentiment analysis and require a robust technique to extract discriminative features that are of high relevance and can reduce model prediction time. In this paper, a hybrid feature selection framework is proposed through a combination of filter and wrapper feature-selection methods. The resultant discriminative features were subjected to three classification algorithms to ascertain the superiority of the proposed hybrid feature selection framework. More specifically, the main contributions of this paper are briefly highlighted as follows:

- Proposing a new hybrid filter- and wrapper-based feature selection framework for sentiment analysis on large corpora.
- Considering different contexts to ascertain the applicability of the proposed hybrid feature selection framework across various domains, including movie review, public opinion and product review.
- Analyzing the performance of the extracted features by conducting extensive comparative evaluation based on the three machine learning classifiers considered in this study.

The remaining sections of this paper are structured as follows. Section 2 gives related studies on sentiment analysis and machine learning techniques that have been deployed for sentiment analysis. Section 3 explains the proposed hybrid feature selection framework for sentiment analysis, the description of the corpora, the evaluation metrics and the feature selection techniques as well as the classification algorithms used in this study. Section 4 discusses the results of the various experiments conducted and comparatively analyzes the two proposed hybrid feature selection approaches. Finally, Section 5 concludes the paper and gives future research directions.

2. RELATED WORKS

Sentiment analysis and opinion mining are two inter-related concepts that deal with the computational study of people's reaction, attitudes, opinions, sentiments and emotions towards a topic, entity, aspect …etc. expressed in texts [17]-[18]. A vital step of sentiment analysis is selecting an appropriate approach in classifying the opinions. The classification methods of opinion mining can be categorized into two groups; namely: machine, learning approach and lexicon-based approach [19]. The machine learning techniques for opinion mining can be broadly categorized into three aspects, which are; supervised, semi-supervised and unsupervised learning. Several studies have employed unsupervised learning using probabilistic classification, stochastic classification or a combination of both [20]. Probabilistic classification is a famous classification approach in opinion mining; it involves using mathematical expressions to classify the sentiments about a given text. Since the techniques are obtained from probabilistic models, they provide a logical way for classification in a complex domain, such as the field of NLP [20]. Thus, it also has an effective application in opinion mining. Some of the prominent methods in the field of opinion mining belonging to this classification include Naïve Bayes, Bayesian Network and Maximum Entropy. In other situations, which might be the nature of the problem at hand, probabilistic classifiers might be ineffective. Thus, the other option for solving the classification problem is by using stochastic classifiers (also called non-probabilistic classifiers). Some widely used non-probabilistic classifiers in sentiment analysis include Neural Network, Support Vector Machine, K-Nearest Neighbour (KNN), Rule-based methods as well as Decision Tree.

The Bag of Words (BoW) is popularly used to depict sentiment analysis in recent research due to its ability to make word objectivity independent and important as well as giving less importance to subjectivity and text arrangement [15], [21]. In their work, a novel framework was proposed which minimized the size of the feature vectors through semantic clustering and data sparseness for sentiment analysis. Due to the challenges of feature extraction in sentiment analysis, Ansari et al. [9] utilized a hybrid filter and wrapper technique for feature vector selection in order to minimize the vector size and boost the classification accuracy. The researchers used the fast rank-based method on the initial feature set and the result is passed through recursive feature elimination RFE and the evolutionary method of binary particle swarm optimization to obtain the ultimate feature subset.

Hassonah et al. [1] proposed the combination of ReliefF and Multi-Verse Optimizer (MVO) feature extraction algorithms to enhance sentiment analysis. Support Vector Machine (SVM) was used for the classification of the model based on positive, neutral and negative emotions. Comparing the result of the feature extraction techniques and SVM in terms of accuracy against other works showed an improved result with a decrease in the feature numbers by approximately 97% from the original set and datasets yielded an improved result as well [1]. Do et al. [6] carried out a review on deep learning approaches for sentiment analysis with a focus on aspect extraction and sentiment classification. The researchers compared major deep learning methods for aspect level of sentiment analysis. It was concluded that both aspect level sentiment analysis and deep learning need more research work focus as most researchers work on extraction or classification alone, whereas combining both will give a better result. Arulmurugan, et al. [4] proposed a cloud-based technique to integrate emotions like calmness, excitement, stress, confusion and frustration in the construction of an intelligent system for sentiment analysis. The system increased the sentence level sentiment through the use of support vector machine, Naïve Bayes and Neural Network algorithms for classification of specific features of the dataset and a modified K-means clustering method for dataset outliers.

A Japanese large word corpus was developed with five billion words derived from Japanese blogs due to the lack of such in existence anywhere. A two-dimensional model of annotation was used to get information on sentence valence used in sentiment analysis. The evaluation was done on the annotations in more than one way and the large corpus can be used for object ontology and significance of action [22]. Word embedding method was used to improve the accuracy of sentiment analysis for pre-trained words. The technique made use of Part-of-Speech (POS) tagging techniques, lexicon-based approaches, word position algorithm and Word2Vec/GloVe methods. Several deep learning algorithms were used to check the accuracy of the proposed method [3].

Hasan et al. [5] developed a framework for sentiment analysis and classification of hashtag (#) messages representing the opinions of political interest on Twitter. The research compared three sentiment lexicons (W-WSD, SentiWordNet and TextBlob). The polarity and subjectivity were derived using several libraries while Naïve Bayes and support vector machine algorithms were applied to the training set in WEKA to derive the classification model. The best result was obtained from the analysis of tweets with W-WSD. In a research conducted by Arulmurugan et al. [4], Binary Cuckoo Search (based on the characteristics of Cuckoo bird) was used for feature selection of online text content for sentiment analysis and supervised algorithm (support vector machine, decision tree, Naïve Bayes, k-nearest neighbour and multilayer perception) for its classification. The result showed an enhanced accuracy in the sentiment classification due to application on BCS on the dataset for optimized feature selection. Cambria et al. [23] proposed ensemble application of symbolic and subsymbolic Artificial Intelligence (AI) for sentiment analysis. The study integrates both top-down and bottom-up learning using an ensemble of symbolic and subsymbolic AI tools. This was then applied to the problem of polarity identification from text data. A common-sense based Application Programming Interface (API) for concept-level sentiment analysis has been proposed in the literature [24].

Jeyapriya and Selvi [25] employed Naïve Bayes in phrase-level opinion mining in customer product review. The datasets were obtained from Amazon, Eponions and Chet. The model was evaluated based on aspect extraction and sentiment orientation. Also, Tripathy et al. [26] compared the performances of SVM and Naïve Bayes for movie review datasets that were obtained from IMDB. The study was able to show that SVM outperforms Naïve Bayes classifier in predicting the sentiment of a movie review. Alfaro et al. [27] compared the results of SVM and kNN based on content classification and opinion mining on weblog comments. Based on the experiments conducted in the study, it was shown that SVM outperforms kNN in terms of accuracy. Hussain and Cambria [28] employed a semi-supervised learning approach for big social data analytics. The study proposed an affective common-sense reasoning architecture based on random projections and SVM which showed a noteworthy improvement in emotion recognition accuracy as well as in polarity detection. Also, Claypo and Jaiyen [29] utilized an unsupervised machine learning approach for a restaurant review dataset which was obtained from TripAdvisor. The study applied an MRF feature selection technique and KMeans for clustering the reviews into positive and negative. In addition, Al-Agha and Abu-Dahrooj [30] conducted a study by taking data from Twitter to analyze world public sentiment about the Palestinian- Israeli crisis. Their study proposed a multi-level research model that utilizes several variables at the group and individual levels, using statistical methods to carry out a systematic public sentiment. Similarly, Kumar et al. [31] used sale tweets to analyze consumers' thoughts about electronic goods. The researchers discovered that the logistic regression technique has a promising result for all datasets employed. Nahar et al. [32] presented a lexicon-based approach to identify the sentiments of posts and comments on Jordanian telecommunication companies on Facebook. The researchers were able to formulate an Arabic Sentiment Lexicon on which they applied three classification algorithms (SVM, kNN and Naïve Bayes).

From the literature, it is evident that many machine learning models for sentiment analysis have been proposed. However, the utilization of a hybrid machine learning algorithm for feature selection on large corpora is still an open research issue. Thus, the goal of this study is to fill this research gap by proposing a robust multilayer hybrid feature selection framework for sentiment analysis across different domains, such as product reviews, movie reviews, public opinions … and so on.

3. METHODOLOGY

This section details the structure and flow of the techniques used in hybridizing the algorithms for sentiment analysis modeling. First, we begin with the process of data collection and then explain the preparation processes as well as the mechanisms applied to the data gathered to make it accessible and useful for machine learning modeling. Next, we offer a comprehensive overview of the datasets with the number of their attributes, instances and classes. The subsequent subsections explain the selected feature selection algorithms and the classification techniques employed.

3.1 Proposed Framework

Figure 1 shows the proposed hybrid feature selection framework, which comprises different stages to achieve the overall aim of this study. Four publicly available datasets were collected to evaluate the performance of the proposed hybrid framework. The subsequent sections provide detailed explanations of the various stages involved in the proposed hybrid framework.

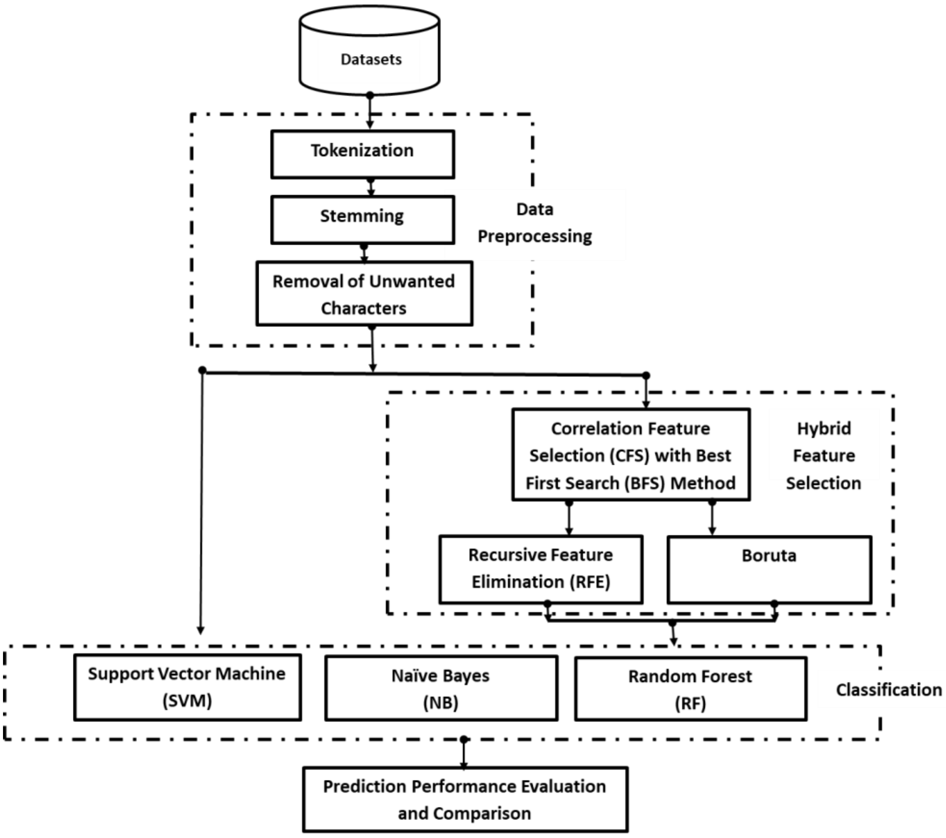

Datasets
Tokenization
Stemming
Removal of Unwanted Characters
Data Preprocessing
Correlation Feature Selection (CFS) with Best First Search (BFS) Method
Hybrid Feature Selection
Recursive Feature Elimination (RFE)
Boruta
Support Vector Machine (SVM)
Naïve Bayes (NB)
Random Forest (RF)
Classification
Prediction Performance Evaluation and Comparison

Figure 1. The proposed hybrid framework for sentiment analysis.

3.2 Data Collection and Description

In evaluating the proposed hybrid feature selection framework for sentiment analysis on textual data, several open-source datasets were utilized. These include Amazon, Yelp, IMDB and Kaggle datasets which are publicly available for research purposes. The brief descriptions of these datasets are as explained in Table 1. The corpora are written in English language.

Table 1. Corpora description.

Dataset	No of instances	No of attributes
Amazon	1000	620
Yelp	1000	691
IMDB	1000	961
Kaggle	13871	1218

The corpora used in this study comprise three datasets of customer reviews (Amazon, Yelp and IMDB datasets) on several products and services and the United States 2016 Presidential debate, which is a Kaggle dataset. The first dataset, Amazon, is an open-source corpus that is publicly available at “https://registry.opendata.aws/amazon-reviews/”. The corpus has 1000 instances of customer reviews on products purchased on Amazon store. Similarly, Yelp is also a corpus consisting of 1000 instances of customers’ reviews on products. This corpus is readily available at https://www.yelp.com/dataset. IMDB is a corpus of customers’ reviews on movies which is also publicly accessible at https://www.kaggle.com/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews. The Kaggle dataset on the other hand is the First US GOP debate which is openly accessible at https://www.kaggle.com/crowdflower/first-gop-debate-twitter-sentiment. This corpus entails tweets about the presidential debate in determining whether the contributor's opinion is positive, negative or neutral. The corpus has 13,871 tweets which were analyzed based on relevancy, the candidate that was mentioned, the subject as well as the sentiment that was given to the tweet.

3.3 Data Pre-processing and Preparation

The corpora obtained undergo a series of pre-processing stages to prepare and transform them into a more consumable form that can be used by the algorithms. These phases include:

i. Tokenization: This is the process of producing several representations of information-enriched texts which can lead to a better classification outcome. It operates by transforming the extracted documents and texts into more practical and machine-consumable forms of texts, such as words, phrases, sentences, …etc. This is the first process of feature extraction where texts are converted into tokens before transforming into vectors.

ii. Stemming: Next, with the "tm_map" feature provided *via* the "tm" package in R, all the derivative words were transformed back to their root form. Stemming is highly valuable as it assists in the recognition of related terms as well as their reduction in data dimensionality.

iii. Removal of unwanted characters: Numbers, unwanted spaces, special characters, …etc. are all eliminated from the word list, as they are unnecessary and meaningless. These do not contribute to the sentiment, as they degrade the performance of the machine learning models.

iv. Feature extraction: Finally, the conversion to the document term frequency matrix was done. Document term frequency is a statistical matrix that displays the frequency of words contained in a record set. In this matrix, each row denotes a document, each column represents one term (word) and each entry value has the number of appearances of that term in that document.

3.4 Hybrid Feature Selection

The major target of attribute selection in sentiment analysis is to discover the best set of features that allows building useful models. The major goal of most applications is to develop a well-performing prediction model. Another, sometimes more important, is to identify those variables that enhance this good prediction; i.e., reducing the large set of measured variables to the ones that contain more information rather than noise [14], [33]. Thus, several feature selection techniques have been proposed using different principles and approaches to report the set of truly relevant variables. In this study, a hybrid feature selection is employed. First is the Correlative Feature Selection (CFS), a filter-based attribute selection technique that is based on Best First Search (BFS) technique. The result of this feature selection process is then passed to either Boruta or RFE which are wrapper-based attribute reduction methods. The final feature subsets are passed to the classification algorithms to evaluate the proposed hybrid feature selection framework.

3.4.1 Correlative Feature Selection

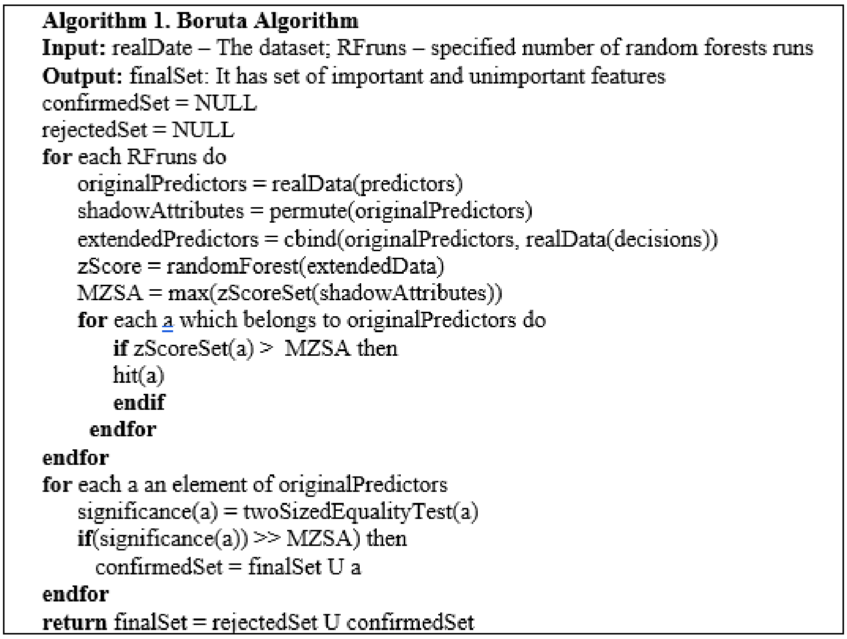
Correlation Feature Selection (CFS) is a typical type of filter feature subset selection methods. Filter feature subset selection evaluates, ranks and selects features based on some properties. CFS generates a feature subset based on the search method that is employed to select features that possess good prediction capacities. The search method traverses the feature space to generate a subset of the features with high predictive potentials. According to [34], CFS considers the existence of better predictive performance when combining features. The performance of CFS varies considerably based on the search method employed. Therefore, we carefully selected one of the best search methods as reported in the literature for the CFS stage, which is based on BFS approach [35]. The best first search strategy works by first emptying a set of attributes, beginning with the complete set of attributes or beginning a quest in any given direction and going backwards (by considering all possible single attribute additions and deletions at a given point).

3.4.2 Boruta Algorithm

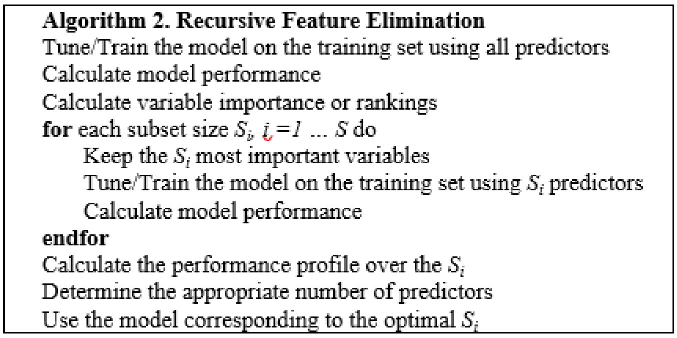
The Boruta algorithm, named after a Slav god of the forest, was created to identify all important attributes in a classification framework [36]. It is a wrapper technique that is built around the Random

Forest classifier, with the main idea of comparing the importance of the real predictor variables (known as real-data/original data) with those of random (also called shadow) variables using statistical testing and some runs of Random Forest. In each run, the set of predictor variables is folded by adding a copy of each variable. The values of those shadow variables are generated by permuting the original values across observations and therefore destroying the relationship with the outcome. A random forest is trained on the extended set of data (called extended data) and the variables' importance values are collected. For each real variable, a statistical test is conducted to compare its importance with the maximum value of all the shadow variables. Variables with significantly larger or smaller importance values are declared as important or unimportant, respectively. All irrelevant features and shadow attributes are eliminated and then the previous steps are repeated until all the features are classified or a pre-specified number of runs have been performed.

The Boruta algorithm is described as follows [36]:



```
Algorithm 1. Boruta Algorithm
Input: realDate – The dataset; RFruns – specified number of random forests runs
Output: finalSet: It has set of important and unimportant features
confirmedSet = NULL
rejectedSet = NULL
for each RFruns do
    originalPredictors = realData(predictors)
    shadowAttributes = permute(originalPredictors)
    extendedPredictors = cbind(originalPredictors, realData(decisions))
    zScore = randomForest(extendedData)
    MZSA = max(zScoreSet(shadowAttributes))
    for each a which belongs to originalPredictors do
        if zScoreSet(a) >  MZSA then
        hit(a)
        endif
     endfor
endfor
for each a an element of originalPredictors
    significance(a) = twoSizedEqualityTest(a)
    if(significance(a)) >> MZSA) then
      confirmedSet = finalSet U a
endfor
return finalSet = rejectedSet U confirmedSet
```



```
Algorithm 2. Recursive Feature Elimination
Tune/Train the model on the training set using all predictors
Calculate model performance
Calculate variable importance or rankings
for each subset size $S_i$, $i$ =$1$ ... $S$ do
    Keep the $S_i$ most important variables
    Tune/Train the model on the training set using $S_i$ predictors
    Calculate model performance
endfor
Calculate the performance profile over the $S_i$
Determine the appropriate number of predictors
Use the model corresponding to the optimal $S_i$
```

3.4.3 Recursive Feature Elimination (RFE)

Recursive Feature Elimination (RFE) is a wrapper-based feature selection technique. It is a brute-force approach to attribute selection that operates by searching for a subset of features. It begins with all attributes in the training dataset and successfully eliminates the weakest features until the desired number of features remains. In RFE, features are classified according to the model's attributes. RFE aims to remove co-linearity and dependencies that may exist in a model by recursively deleting a limited number of features per cycle.

3.5 Classification Algorithms

This study investigates the performance of three machine learning algorithms to ascertain the applicability of the proposed hybrid feature selection framework for sentiment analysis across different domains. The classification algorithms were selected due to their wide range of acceptability for similar classification tasks in the literature [1].

3.5.1 Support Vector Machine

Support Vector Machine (SVM) is a well-known supervised machine learning technique used for classification. It is an efficient machine learning technique based on the principle of structural risk minimization; it is capable of solving the small-sample and nonlinear classification problems. The basic principle of SVM is that it searches for optimal separating hyperplane so that the classification problem becomes linearly separable. Given a set of labelled data where there are two possible label classes, SVM builds a model that maps the data as points in a space so that the two separate classes of labelled data are divided by a clear gap as wide as possible. Thereafter, the model is used in mapping unknown data into the previously mentioned space and predicting the label class of the unknown data based on which side of the gap it is mapped.

3.5.2 Random Forest

Random Forest is a prominent machine learning classification algorithm that has a collection of tree predictors. Each of these predictors is used for classifying an unknown instance. The resulting classification for the unknown instance is selected based on the majority result of the trees' predictions. Random Forest is a class of decision tree algorithms based on an ensemble approach [37]. It creates an ensemble of classifiers by creating several decision trees using a random feature selection and bagging approach at the training stage. This decision tree yields two types of nodes: the leaf node labelled as a class and the interior node associated with an attribute. A different subset of training data is selected with a replacement in training each tree. Entropy is applied to compute the information gain contributed by each feature. Let *D* represent the corpus with the labelled instances and *C* the class such that $C = \{C_1, C_2, C_3, \ldots, C_j\}$, where *j* is the number of classes considered. In this paper, the value of *j* is set to 2 or 3 depending on the specific corpus used, as earlier discussed. Formally, the information needed to identify the class of an instance in the corpus *D* is denoted as *Info(D) = Entropy(P)*, where *P* is the class probability distribution, such that:

$$P = \left\{ \frac{|C_1|}{|D|}, \frac{|C_2|}{|D|}, \frac{|C_3|}{|D|}, \ldots, \frac{|C_j|}{|D|} \right\} \tag{1}$$

By partitioning *D* based on the value of a feature *F* according to subsets $\{D_1, D_2, D_3, \ldots, D_n\}$, *Info(F,D)* with respect to *F* can be computed as:

$$Info(F, D) = \sum_{i=1}^{n} \frac{|D_i|}{|D|} Info(D_i) \tag{2}$$

The corresponding information gain after obtaining the value of *F* is computed as:

$$Gain(F,D) = Info(D) - Info(F,D) \tag{3}$$

Thus, the *GainRatio* is defined as:

$$GainRatio(F, D) = \frac{Gain(F, D)}{SplitInfo(F, D)} \tag{4}$$

where, *SplitInfo(F, D)* shows the information due to the splitting of *D* according to the feature *F*. Random Forest uses the majority voting of all the individual decisions to obtain the final decision of the classifier.

3.5.3 Naïve Bayes

A Naïve Bayes is a supervised probabilistic machine learning classifier that is based on Bayes' theorem with strong independence (naïve) assumption among the features. Naïve Bayesian classification assumes that the variables are independent given the classes. That is, Naïve Bayes assumes that the presence of a specific attribute in a class is unrelated to the presence of any other attributes.

Formally, let *C* be the random variable denoting the class of an instance; Let *X* be a vector of a random variable denoting the observed attribute values. Let *c* be a particular class label and *x* represent a particular observed attribute value. According to the independence assumption, attributes x_1, x_2,…, x_n are all conditionally independent of one another, given C. The value of this assumption is that it simplifies the representation of conditional probability *P(x|c).* Naïve Bayes gives a way of finding the conditional probability *P(x|c)* from *P(c)*, *P(x)* and *P(c|x)*. This relationship is as described in Equation (5) below.

$$P(x \mid c) = \frac{P(c \mid x)P(x)}{P(c)} \tag{5}$$

where, *P(x|c)* is the posterior probability of class *x,* given *c, P(x)* is the prior probability of the class, *P(c|x)* represents the likelihood which is the probability of predictor, given class and *P(c)* represents the prior probability of predictor.

3.6 Evaluation Metrics

The details of the evaluation metrics employed in this study are discussed in this section. The metrics provide globally acceptable techniques to check the performance of the proposed method. In machine learning, model classification performance can be obtained *via* a confusion matrix to ascertain the model ability in classifying the instances under consideration. The confusion matrix, shown in Table 2, is a matrix that gives the classification performance on how well a classifier can separate one class from another. The table presents the confusion matrix general structure for the binary class classification problem. In this table, True Positive (TP) and True Negative (TN) refer to the number of correctly classified positive and negative sentiments, respectively. False Positive (FP) represents the number of negative sentiment documents classified as positive, while False Negative (FN) represents the number of positive sentiment documents classified as negative.

Table 2. Confusion matrix for a binary class problem (positive and negative sentiments).

		Predicted Class	
		Class = Positive sentiment	Class = Negative sentiment
Actual Class	Class = Positive sentiment	TP	FN
	Class = Negative sentiment	FP	TN

The parameters TP, TN, FP and FN, as shown in Table 2, can be used to derive some standard metrics, such as Accuracy, Precision, Recall, F-Measure and Receiver Operating Characteristics (ROCs). Each of these metrics is discussed as follows:

Accuracy: Accuracy is the most intuitive indicator of performance, as it a ratio of appropriately predicted observations to the overall observations. Formally, it is represented as:

$$Accuracy = \frac{Tp + Tn}{Tp + Tn + Fp + Fn} \tag{6}$$

Precision: Precision is a measure that evaluates the correct number of positive predictions. It is calculated as:

$$\Pr ecision = \frac{Tp}{Tp + Fp} \tag{7}$$

Recall: It is otherwise referred to as sensitivity which measures the model’s ability to correctly identify the true positives. Mathematically, it is expressed as:

$$\mathrm{Re}\, call = \frac{Tp}{Tp + Fn} \tag{8}$$

F-Measure: F-measure, otherwise called F-score, is a common evaluation metric for machine learning models. It is described as the harmonic mean of the precision and recall of a model. The relationship is as described in Equation 9.

$$F - Measure = 2 \times \frac{preccision \times recall}{precision + recall} \tag{9}$$

Receiver Operating Characteristics (ROCs): ROC term illustrates how well a classification model performs at all classification levels. A ROC curve is a graph that reveals the relationship between sensitivity and specificity for every individual possible cut-off. ROC curve is a graph in which the x-axis represents 1 – specificity, while the y-axis is the value of sensitivity.

4. EXPERIMENTAL RESULTS AND DISCUSSION

Attribute reduction is a vital phase of machine learning modeling, as it assists greatly in reducing the number of features needed for classification and subsequently reducing the classifier's processing time. This section discusses the results obtained when the hybrid feature selection framework was used on different corpora. It first presents the environment and detail specifications for the various experiments conducted. We then discuss the results obtained for each of the classification algorithms employed in this study without the proposed feature selection techniques. Finally, we present the results of the proposed hybrid feature selection framework based on filter and wrapper feature selection techniques. In this research, three machine learning classifiers (SVM, Naïve Bayes and Random Forest) were considered, with or without the combination of three feature selection optimizers (CFS, Boruta and RFE). The performance of the selected classifiers was evaluated using Accuracy, Precision, Recall, F-measure and ROCs.

4.1 Experimental Setup

Several experiments were conducted in this research based on two different machine learning tools; R and Weka. RStudio version 1.2.5001 was used as the Integrated Development Environment (IDE) for coding the R scripts. R language was used for preprocessing, feature extraction and feature selection including implementation of CFS, Boruta and RFE techniques. Weka was used to implement the selected classification algorithms. Four (4) datasets; namely, Amazon, Yelp, IMDB and Kaggle were considered. The descriptions of the datasets have been discussed in Section 3. The classification experiments were conducted on WEKA version 3.8 running on a 32GB-RAM personal computer with Core i9 2.90GHz processor speed. The computer is running on a 64-bit Windows 10 operating system. For training and testing of the classification algorithms, 10-fold cross-validation has been employed, which allows to obtain models that can be generalized when deployed in real-world for sentiment analysis on large corpora.

4.2 Discussion of Results

Table 3 summarizes the different experiments conducted, which are discussed in the subsequent sections. The first three experiments help investigate the performance of the selected classifiers using the original features extracted from each of the datasets. Experiments 4, 5 and 6 discussed the results of the three classification algorithms when considering the proposed hybrid feature selection approach with CFS combined with Boruta. The last three experiments discussed the results of the classification algorithms on the proposed hybrid features by combining CFS with RFE.

Table 3. Summary of experiments.

S/No.	Description of experiments
1.	Experiment based on SVM with original features
2.	Experiment based on Naïve Bayes with original features
3.	Experiment based on Random Forest with original features
4.	Experiment based on SVM with hybrid features using CFS and Boruta
5.	Experiment based on Naïve Bayes with hybrid features using CFS and Boruta
6.	Experiment based on Random Forest with hybrid features using CFS and Boruta
7.	Experiment based on SVM with hybrid features using CFS and RFE
8.	Experiment based on Naïve Bayes with hybrid features using CFS and RFE
9.	Experiment based on Random Forest with hybrid features using CFS and RFE

4.2.1 Results of SVM with Original Features

To statistically understand the contribution of the proposed hybrid framework, each of the classifiers selected was evaluated with original features as extracted from the four datasets that were considered in this study. Table 4 shows the results of SVM with original feature subsets. SVM algorithm was evaluated on the four datasets without any feature selection technique. The results show that SVM achieved an accuracy of 80.1%, 75.3%, 74.2% and 66.8% on Amazon, Yelp, IMDB and Kaggle datasets, respectively. We observed that the performance of SVM classifier drops slightly when the number of features is increasing. For instance, SVM yielded an accuracy of 66.8% on the Kaggle dataset which has the highest number of features. The results obtained based on the other performance metrics using SVM classifier have been highlighted in Table 4.

Table 4. Results of SVM with original features.

		Datasets			
		Amazon	**Yelp**	**IMDB**	**Kaggle**
Algorithm	**# of Features**	**620**	**691**	**961**	**1218**
SVM	Accuracy	0.801	0.753	0.742	0.668
	Precision	**0.801**	**0.754**	**0.743**	**0.649**
	Recall	0.801	0.753	0.742	0.668
	F-Measure	0.801	0.753	0.742	0.654
	ROC	0.801	0.753	0.741	**0.687**

4.2.2 Results of Naïve Bayes with Original Features

Similarly, we evaluated the performance of Naïve Bayes classifier with original feature subsets. The results based on the four datasets are presented in Table 5. This results show a slight drop in the performance of Naïve Bayes when compared with SVM results in Table 4 based on the accuracy metric. However, the ROC results across the four datasets produced better results when compared with SVM in Table 4. ROC values of 80.9%, 78.8%, 78.5% and 71.1% were obtained on Amazon, Yelp, IMDB and Kaggle datasets, respectively. Similarly, we observed a similar pattern in the results produced by the classifier when the number of features is reduced.

Table 5. Results of Naïve Bayes with original features.

		Datasets			
		Amazon	**Yelp**	**IMDB**	**Kaggle**
Algorithm	**# of Features**	**620**	**691**	**961**	**1218**
Naïve Bayes	Accuracy	0.717	0.726	0.716	0.590
	Precision	0.717	0.735	0.716	0.61
	Recall	0.717	0.726	0.716	0.59
	F-Measure	0.717	0.723	0.716	0.598
	ROC	**0.809**	**0.788**	**0.785**	**0.711**

4.2.3 Results of Random Forest with Original Features

Random Forest classifier achieved the best result with original feature subsets based on the ROC metric. This result produced 86%, 85.2%, 81% and 77.8% for Amazon, Yelp, IMDB and Kaggle datasets, respectively. The result based on the Kaggle dataset also drops slightly when compared with other datasets as observed in the previous results. This result is shown in Table 6.

Table 6. Results of Random Forest with original features.

		Datasets			
		Amazon	**Yelp**	**IMDB**	**Kaggle**
	# of Features	**620**	**691**	**961**	**1218**
Random Forest	Accuracy	0.777	0.763	0.744	0.675
	Precision	0.777	0.769	0.746	0.657
	Recall	0.777	0.765	0.744	0.675
	F-Measure	0.777	0.700	0.743	0.658
	ROC	**0.86**	**0.852**	**0.81**	**0.778**

4.2.4 Results' Comparison Based on Original Features

To understand the variation in the results obtained with original feature subsets, this sub-section provides a detailed discussion of the results of the classification algorithms based on the four datasets considered.

(a) **Results of Classifiers with the Original Amazon Dataset:** Table 7 shows the results of three classification algorithms without using the proposed feature selection approach. SVM had the highest Accuracy of 80.1% using the original dataset, as against Naïve Bayes and Random Forest having 71.7% and 77.7%, respectively. It also had the highest F-Measure point of 0.801, using the original dataset, against Naïve Bayes and Random Forest that had 0.717 and 0.777, respectively. Using ROC metrics on the three classifiers, it was observed that Random Forest had a better ROC of 0.86 than SVM (0.801) and Naïve Bayes (0.809). Therefore, it was shown that SVM had better performance than Naïve Bayes and Random Forest in sentiment classification on Amazon dataset in terms of Accuracy, Precision, Recall and F-measure, while Random Forest slightly outperformed SVM and Naïve Bayes based on ROC metric.

Table 7. Results' comparison with original features based on Amazon dataset.

	# of Features	Accuracy (%)	Precision	Recall	F-Measure	ROC
SVM	620	**80.1**	**0.801**	**0.801**	**0.801**	0.801
Naïve Bayes	620	71.7	0.717	0.717	0.717	0.809
Random Forest	620	77.7	0.777	0.777	0.777	**0.860**

(b) **Results of Classifiers with the Original Yelp Dataset:** From Table 8, the performance of the three classifiers without using the proposed feature selection approach shows that Random Forest with an Accuracy of 76.25% outperforms both SVM (75.3%) and Naïve Bayes (72.6%). SVM has 0.753 value of F-Measure which is higher than the corresponding values for both Naïve Bayes and Random Forest classifiers with F-Measure of 0.723 and 0.700, respectively. Equally, Random Forest classifier has a higher ROC point (0.852) over both SVM (0.753) and Naïve Bayes (0.788) classifiers. Therefore, Random Forest classifier outperforms both SVM and Naïve Bayes classifiers on Yelp dataset by considering Accuracy, Precision, Recall and ROC. However, SVM has the highest F-Measure.

Table 8. Results' comparison with original features based on Yelp dataset.

	# of Features	Accuracy (%)	Precision	Recall	F-Measure	ROC
SVM	691	75.30	0.754	0.753	**0.753**	0.753
Naïve Bayes	691	72.60	0.735	0.726	0.723	0.788
Random Forest	691	**76.25**	**0.769**	**0.765**	0.700	**0.852**

(c) **Results of Classifiers with the Original IMDB Dataset:** Table 9 shows the results of the performances of the three classifiers on IMDB dataset without the application of the proposed feature selection approach. It was observed that Random Forest classifier has the records of 74.37% Accuracy, 0.743 F-Measure and 0.81 ROC, which outperforms the results obtained with SVM and Naïve Bayes. This results show that Random Forest classifier is a promising algorithm for sentiment analysis on IMDB dataset.

Table 9. Results' comparison with original features based on IMDB dataset.

	# of Features	Accuracy (%)	Precision	Recall	F-Measure	ROC
SVM	961	74.17	0.743	0.742	0.742	0.741
Naïve Bayes	961	71.57	0.716	0.716	0.716	0.785
Random Forest	961	**74.37**	**0.746**	**0.744**	**0.743**	**0.810**

(d) **Results of Classifiers with the Original Kaggle Dataset:** Table 10 shows the results of the three classifiers without the proposed feature selection approach. The best performance is recorded by Random Forest classifier which achieved an Accuracy of (67.53%), an F-Measure of (0.658) and a ROC metric of (0.778). This performance is followed by SVM classifier with an accuracy of 66.75%, an F-Measure of (0.654) and a ROC metric of (0.687). It is noteworthy that despite the least performance record credited to the Naïve Bayes classifier, it is observed that it outperforms SVM classifier based on ROC evaluation metric. This result implies that Random Forest classifier is still considered as the promising classification algorithm for sentiment analysis based on the Kaggle dataset.

Table 10. Results' comparison with original features based on Kaggle dataset.

	# of Features	Accuracy (%)	Precision	Recall	F-Measure	ROC
SVM	1218	66.75	0.649	0.668	0.654	0.687
Naïve Bayes	1218	59.02	0.61	0.59	0.598	0.711
Random Forest	1218	**67.53**	**0.657**	**0.675**	**0.658**	**0.778**

4.2.5 Results of the Hybrid Framework Using CFS+Boruta and SVM Classifier

Table 11 shows the results obtained using the proposed hybrid framework, which comprises the hybridization of CFS and Boruta algorithms. These results show that across the four datasets used in this study, SVM can provide an accuracy of 78.5%, 76.9%, 71.67% and 65.16% on Amazon, Yelp, IMDB and Kaggle datasets, respectively despite a significant reduction in the number of features for the classification task. This shows that the proposed hybrid framework produced promising results despite the huge reduction in the number of features considered. The features selected by the proposed CFS + Boruta feature selection method for each dataset using SVM classifier are shown in the Appendix.

Table 11. Results for SVM Classifier based on hybrid (CFS + Boruta) feature selection.

			Amazon	**Yelp**	**IMDB**	**Kaggle**
Hybrid Feature Selection Algorithm	**Classifier**	**# of Features**	620	691	961	1218
CFS + Boruta	SVM	Selected Features	**23**	**25**	**25**	**44**
		Accuracy (%)	**78.5**	**76.9**	**71.67**	**65.16**
		Precision	0.812	0.801	0.754	0.611
		Recall	0.785	0.769	0.717	0.652
		F-Measure	0.78	0.763	0.706	0.59
		ROC	**0.785**	**0.769**	**0.716**	**0.62**

4.2.6 Results of the Hybrid Framework Using CFS+Boruta and Naïve Bayes Classifier

Table 12 shows the results obtained using the proposed hybrid framework; that is, hybridization of CFS and Boruta algorithms. These results show that across the four datasets used in this study, Naïve Bayes can provide an accuracy of 74.1%, 71.7%, 71.37% and 63.38% on Amazon, Yelp, IMDB and Kaggle datasets, respectively. This result shows a slight drop in performance of Naïve Bayes when compared with SVM algorithm with the proposed hybrid framework that is based on CFS and Boruta algorithms. However, a significant improvement in the ROC metric was observed when Naïve Bayes was used as the classification algorithm, as shown in Table 12.

Table 12. Results for Naïve Bayes Classifier based on hybrid (CFS + Boruta) feature selection.

			Amazon	**Yelp**	**IMDB**	**Kaggle**
Hybrid Feature Selection Algorithm	**Classifier**	**# of Features**	620	691	961	1218
CFS + Boruta	Naïve Bayes	Selected Features	**23**	**25**	**25**	**44**
		Accuracy (%)	**74.1**	**71.7**	**71.37**	**63.38**
		Precision	0.783	0.765	0.727	0.594
		Recall	0.741	0.717	0.714	0.634
		F-Measure	0.731	0.704	0.709	0.582
		ROC	**0.791**	**0.794**	**0.76**	**0.691**

4.2.7 Results of the Hybrid Framework Using CFS+Boruta and Random Forest Cassifier

Random Forest algorithm when used with the proposed hybrid framework that is comprised of CFS and Boruta algorithm was able to produce the best results in terms of accuracy and ROC metrics, as shown in Table 13. According to this table, accuracies of 78.9%, 76.9%, 71.87% and 65.76% were obtained on Amazon, Yelp, IMDB and Kaggle datasets, respectively. The ROC results also show a considerable increase in the performance of Random Forest when compared with SVM and Naïve Bayes classifiers based on the four datasets considered in this study. This result further testifies to the applicability of the proposed hybrid framework for sentient analysis considering different problem domains. The result obtained is promising despite a significant reduction in the number of features as compared with the original datasets.

Table 13. Results for Random Forest Classifier based on hybrid (CFS + Boruta) feature selection.

			Amazon	**Yelp**	**IMDB**	**Kaggle**
Hybrid Feature Selection Algorithm	**Classifier**	**# of Features**	620	691	961	1218
CFS + Boruta	Random Forest	Selected Features	**23**	**25**	**25**	**44**
		Accuracy (%)	**78.9**	**76.9**	**71.87**	**65.76**
		Precision	0.816	0.801	0.756	0.621
		Recall	0.789	0.769	0.719	0.658
		F-Measure	0.784	0.763	0.708	0.594
		ROC	**0.808**	**0.801**	**0.776**	**0.711**

4.2.8 Results of the Hybrid Framework Using CFS+RFE and SVM Classifier

Next is to discuss the results obtained when using the proposed hybrid framework for feature selection for sentiment analysis that considered the hybridization of CFS and RFE algorithms to effectively select the most discriminative features. Table 14 shows the results obtained using this approach. Accuracies of 64.2%, 68.1%, 72.27% and 63.56% were obtained based on Amazon, Yelp, IMDB and Kaggle datasets, respectively. Hybridization using CFS and RFE algorithms selected the least number of features in most cases when compared with the number of features selected using CFS and Boruta algorithms. It can easily be seen that the results obtained with the CFS + Boruta outperformed those obtained when using the CFS + RFE method. This was also manifested in the results obtained when considering ROC metric. The features selected by the proposed CFS + RFE feature selection method for each dataset using SVM classifier are shown in the Appendix.

Table 14. Results for SVM Classifier based on hybrid (CFS + RFE) feature selection.

			Amazon	**Yelp**	**IMDB**	**Kaggle**
Hybrid Feature Selection Algorithm	**Classifier**	**# of Features**	620	691	961	1218
CFS + RFE	SVM	Selected Features	**3**	**7**	**70**	**22**
		Accuracy (%)	**64.2**	**68.1**	**72.27**	**63.56**
		Precision	0.738	0.742	0.725	0.536
		Recall	0.642	0.681	0.723	0.636
		F-Measure	0.602	0.66	0.722	0.535
		ROC	**0.642**	**0.681**	**0.722**	**0.577**

4.2.9 Results of the Hybrid Framework Using CFS+RFE and Naïve Bayes Classifier

Table 15 shows the results of Naïve Bayes classifier based on hybridization of CFS and RFE algorithms. These results show accuracies of 64.2%, 68.2%, 73.87% and 62.71% on Amazon, Yelp, IMDB and Kaggle datasets, respectively. This result still shows a reduction in performance when compared with the results obtained using CFS and Boruta algorithms with Naïve Bayes classifier.

4.2.10 Results of the Hybrid Framework Using CFS+RFE and Random Forest Classifier

Table 16 shows the results of the Random Forest algorithm based on CFS and RFE hybridization. These results show accuracies of 64.2%, 68.2%, 76.38% and 63.94% on Amazon, Yelp, IMDB and

Table 15. Results for Naïve Bayes Classifier based on hybrid (CFS + RFE) feature selection.

			Amazon	**Yelp**	**IMDB**	**Kaggle**
Hybrid Feature Selection Algorithm	**Classifier**	**# of Features**	620	691	961	1218
CFS + RFE	Naïve Bayes	Selected Features	**3**	**7**	**70**	**22**
		Accuracy (%)	**64.2**	**68.2**	**73.87**	**62.71**
		Precision	0.738	0.744	0.75	0.703
		Recall	0.642	0.682	0.739	0.627
		F-Measure	0.602	0.66	0.736	0.54
		ROC	**0.631**	**0.7**	**0.818**	**0.668**

Kaggle datasets, respectively. When compared with the results of the Random Forest classifier based on hybridization of CFS and Boruta algorithms, there is a significant drop in performance based on CFS and RFE approach proposed in this study. Similar results were obtained using the ROC metric as an evaluation metric.

Table 16. Results for Random Forest classifier based on hybrid (CFS + RFE) feature selection.

			Amazon	**Yelp**	**IMDB**	**Kaggle**
Hybrid Feature Selection Algorithm	**Classifier**	**# of Features**	620	691	961	1218
CFS + RFE	Random Forest	Selected Features	**3**	**7**	**70**	**22**
		Accuracy (%)	**64.2**	**68.2**	**76.38**	**63.94**
		Precision	0.738	0.744	0.779	0.565
		Recall	0.642	0.682	0.764	0.639
		F-Measure	0.602	0.66	0.761	0.543
		ROC	**0.631**	**0.697**	**0.852**	**0.679**

4.2.11 Results' Comparison of the Classifiers Based on the Hybrid Framework

In this sub-section, we now compare the results of the two proposed hybrid methods (CFS + Boruta and CFS + RFE) based on the individual datasets.

(a) Amazon Dataset

The results in Table 17 show that the proposed hybridization method using CFS + Boruta selected 23 features on Amazon dataset as compared with 620 features available in the original dataset, while CFS + RFE approach selected 3 features. The classification results revealed that Random Forest classifier outperformed the other two classification algorithms based on all the metrics used for evaluation in this study. Using 23 reduced features based on CFS + Boruta, Random Forest was able to produce the following metrics: Accuracy (78.9%), Precision (81.60%), Recall (78.9%), F-measure (78.4%) and ROC (80.80%). This result is promising when considering the number of features used for the classification task.

Table 17. Performance evaluation of the hybrid feature selection framework based on Amazon dataset.

	# of Features	**Accuracy (%)**	**Precision**	**Recall**	**F-Measure**	**ROC**
CFS+Boruta+SVM	23	78.5	0.812	0.785	0.78	0.785
CFS+Boruta+NB	23	74.1	0.783	0.741	0.731	0.791
CFS+Boruta+RF	**23**	**78.9**	**0.816**	**0.789**	**0.784**	**0.808**
CFS+RFE+SVM	3	64.2	0.738	0.642	0.602	0.642
CFS+RFE+NB	3	64.2	0.738	0.642	0.602	0.631
CFS+RFE+RF	3	64.2	0.738	0.642	0.602	0.631

(b) Yelp Dataset

The results in Table 18 show that the proposed hybridization method using CFS + Boruta selected 25 features on Yelp dataset as compared with 691 features available in the original dataset, while CFS + RFE approach selected 7 features. The classification results revealed that Random Forest classifier outperformed the other two classification algorithms based on all the metrics used for evaluation. With the 25 reduced features based on CFS + Boruta, Random Forest was able to produce the following metrics: Accuracy (76.9%), Precision (80.10%), Recall (76.9%), F-measure (76.3%) and ROC (80.10%). SVM classifier also produced similar results as compared with Random Forest classifier except for the ROC of Random Forest classifier that is slightly higher than the one obtained with SVM. Furthermore, this result is promising when considering the number of features used for the classification task.

Table 18. Performance evaluation of the hybrid feature selection framework based on Yelp dataset.

	# of Features	Accuracy (%)	Precision	Recall	F-Measure	ROC
CFS+Boruta+SVM	25	76.9	0.801	0.769	0.763	0.769
CFS+Boruta+NB	25	71.7	0.765	0.717	0.704	0.794
CFS+Boruta+RF	**25**	**76.9**	**0.801**	**0.769**	**0.763**	**0.801**
CFS+RFE+SVM	7	68.1	0.742	0.681	0.66	0.681
CFS+RFE+NB	7	68.2	0.744	0.682	0.66	0.7
CFS+RFE+RF	7	68.2	0.744	0.682	0.66	0.697

(c) IMDB Dataset

The results in Table 19 show that the proposed hybridization method using CFS + Boruta selected 25 features on IMDB dataset as compared with 961 features available in the original dataset, while CFS + RFE approach selected 70 features. This result shows an increase in the number of features selected by RFE when compared with the previous results. The classification results revealed a different scenario in which CFS + RFE outperformed CFS + Boruta according to the results of the RFE classifier based on the evaluation metrics. With the 70 reduced features based on CFS + RFE, Random Forest was able to produce the following metrics: Accuracy (76.38%), Precision (77.9%), Recall (76.4%), F-measure (76.1%) and ROC (85.20%).

Table 19. Performance evaluation of the hybrid feature selection framework based on IMDB dataset.

	# of Features	Accuracy (%)	Precision	Recall	F-Measure	ROC
CFS+Boruta+SVM	25	71.67	0.754	0.717	0.706	0.716
CFS+Boruta+NB	25	71.37	0.727	0.714	0.709	0.76
CFS+Boruta+RF	25	71.87	0.756	0.719	0.708	0.776
CFS+RFE+SVM	70	72.27	0.725	0.723	0.722	0.722
CFS+RFE+NB	70	73.87	0.75	0.739	0.736	0.818
CFS+RFE+RF	**70**	**76.38**	**0.779**	**0.764**	**0.761**	**0.852**

(d) Kaggle Dataset

The results in Table 20 show that the proposed hybridization method using CFS + Boruta selected 44 features on the Kaggle dataset as compared with 1218 features available in the original dataset, while CFS + RFE approach selected 22 features. The classification results revealed that Random Forest classifier still outperformed the other two classification algorithms as revealed by the results of the different evaluation metrics. Using the reduced 44 features based on CFS + Boruta showed that Random Forest classifier was able to produce the following metrics: Accuracy (65.76%), Precision (62.10%), Recall (65.80%), F-measure (59.40%) and ROC (71.10%). Taking a closer look at the results in this table, it can be seen that SVM classifier also achieved very close results with Random

Forest algorithm. Besides, it was noticed that the Precision (70.30%), accounting for the highest value of precision for all the classification cases, was obtained with CFS + RFE + NB classifier. This is the only scenario where Naïve Bayes outperformed the other classification algorithms from the various results obtained in this study based on the four datasets considered for analysis. The results in Table 20 further strengthen the superiority of the proposed CFS + Boruta hybrid algorithm based on Random Forest classifier.

Table 20. Performance evaluation of the hybrid feature selection framework based on Kaggle dataset.

	# of Features	Accuracy (%)	Precision	Recall	F-Measure	ROC
CFS+Boruta+SVM	44	65.16	0.611	0.652	0.59	0.62
CFS+Boruta+NB	44	63.38	0.594	0.634	0.582	0.691
CFS+Boruta+RF	**44**	**65.76**	**0.621**	**0.658**	**0.594**	**0.711**
CFS+RFE+SVM	22	63.56	0.536	0.636	0.535	0.577
CFS+RFE+NB	22	62.71	0.703	0.627	0.54	0.668
CFS+RFE+RF	22	63.94	0.565	0.639	0.543	0.679

4.3 Results' Comparison with and without the Proposed Hybrid Algorithm

In this sub-section, we compare the results of the classification algorithms with and without the application of the proposed hybrid feature selection algorithm. More specifically, we selected the best results obtained in each case for analysis. This involves comparing the best results of the classifiers on the original features and also on the feature subsets selected by the proposed hybrid feature selection algorithm. The results' analyses have been grouped under the dataset used for the experiment in each scenario.

(a) Results from Comparison Based on Amazon Dataset with and without Hybrid Feature Selection

Figure 2 shows the results obtained with and without the proposed hybrid feature selection method. According to this figure, Accuracy (80.1%), Precision (80.1%), Recall (80.1%), F-measure (80.1%) and ROC (80.1%) were obtained based on SVM algorithm with original 620 features available on Amazon dataset, while Accuracy (78.9%), Precision (81.6%), Recall (78.9%), F-measure (78.4%) and ROC (80.8%) were obtained when the proposed hybrid feature selection algorithm was used based on 23 features. This result shows a significant improvement when considering the number of irrelevant features that have been removed from the original Amazon dataset. Specifically, the noticeable achievement was observed in the ROC result when the proposed hybrid feature selection algorithm was used. A similar thing was noticed for the Precision result. By considering the percentage of reduction in the number of features (96.29%), the results obtained with the proposed hybrid feature selection are promising and further confirmed the superiority and applicability of the proposed hybrid feature selection method.

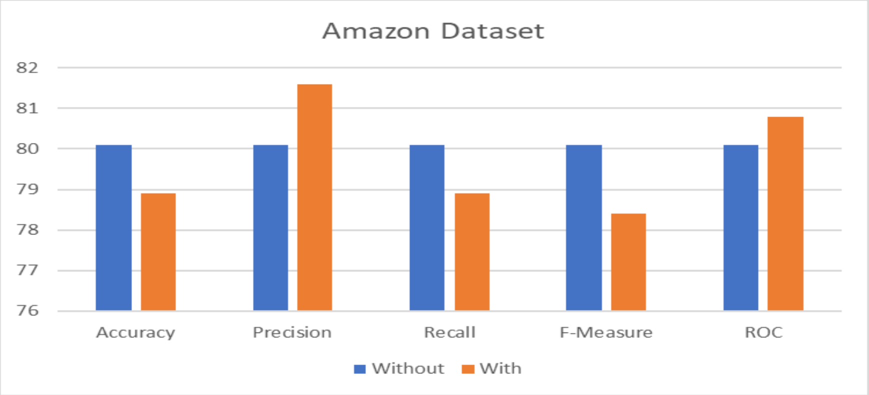


Figure 2. Results of comparison based on Amazon dataset with and without hybrid feature selection.

(b) Results' Comparison Based on Yelp Dataset with and without Hybrid Feature Selection

Figure 3 shows the results obtained with and without the proposed hybrid feature selection method using Yelp dataset. According to this figure, Accuracy (76.25%), Precision (76.9%), Recall (76.5%), F-measure (70.00%) and ROC (85.2%) were obtained based on Random Forest algorithm with original 691 features available on Yelp dataset, while Accuracy (76.9%), Precision (80.1%), Recall (76.9%), F-measure (76.3%) and ROC (80.1%) were obtained when the proposed hybrid feature selection algorithm (CFS + Boruta) was used based on 25 features. This result shows a significant improvement when considering the number of irrelevant features that have been removed from the original Yelp dataset. More importantly is the improvement cut across the different evaluation metrics used in this study as shown in the figure. However, the ROC result of the hybrid framework drops slightly when compared with the ROC result obtained based on original features. By considering the percentage of reduction in the number of features (96.38%), the results obtained with the proposed hybrid feature selection are promising and superior. Also, this further confirmed the superiority and applicability of the proposed hybrid feature selection method.

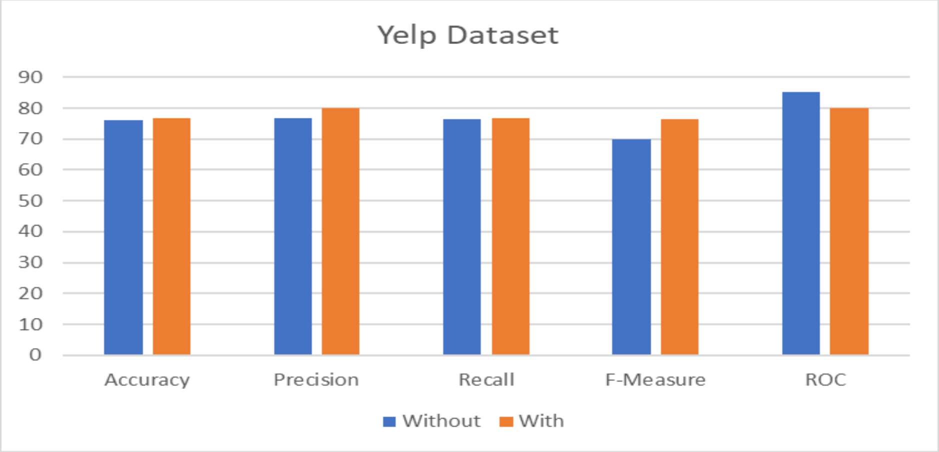


Figure 3. Results of comparison based on Yelp dataset with and without hybrid feature selection.

(c) Results' Comparison Based on IMDB Dataset with and without Hybrid Feature Selection

As shown in Figure 4, the results obtained with the proposed hybrid feature selection method outperformed the results achieved when feature selection was not used on IMDB dataset. According to this figure, Accuracy (74.37%), Precision (74.46%), Recall (74.4%), F-measure (74.30%) and ROC (81.0%) were obtained based on Random Forest algorithm with original 961 features available on IMDB dataset, while Accuracy (76.38%), Precision (77.9%), Recall (76.40%), F-measure (76.10%) and ROC (85.2%) were obtained when the proposed hybrid feature selection algorithm was used. This result shows a significant improvement when considering the number of irrelevant features that have been removed from the original IMDB dataset. More importantly is the improvement cut across the

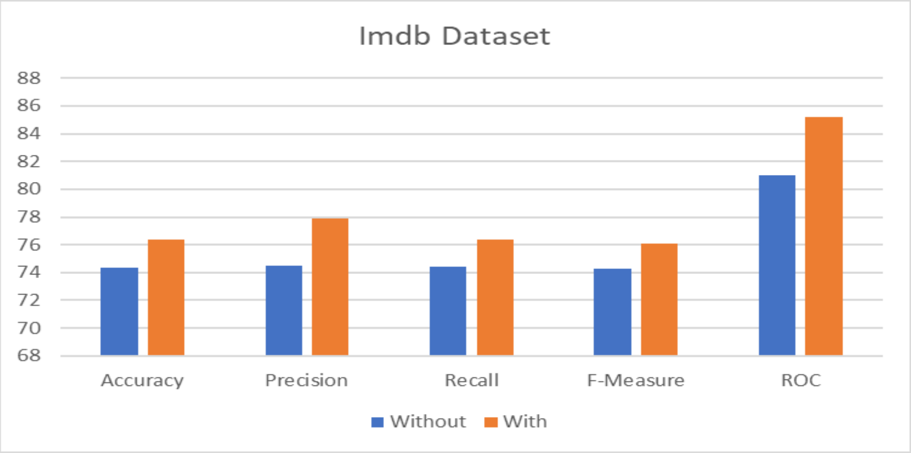


Figure 4. Results of comparison based on IMDB dataset with and without hybrid feature selection.

different evaluation metrics used in this study as shown in the figure. Based on the percentage of reduction in the number of features (92.72%), the results obtained with the proposed hybrid feature selection are promising and superior. Also, this further confirmed the superiority and applicability of the proposed hybrid feature selection method.

(d) Results' Comparison Based on Kaggle Dataset with and without Hybrid Feature Selection

Figure 5 shows the results obtained with and without the proposed hybrid feature selection method using the Kaggle dataset. About 96.39% reduction in the original features available in the dataset was achieved. According to this figure, Accuracy (67.53%), Precision (65.7%), Recall (67.5%), F-measure (65.8%) and ROC (77.8%) were obtained based on Random Forest algorithm with original 1218 features available on the Kaggle dataset, while Accuracy (65.76%), Precision (62.1%), Recall (65.8%), F-measure (59.4%) and ROC (71.1%) were obtained when the proposed hybrid feature selection algorithm (CFS + Boruta) was used based on 44 features. The results obtained without the use of the proposed hybrid feature selection algorithm dropped on the Kaggle dataset. However, by considering the number of features used for the classification, the results obtained across the different evaluation metrics still show the applicability of the proposed hybrid feature selection algorithm to reduce model complexity while still achieving comparable performance with the original features.

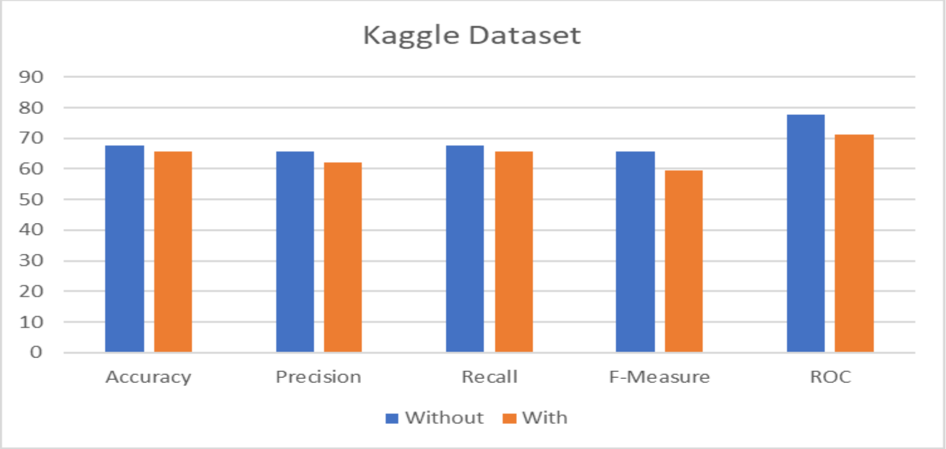


Figure 5. Results of comparison based on Kaggle dataset with and without hybrid feature selection.

5. Conclusions

Sentiment analysis studies have been receiving wide attention in recent years, mostly due to their significant role in opinion mining and prediction. Sentiment analysis has helped organizations understand customers' opinions as well as their attitudes towards particular products or brands. In this study, a hybrid feature selection framework was proposed to address the research issue of identifying discriminating attributes that can be used to model customers' opinions across different domains. This study employed four public datasets (Amazon, Yelp, IMDB and Kaggle) to examine the applicability of the proposed hybrid feature selection framework for sentiment analysis. The proposed hybrid feature selection framework has two levels of feature selection strategies by employing filter- and wrapper-based feature selection. Correlation-based feature selection (CFS) with Best First Search (BFS) method has been used at the top layer of the proposed framework and the bottom layer is based on the combination of CFS with either Boruta or Recursive Feature Elimination (RFE) wrapper-based feature selection method. The goal is to examine the specific combination of the feature selection approach that will provide improved performance for sentiment analysis across different domains. Therefore, based on several experiments conducted using three classification algorithms: SVM, Naïve Bayes and Random Forest, the study was able to establish the superiority and applicability of the proposed hybrid feature selection framework using well-known evaluation metrics.

CFS combined with Boruta produced the most promising results. Therefore, it is recommended to build a prototype for sentiment analysis tasks across different domains. The proposed hybrid feature selection approach reduced the number of features from the original datasets by 95% on average while still maintaining promising classification results. It was observed that the Random Forest algorithm demonstrated superiority over the two other classifiers by producing interesting results across the four datasets used in this study. Despite a considerable reduction in the number of features used for classification, performance figures are somewhat lower, but relatively on par with evaluation using the full feature set, but the computing time of the resulting model is shorter as a result of the proposed hybrid feature selection framework. As observed in the results produced during this study, there is still the need to focus on improving the classification accuracy of the proposed framework while still ensuring that model complexity is reduced to save prediction time. Also, future work should further investigate other feature selection strategies, which may likely produce better results when still considering domain-specific sentiment analysis.

REFERENCES

[1] M. A. Hassonah, R. Al-Sayyed, A. Rodan, A.-Z. Ala'm, I. Aljarah and H. Faris, "An Efficient Hybrid Filter and Evolutionary Wrapper Approach for Sentiment Analysis of Various Topics on Twitter," Knowledge-based Systems, vol. 192, p. 105353, 2020.

[2] Y. A. Alsariera, A. V. Elijah and A. O. Balogun, "Phishing Website Detection: Forest by Penalizing Attributes Algorithm and Its Enhanced Variations," Arabian Journal for Science and Engineering, vol. 45, pp. 10459-10470, 2020.

[3] S. M. Rezaeinia, R. Rahmani, A. Ghodsi and H. Veisi, "Sentiment Analysis Based on Improved Pre-trained Word Embeddings," Expert Systems with Applications, vol. 117, pp. 139-147, 2019.

[4] R. Arulmurugan, K. Sabarmathi and H. Anandakumar, "Classification of Sentence Level Sentiment Analysis Using Cloud Machine Learning Techniques," Cluster Comp., vol. 22, pp. 1199-1209, 2019.

[5] A. Hasan, S. Moin, A. Karim and S. Shamshirband, "Machine Learning-based Ssentiment Analysis for Twitter Accounts," Mathematical and Computational Applications, vol. 23, p. 11, 2018.

[6] H. H. Do, P. Prasad, A. Maag and A. Alsadoon, "Deep Learning for Aspect-based Sentiment Analysis: A Comparative Review," Expert Systems with Applications, vol. 118, pp. 272-299, 2019.

[7] Y. Wang, M. Wang and H. Fujita, "Word Sense Disambiguation: A Comprehensive Knowledge Exploitation Framework," Knowledge-based Systems, vol. 190, p. 105030, 2020.

[8] M. Taboada, J. Brooke, M. Tofiloski, K. Voll and M. Stede, "Lexicon-based methods for sentiment analysis," Computational Linguistics, vol. 37, pp. 267-307, 2011.

[9] G. Ansari, T. Ahmad and M. N. Doja, "Hybrid Filter–Wrapper Feature Selection Method for Sentiment Classification," Arabian Journal for Science and Engineering, vol. 44, pp. 9191-9208, 2019.

[10] Y. A. Alsariera, V. E. Adeyemo, A. O. Balogun and A. K. Alazzawi, "AI Meta-learners and Extra-trees Algorithm for the Detection of Phishing Websites," IEEE Access, vol. 8, pp. 142532-142542, 2020.

[11] M. S. Akhtar, D. Gupta, A. Ekbal and P. Bhattacharyya, "Feature Selection and Ensemble Construction: A Two-step Method for Aspect Based Sentiment Analysis," Knowledge-Based Systems, vol. 125, pp. 116-135, 2017.

[12] A. O. Balogun, S. Basri, S. Mahamad, S. J. Abdulkadir, M. A. Almomani, V. E. Adeyemo et al., "Impact of Feature Selection Methods on the Predictive Performance of Software Defect Prediction Models: An Extensive Empirical Study," Symmetry, vol. 12, p. 1147, 2020.

[13] B. A. Oluwagbemiga, B. Shuib, S. J. Abdulkadir and A. Sobri, "A Hybrid Multi-filter Wrapper Feature Selection Method for Software Defect Predictors," International Journal of Supply Chain Management, vol. 8, pp. 916-922, 2019.

[14] A. O. Balogun, S. Basri, S. J. Abdulkadir and A. S. Hashim, "Performance Analysis of Feature Selection Methods in Software Defect Prediction: A Search Method Approach," Applied Sciences, vol. 9, p. 2764, 2019.

[15] B. Agarwal and N. Mittal, "Machine Learning Approach for Sentiment Analysis," Proc. of Prominent Feature Extraction for Sentiment Analysis, pp. 21-45, Springer, 2016.

[16] K. S. Adewole, T. Han, W. Wu, H. Song and A. K. Sangaiah, "Twitter Spam Account Detection Based on Clustering and Classification Methods," The Jour. of Supercomputing, vol. 76, pp. 4802-4837, 2020.

[17] L. Zhang and B. Liu, "Sentiment analysis and opinion mining," Encyclopedia of Machine Learning and Data Mining, pp. 1152-1161, 2017.

[18] B. Liu, Sentiment Analysis: Mining Opinions, Sentiments and Emotions, Cambridge Uni. Press, 2020.

[19] S. Ahmed and A. Danti, "Effective Sentimental Analysis and Opinion Mining of Web Reviews Using Rule Based Classifiers," Proc. of Computational Intelligence in Data Mining, vol. 1, pp. 171-179, Springer, 2016.

[20] F. Hemmatian and M. K. Sohrabi, "A Survey on Classification Techniques for Opinion Mining and Sentiment Analysis," Artificial Intelligence Review, vol. 52, pp. 1495-1545, 2019.

[21] E. Cambria, D. Das, S. Bandyopadhyay and A. Feraco, "Affective Computing and Sentiment Analysis," Proc. of a Practical Guide to Sentiment Analysis, pp. 1-10, Springer, 2017.

[22] M. Ptaszynski, R. Rzepka, K. Araki and Y. Momouchi, "Automatically Annotating a Five-billion-word Corpus of Japanese Blogs for Sentiment and Affect Analysis," Computer Speech & Language, vol. 28, pp. 38-55, 2014.

[23] E. Cambria, Y. Li, F. Z. Xing, S. Poria and K. Kwok, "SenticNet 6: Ensemble Application of Symbolic and Subsymbolic AI for Sentiment Analysis," Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pp. 105-114, DOI: 10.1145/3340531.3412003, 2020.

[24] E. Cambria, S. Poria, A. Gelbukh and K. Kwok, "Sentic API: A Common-sense Based API for Concept-level Sentiment Analysis," Proc. of Making Sense of Microposts (# Microposts2014), p. 2, [Online], Available: https://hdl.handle.net/10356/84835, 2014.

[25] A. Jeyapriya and C. K. Selvi, "Extracting Aspects and Mining Opinions in Product Reviews Using Supervised Learning Algorithm," Proc. of the 2nd IEEE International Conference on Electronics and Communication Systems (ICECS), pp. 548-552, Coimbatore, India, 2015.

[26] A. Tripathy, A. Agrawal and S. K. Rath, "Classification of Sentimental Reviews Using Machine Learning Techniques," Procedia-Computer Science, vol. 57, pp. 821-829, 2015.

[27] C. Alfaro, J. Cano-Montero, J. Gómez, J. M. Moguerza and F. Ortega, "A Multi-stage Method for Content Classification and Opinion Mining on Weblog Comments," Annals of Operations Research, vol. 236, pp. 197-213, 2016.

[28] A. Hussain and E. Cambria, "Semi-supervised Learning for Big Social Data Analysis," Neurocomputing, vol. 275, pp. 1662-1673, 2018.

[29] N. Claypo and S. Jaiyen, "Opinion Mining for Thai Restaurant Reviews Using K-Means Clustering and MRF Feature Selection," Proc. of the 7th IEEE International Conference on Knowledge and Smart Technology (KST), pp. 105-108, Chonburi, Thailand, 2015.

[30] I. Al-Agha and O. Abu-Dahrooj, "Multi-level Analysis of Political Sentiments Using Twitter Data: A Case Study of the Palestinian-Israeli Conflict," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 5, no.3, pp. 195-215, 2019.

[31] 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 (JJCIT), vol. 5, no. 1, pp. 43-59, 2019.

[32] K. M. Nahar, A. Jaradat, M. S. Atoum and F. Ibrahim, "Sentiment Analysis and Classification of Arab Jordanian Facebook Comments for Jordanian Telecom Companies Using Lexicon-based Approach and Machine Learning," Jordanian Jour. of Comp. and Inf. Tech. (JJCIT), vol. 6, no.3, pp. 247-262, 2020.

[33] F. Degenhardt, S. Seifert and S. Szymczak, "Evaluation of Variable Selection Methods for Random Forests and Omics Datasets," Briefings in Bioinformatics, vol. 20, pp. 492-503, 2019.

[34] S. S. Rathore and A. Gupta, "A Comparative Study of Feature-ranking and Feature-subset Selection Techniques for Improved Fault Prediction," Proceedings of the 7th India Software Engineering Conference, pp. 1-10, Chennai, India, 2014.

[35] Z. Xu, J. Liu, Z. Yang, G. An and X. Jia, "The Impact of Feature Selection on Defect Prediction Performance: An Empirical Comparison," Proc. of the IEEE 27th International Symposium on Software Reliability Engineering (ISSRE), pp. 309-320, Ottawa, Canada, 2016.

[36] M. B. Kursa and W. R. Rudnicki, "Feature Selection with the Boruta Package," J. Stat. Softw., vol. 36, pp. 1-13, 2010.

[37] A. O. Balogun, S. Basri, S. J. Abdulkadir, V. E. Adeyemo, A. A. Imam and A. O. Bajeh, "Software Defect Prediction: Analysis of Class Imbalance and Performance Stability," Journal of Engineering Science and Technology, vol. 14, pp. 3294-3308, 2019.

APPENDIX: (a) Selected features using CFS + Boruta and (b) Selected features using CFS + RFE.

(a) Dataset/No. of features	Features
Amazon (23)	awesom beauti best charm comfort disappoint excel fine flawless good great happier love nice perfect poor price rock seller setup sturdy wast well
Yelp (25)	amaz awesom bad bread delici delight don’t excel fantast friend fun good great happi incred love minut mouth nice outstand perfect town wasn’t white wonder
IMDB (25)	actual amaz bad beauti best brilliant cast cinema enjoy entertain excel film funni great love nice perform play portray right stupid terribl wast will wonder
Kaggle (44)	attent best carlyfiorina carson democraticdeb don’t character enjoy favorit fox goldietaylor gopdeb great httptcospzaa imwithhuck jeb job johnkasich just kasich look love lrihendri marcorubio mostretweet nail rate reaction realbencarson realdonaldtrump realli rubio rwsurfer girl superman hotmal tedcruz thank transcript truth until via vine winner women won

(b) Dataset/No. of features	Features
Amazon (3)	great good price
Yelp (7)	great good love delici don’t friend amaz
IMDB (70)	bad film love great even play wonder wast best enjoy cast didn’t excel stupid beauti bore funni terribl perform will worst portray wors hour job amaz suck start actual disappoint cinema role right mess avoid lack subtl poor brilliant cool horribl histori fail cheap annoy ridicul hole nice crap fine lame entertain impress yet pathet hope tortur hilari fun joy thriller tom low unbeliev three fascin dislik trash wouldn’t embarrass
Kaggle (22)	rwsurfergirl character realdonaldtrump fox tedcruz rubio gopdeb thank look jeb just job don’t rate great carlyfiorina best love lrihendri carson enjoy women

ملخص البحث:

اجتذب تحليل العواطف حديثاً اهتماماً ملحوظاً في السّنوات الأخيرة؛ بسبب ما له من قابليّة للتّطبيق في تحديد آراء المستخدمين وعواطفهم وانفعالاتهم من مجموعات بياناتٍ ضخمة تحتوي على كمّ هائل من النّصوص. ويتركّز الهدف من التّحليل العاطفي على تحسين خبرة المستخدمين عن طريق توظيف تقنيات متينة يمكنها التّنقيب عن الآراء والعواطف من مجموعات البيانات الضّخمة. وهناك دراسات متعددة تناولت التّحليل العاطفي والتّنقيب عن الآراء من المعلومات النصية. ومع ذلك، فإنّ وجود كلماتٍ خاصةٍ بمجالٍ ما دون غيره؛ الى جانب اللّهجات العاميّة والاختصارات والاخطاء القواعدية، فرض تحدّيات جدّية إضافية على طرق التّحليل العاطفي القائمة. في هذه الورقة، نركّز على تحديد مجموعة فرعيّة مميزة فعّالة من المميزات التي تساعد في تصنيف آراء المستخدمين من مجموعات البيانات الضّخمة. تقترح هذه الدراسة إطاراً هجيناً لانتقاء المميزات مبنياً على تهجين طريقة اختيار المميزات القائمة على الفلترة وطريقة اختيار المميزات القائمة على الحَجْب. ويستخدم انتقاء المميزات المستند الى الارتباط (CFS)، المهجّن مع بوروتا (Boruta) وإزالة المميزات الثانوية (RFE) من أجل تحديد المجموعات الفرعيّة من المميزات الأكثر تمييزاً للتّحليل العاطفي. وقد تمّ اعتبار أربعٍ من مجموعات البيانات المتاحة للعموم هي: (Amazon، Yelp، و IMDB، و Kaggleٍ) من أجل تقييم أداء إطار انتقاء المميزات الهجين المقترح. وتعمل هذه الدراسة على تقييم أداء ثلاث خوارزميات تصنيف هي: (SVM، و NB، وRF) للتحقق من تفوُّق النّظام المقترح. وبينت نتائج التجارب العملية المجراة على مجموعات البيانات والخوارزميات المذكورة آنفاً، أن انتقاء المميزات المستند الى الارتباط (CFS) المستخدم مع بوروتا (Boruta) نتجت عنه نتائج واعدة وخصوصاً عندما يتم تمرير المميزات المختارة الى خوارزمية (RF). وفي الحقيقة، فإن الإطار الهجين المقترح يقدم طريقةً فعالةً لتوقُّع آراء المستخدمين وعواطفهم مع اعتبارٍ أساسيٍ لدقة التوقُّع. وتجدر الإشارة الى أنّ زمن الحساب للنموذج الناتج أقصر نتيجةً لإطار انتقاء المميزات الهجين المقترح.