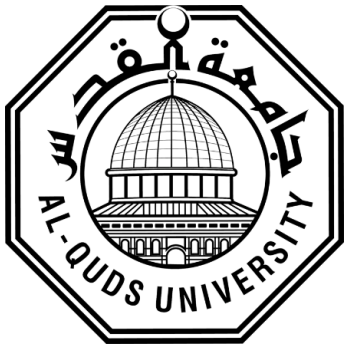
Deanship of Graduate Studies
Al-Quds University



Comparative Study on Feature Selection and Ensemble Methods for Sentiment Analysis Classification

Zahir Mohammad Adnan Younis

M.Sc Thesis

Jerusalem-Palestine

1441-2020

Comparative Study on Feature Selection and Ensemble Methods for Sentiment Analysis Classification

Prepared By:

Zahir Mohammad Adnan Younis

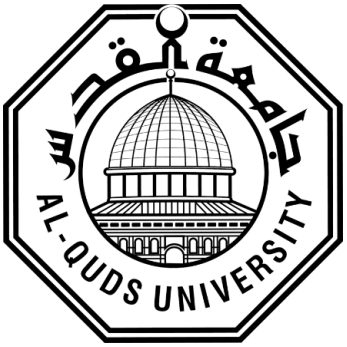
B.Sc. Computer Science, Al-Najah National University, Palestine

Supervisor: Dr. Nidal Kafri

A thesis submitted in partial fulfilment of the requirements for the degree of Master of Computer Science - Department of Computer Science - Faculty of Science and Technology - Al Quds University.

1441 - 2020

Al-Quds University
Deanship of Graduate Studies
Master of Computer Science



Thesis Approval

Comparative Study on Feature Selection and Ensemble Methods for Sentiment Analysis Classification

Prepared By: Zahir Mohammad Adnan Younis
Registration No: 21610087

Supervisor: Dr. Nidal Kafri

Master thesis submitted and accepted, Date: June 10, 2020

The name and signatures of examining committee members are as follows:

1 Head of Committee: Dr. Nidal Kafri Signature: .....

2 Internal Examiner: Dr. Saeed Salah Signature: .....

3 External Examiner: Dr. Radi Jarrar Signature: .....

Jerusalem-Palestine

1441 – 2020

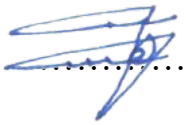
Dedication

I dedicate my thesis to my dearest parents, my wife Aaeda, my children Haya, Lama, Jana, Murad, and Lana.

Zahir Mohammad Adnan Younis

Declaration

I certify that this thesis submitted for the degree of Master, is the result of my own research, except where otherwise acknowledged, and that this study (or any part of the same) has not been submitted for a higher degree to any other university or institution.

Signed……  ……

Zahir Mohammad Adnan Younis

Date: June 10, 2020

Acknowledgment

First and foremost, praise be to Allah, the Almighty, for his blessings, grace, and generosity.

I would like to express my deepest gratitude to my research supervisor, Dr. Nidal Kafri, for his support and invaluable guidance. Having the opportunity to work with such a great scholar, teacher, and researcher of his stature was a blessing.

My utmost sincere thanks go to all members of faculty at the Department of Computer Science at Al-Quds University, your hard work and dedication in teaching us is admirable.

Lastly, I am extremely grateful to my parents, my siblings, my wife, and my children for their support, understanding, sacrifices, and prayers. Thank you for being the inspiration that guides me through life.

Abstract

People use the Web and social media to express their opinions and comments on various topics and posts generating huge amounts of data. Hence, comes the necessity to analyze this large amount of text regarding a certain subject and figuring out what people think of it. The interest and necessity of this analysis is continuously rising in many fields, such as politics, marketing, entertainment, sports, etc., to figure out people opinions, thinking, interests, preferences, and trends. Consequently, analysis, classification and clustering of this huge amount of text data regarding certain subjects became an interest of a vast number of researchers and beneficiaries. This analysis of text data content is known as sentiment analysis.

Sentiment Analysis (SA) is a text-mining field that computationally treats and analyses these sentiments (opinions, thinks, subjectivity, interests, preferences, etc.,.) of available text. SA aims to classify expressions in a text as positive, negative or neutral opinion towards the subject of interest.

The main objective of this research is to carry out a comparative study on the accuracy and performance of feature selection and ensemble methods for SA classification. The comparison was carried out using different combinations of classification algorithms for classifying text to being either positive or negative.

During the comparison of the algorithms and methods, the results showed that better accuracy can be achieved based on the used feature selection method (i.e., statistical, wrapper, or embedded). Additionally, it showed which feature selection method outperforms and is more suitable than other methods for the type of data and classification algorithms. Furthermore, when using combined ensemble methods (Bagging, Boosting, Stacking and Vote) performed better than using a single classifier by means of accuracy. Moreover, merging feature subsets selected by embedded method improved classification accuracy. Finally, tuning the parameters of feature selection methods improved the classification accuracy and reduced the time needed to select feature subsets.

Particularly, the results showed that accuracy depends on the feature selection method, ensemble methods, number of selected features, type of classifier, and tuning parameters of the algorithms used. A high accuracy of up to 99.85% was achieved by merging features of two embedded methods when using stacking ensemble method. Also, a high accuracy of 99.5% was achieved by tuning parameters in stacking method, and it reached 99.95% and

100% by tuning parameters in SVMAttributeEval method using statistical and machine learning approaches, respectively. Furthermore, tuning algorithms' parameters reduced the time needed to select feature subsets.

Table of Contents

List of Tables

List of Figures

List of Equations

List of Abbreviations

SA	Sentiment Analysis
OM	Opinion Mining
TF	Term frequency
DF	Document Frequency
TF-IDF	Term Frequency-Inverse Document Frequency
VSM	Vector Space Model
ME	Maximum Entropy
IG	Information Gain
J48	Decision Tree
NB	Naive Bayes
CHI	Chi Square
SVM	Support Vector Machine
REPTree	Reduced Error Pruning Tree
MI	Mutual Information
GA	Genetic Algorithm
mRMR	Minimum Redundancy and Maximum Relevance
EL	Ensemble Learning
Bagging	Bootstrap Aggregating
FN	False-negative
FP	False-positive
TN	True-Negative
TP	True-positive
Parameter	percentToEliminatePerIteration

Chapter1

Introduction

Chapter one contains a brief introduction of this research. It presents the problem statement, some previous related works, motivation, our contribution, and thesis organization.

1.1 Introduction

Web sites are important in human life, that help connect people to each other, through social networking sites that provides commenting on public posted material. People express their opinions on various topics. Many posts and comments about the news, business, politics, education, entertainments and others every day through the web. This huge volume of data leads to thinking of building systems that make it easy to analyze people's opinions towards subject. Hence It is necessary to analyze this big data of text to know what people think towards product, policy, services, and others. Sentiment Analysis SA classifies expressions as positive or negative opinions towards the subject of interest after identifying the sentiment expressions, determining their polarity, and relationship to the subject [1]-[10].

Nowadays, researchers take great interest in opinion mining (OM), taking into account the huge available data provided by the Internet and World Wide Web (WWW). Since people tend to be biased when analyzing data according to their personal preferences. Building a system that analyses opinions accurately and in an unbiased manner became a necessity in order to aid decision makers to take the right decisions i.e., sentiment analysis SA.

1.2 Problem Statement

Data size on the web is enormous and grows rapidly. Processing and analyzing this size of data is hard and costly. Therefore, existing solutions of SA are suffering from several problems, such as low accuracy, the selection of an appropriate classification algorithm for a particular data, and methods used to increase classification accuracy.

The process of selecting relevant features and using of ensemble methods is a matter of research to improve classification accuracy.

1.3 Related Work

Many researchers have studied SA, using different methodologies and algorithms using various datasets. H. Zin et al. [1] discussed many of pre-processing strategies (such as removing stop words, meaningless, numbers) that effect the classification performance of the online movie reviews. Support Vector Machine (SVM) achieved high performance results for both features representation, Term Frequency (TF) and Term Frequency-Inverse Document Frequency (TFIDF).

M. Islam and N. Sultana [2] compared the performance of multiple machine learning algorithms for SA, the results showed that the Linear SVM achieved high performance.

P. Kumbhar and M. Mali [3] presented many feature selection techniques (Filter, Wrapper, Embedded) with different classifiers for SA. They concluded that filter methods outperformed others in processing time. Also, wrapper method gives more accurate results. Furthermore, the embedded method that is combination of filter and wrapper reduces the computation time taken up for reclassifying different subsets which is done in wrapper methods.

N. Joshi and S. Srivastava [4] showed using ensemble technique (Bagging) to improve classification accuracy by using different decision trees as the base classifiers.

S. Pant and K. Jain [5] presented a survey about types of sentiment analysis (Document, Sentence, Aspect) and techniques of sentiment classification such as Naïve Bayes and SVM.

V. Sahayak et al. [6] presented an approach which automatically classifies the tweets as positive, negative or neutral with respect to the query term. It uses the pos-tagging and the tree kernel to prevent the need for feature engineering, but the difficulty increases with the complexity.

G. Gautam and D. Yadav [7] studied an approach in which they extracted the adjective from a dataset (labeled tweets) that have some meaning which is called feature vector. Then select the feature vector list and thereafter applied machine learning based classification algorithms namely: Naïve Bayes, maximum entropy (ME) and SVM along with the semantic orientation based wordnet which extracts synonyms and similarity for the content feature. The results showed that the Naïve Bayes technique when subjected to unigram model gives a better result than the ME and SVM. Further the accuracy is again improved when the semantic analysis wordnet is followed up, which raises it from 88.9% to 89.9%.

1.4 Motivation

Since it is necessary to analyze people opinions on various topics, like product quality, services, education, and others. This analyzing is useful for evaluation any tweet, opinion, purpose or reputation for university, company, mobile etc. It should be noted that the achieved classification accuracy is prone to the utilization of different algorithm for a particular data set. Therefore, it is appropriate to identify the most suitable algorithms and approaches for particular datasets type to achieve high accuracy.

This research helps decision makers to evaluate opinion of users via tweets and comments on the social media and networks sites, using appropriate classification algorithms.

1.5 Contribution

The aim of this work is to compare the accuracy and performance of feature selection and ensemble methods using different classification algorithms.

During the comparison of algorithms, we found many differences in the achieved performance and classification accuracy. Precisely, in this work we find answers for the following questions:

Which classification algorithm is more suitable for this data?

Which feature selection method is more suitable for this data and for these classification algorithms?

Which ensemble method will give more classification accuracy?

How tuning parameters affect the result of classification algorithm accuracy, and reduce time needed to select features subset?

Briefly, the answers can be obtained by the following actions:

- Reducing the size of features subset that resulted from reviews processing.
- Using methods for feature selection such as statistical method (filter), machine learning method (wrapper).
- Using improved method for feature selection such as embedded method.
- Using improved method by merge features subsets that resulted from embedded method.
- Using ensemble methods.
- Using genetic algorithm (GA) as search method for features.
- Tuning the parameters of the algorithms.

1.6 Thesis Organization

This thesis is organized as follows:

- Next, Chapter 2 introduces a background and illustrates the concepts and processing steps for text data and SA.
- While Chapter 3 covers literature review regarding the proposed approaches and techniques provided in the literature. Also, it presents the literature and previous research that tackle the SA, used methods for feature selection.
- In Chapter 4 we present the proposed approach and methodology to achieve better accuracy in SA. We present a comparative study among various algorithms to select the best one amongst them to be used in proposed approach.
- Chapter 5 explores the utilized environment, datasets, tools, algorithms and settings for our experiments. Also, it presents the experimental results analysis using Weka tool, explaining Weka tool, making experiment for each feature selection method. Moreover, in each experiment we present evaluation measures (accuracy, precision, recall, f-measure), but mainly we focus on accuracy during comparison.
- Finally, Chapter 6 concludes this work, the obtained results and propose a future work in this subject.

Chapter 2

Background

In this chapter we introduce some concepts and algorithms used in our work. It provides illustration of sentiment analysis (SA), data preprocessing such as tokenization, stop words removal, case normalization. Also, illustration of vector space model, feature selection, feature types, feature weighting, TF-IDF, machine learning algorithms, classification, ensemble methods, clustering, validation, and evaluation.

2.1 Introduction

SA is the use of Machine Learning techniques in Data Mining. It is the process of analyzing opinions and emotions to infer the tendencies shown in the analyzed data, and classify them into positive, negative or sometimes neutral [8].

SA often used in several areas including:

- **Politics**: Governments want to know how voters feel about state policy in running government departments, agreements, and public services [9].
- **Products**: Companies are interested in knowing what the consumers and customers feel about a product or a service, what are their recommendations, and what is their satisfaction level with this product or service [10].
- **Education**: Universities seek to know the opinion of their students and the community regarding the performance and quality of their services; such as their satisfaction with the electronic services provided by the university (e.g., registration portal, e-learning support, etc.) [9].
- **Events**: Mining people's opinions will help governments, and election candidates to plan public policy in countries. There is an imperative need to know people's opinion about events that disturb security, or cause a blow to the country's economy; hence increasing the state's ability to manage crises and achieve a safe life for the people [9].

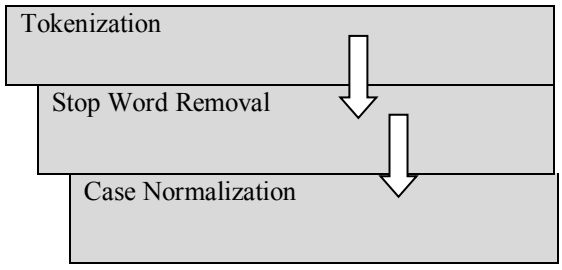
- **Health**: Hospitals are concerned with knowing patients' opinions and their satisfaction with the health services they provide in order to improve these services [11].

There are three levels of SA. They are document, sentence, and aspect level [8][12]. In the document level the classification is carried on to classify the hall document positively or negatively. While sentence level classifies sentences positively or negatively, and aspect level classifies sentiment with respect to meaning of entities. In this research we carried out experiments on document level, in which a document is treated as one piece of information for classification. Thus, the document is classified either positive or negative depending on the feelings and opinions found they contain. The idea behind (SA) is to provide this information by building a system that can classify documents as positive or negative. Therefore, it needs some special data preprocessing.

2.2 Data Preprocessing

Data preprocessing is a critical and time-consuming step in (SA). By this step, we can reduce feature space. Preprocessing techniques mean tokenization, stop word removal, and case normalization [1] [13].

Figure 2.1: Preprocessing Steps



- **Tokenization**: Tokenization is a first step in the data processing. This means split the document into separate words (tokens) using the space character for example the sentence “Al-Quds University in Palestine”, this sentence consists of words: Al-Quds, University, in, Palestine.
- **Stop Words Removal**: Stop Words in (SA) are not necessary, and not effect on the meaning of sentence (e.g., a, an, and, the, but, if, or, etc.). So, these words are removed to reduce features space [14].

- **Case Normalization**: The text is case normalized if all letters of the document are converted into lower case.

2.3 Vector Space Model (VSM)

Vector space model is a model for representing text documents, it is used in document classification, text filtering, and relevant features ranking. Set of documents is represented as vectors is called vector space model. Each dimension in the vectors corresponds to separate terms in the document, the value in the vector will be non-zero if the term appears in the document, else zero if the term doesn't appear in the document. TF-IDF weighting is method to calculate those values of vectors [15][16]. As an example, on VSM:

Table 2.1: Dataset Example

Document	**Review**
Doc A	This Movie is Beautiful
Doc B	This Movie is Bad
Doc C	This Clip in This Movie is Bad

Table 2.2: VSM Representation

Document	**This**	**Clip**	**in**	**Movie**	**is**	**Bad**	**Beautiful**
Doc A	1			1	1		1
Doc B	1			1	1	1	
Doc C	2	1	1	1	1	1	
Document Frequency	3	1	1	3	3	2	1

2.4 Feature Selection

Process of extracting features from unstructured text is important to SA. Feature is relevant if its existence improves the classification performance and accuracy. On the other hand, a feature is irrelevant if its existence decreases the classification performance. Thus, it is important to know the right way to extract features [17]. To recognize features as relevant, irrelevant, or redundant we need to calculate Entropy and Information Gain (IG) of features to identify the ranks/weights of the features and to decide which feature has max IG [18]. In another words, relevant features are considered as the features with relatively high IG, while the features with very low IG can be considered as irrelevant features (i.e., the higher the IG of a feature the higher the relation and impact on the classification accuracy). Information Gain is a measure that evaluate the strength of a feature importance. IG of a feature is calculated based on the Entropy of that feature, where the Entropy measures how informative the feature (as a random variable) is averaged on all its possible outcomes. Information Gain, and Entropy can be calculated by the following formula:

$\boldsymbol{Entropy(ent)}$ $(\boldsymbol{s})$=$\sum_{i=1}^{c} - p_i \log_2 p_i$, where p_i is the probability of choosing class i. (2.1)

$\boldsymbol{Info\ Gain}$ $(\boldsymbol{S.Fj})$=$\boldsymbol{Entropy(s)}$ - $\sum_{Vi\varepsilon VFj} \frac{|Svi|}{|S|}$. Entropy(Svi). (2.2)

Where VF_j represents the whole amounts of attribute (F_j), (Sv_i) is a subset of (S) about that feature (F_j) has value (Vi) [19].

2.4.1 Feature Types

An n-gram (i.e., unigrams, bigrams, trigrams, etc.) is a sequence of n contiguous terms of text. These terms can be letters, words, phonemes, and syllables. N-gram is a group of words that occur in a specific frame, in another words: n=1 is unigram, n=2 is bigram, n=3 is trigram and so on [20]. When each single word is taken separately, this is called a Unigram. As an example, suppose we have sentence “I have a beautiful university”. In case of unigram the sentence is split as: “I”, “have”, “a”, “beautiful”, “university”. In case of bigram the sentence is split as: “I have”, “have a”, “a beautiful”, “beautiful university”, and so on.

2.4.2 Feature Weighting

Feature weighting is a necessary technique to find optimal weights of features to optimize accuracy of classification.

Feature weighting can be considered as a generalization of feature selection. In feature selection, feature weights a helpful measure to decide whether the feature to be used or not. Feature weighting by assigning each a continuous valued weight allows finer differentiation between features. Features can be weighted by many methods such as statistical and machine learning method [21][22].

- **Weight by Machine Learning**: Using classifier in machine learning we can calculate weights of features with respect to the class, to figure out how relevant a feature when building model to have good predictions and high accuracy of classification. By machine learning we train a model using subset of features. Then add or remove features from subset depending on results we have from previous model. The selected subsets of features from a dataset and the ranking of features is depending on the used machine learning classifier [22].
- **Weight by Correlation:** Correlation is one of statistical techniques by which features can be assigned weights with respect to a class. This weighting scheme is based upon correlation and it returns the absolute or squared value of correlation as feature weight [23]. Thereafter, upon the calculated weight the features with N top ranks are selected to be in the feature subset.

  Correlation evaluates the worth of a feature by measuring the correlation (Pearson's) between it and the class, it gives ranking of the features from higher to lower, it gives the result weight of features without support of any machine learning algorithm like J48, Naive Bayes (NB), SVM and others. Pearson correlation is the most used correlation statistic to measure the strength and relationship between linearly related features [23]. The following is a statistical formula used to calculate Pearson r correlation:

$$r_{xy} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\sqrt{n\sum y_i^2 - (\sum y_i)^2}} \quad (2.3)$$

r_{xy} = Pearson r correlation coefficient between x and y

n = number of observations (number of pairs of scores)

x_i = value of x (for i^{th} observation)

y_i = value of y (for i^{th} observation)

- **Term Frequency-Inverse Document Frequency (TF-IDF)**

Term frequency (TF) is the weight of term in a document, it assumes that more occurrences of a term in a short document gives it higher and more significance weight. Document Frequency (DF) represents the number of documents in the dataset in which the term occurs, it assumes that the more documents the term occurs in the more significant it is.

Term Frequency-Inverse Document Frequency (TF-IDF) assumes that less frequent terms in the dataset are more significant in the document and vice versa.

TF–IDF, it means term frequency–inverse document frequency, to show how important a word is in a document in a corpus, if the number of times a word increases in the document then the value of TF-IDF will increases proportionally, but is offset by the frequency of the word in the collection of documents.

The word is more important if it appears more frequently in a document while the word itself loses its importance if it appears more frequently in the corpus (set of documents).

TF term = (times terms appears in the document)/ (total number of terms in the document).

TF term = [1+log_e (times terms appears in the document)]. (2.4)

Inverse Document Frequency: It checks how the term is important across all documents, the term appears among or rare in all documents in the corpus.

IDF term = log_e (total number of documents/Number of documents the term appears in). (2.5)

The TF-IDF weight of a term is the product of its TF weight and its IDF weight

TF–IDF = (TF.IDF) [15][24]. (2.6)

2.5 Machine Learning Algorithms

One of the primary tasks of machine learning is to extract valuable information from training data and then use it to build a model that is able to predict the shape of new data. If a computer program can execute certain tasks by drawing on previous experience, we can say that it has been learned. Where design and development of machine learning algorithms and technologies enable the computer to possess the learning feature.

Machine learning is classified into supervised, unsupervised, semi-supervised, reinforcement, transduction, and learning to learn [25] as shown below:

- **Supervised Learning**: The system learns its' function that maps inputs to outputs by inferring from examples of input-output pairs called labeled training data which is user fed to the system. In this work we will concentrate on this type of machine learning algorithms.
- **Unsupervised Learning**: Is a type of machine learning that detects patterns in a data set with no pre-existing labels. The system attempts to deduce the structure of unlabeled data, leaving the learning algorithm to rely on itself to explore its input structure. In this type the algorithm discovers hidden patterns in the unlabeled data, it attempts to extrapolate relationships by extracting features and patterns on its own. It works by trying to find useful relation between the elements of the target set.
- **Semi-supervised Learning**: The system uses both labeled and unlabeled data to construct the prediction model.
- **Reinforcement learning**: Is a machine learning area concerned with taking actions by software agents in an environment while maximizing the cumulative reward. In the absence of training data, the algorithm learns by experience. It collects training examples and knows what is a good procedure and what is a bad procedure through taking actions in an environment, which is interpreted into a reward and a representation of state, which are fed back to the system.
- **Transduction**: tries to predict new outputs based on training inputs, training outputs, and new inputs.
- **Learning to learn**: The algorithm learns its own inductive bias based on previous experience.

2.5.1 Classification

By classification we mean a form of data analysis that builds a model that describes important data categories. It is called classifiers that predicate the class label of unknown records and categorizes the feature in one of several predefined categories. In this section we introduce some classification algorithms such as Bayes, SVM, Decision tree. Also, using ensemble methods with these algorithms. There are vast number of supervised machine learning approaches and algorithms in the literature. We introduce the most popular and well-known classification algorithms are described below:

- **Bayes (Naïve Bayes Multinomial, Naïve Bayes)**: are family of classifiers based on Bayes probability theorem, these algorithms are mostly used in (SA). Naïve Bayes refers to conditional independence of each of the features in the model, while Multinomial Naïve Bayes uses a Multinomial distribution for each of the features. Naïve Bayes loses accuracy because independence of each of the features in the model [26]-[32].

$$\boldsymbol{P(y|X) = \frac{P(X|y)P(y)}{P(X)}} \tag{2.7}$$

  Where the class variable y is positive or negative value, X represent the features, P(y|X) is conditional probability. The probability that event y occurs given that event X occurs.

- **SVM Classifier (LibLINEAR, LibSVM, SMO)**: The idea of support vector machine is to put line(hyperplane) between the two classes, finding a hyperplane that best divides a dataset into two classes, trying to drive the widest channel between the two classes [33][34] as shown in Figure 2.2.

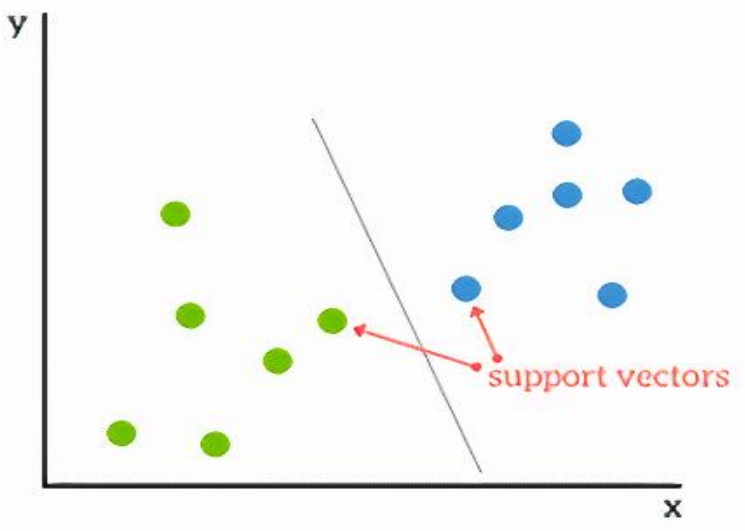


Figure 2.2: Hyperplane in Support Vector Machine

- **Decision Tree**: are a type of Supervised Machine Learning to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. Some different decision tree algorithms are listed below:
    - **J48**: It is an open source Java implementation of the C4.5 algorithm developed by the WEKA project team, to predict the target variable of a new dataset record [35].
    - **REPTree**: This Algorithm is a fast decision tree learner; it is also based on C4.5 algorithm and can produce classification (discrete outcome) or regression trees (continuous outcome). This early random decision tree method combines bagging and random feature selection methods to generate multiple classifiers [36].
    - **Decision Stump**: is a machine learning model consisting of a one-level decision tree [37].
    - **Hoeffding tree**: It is an incremental, anytime decision tree induction algorithm that is capable of learning from massive data streams [36][38]. It uses the Hoeffding bound for construction and analysis of the decision tree. Hoeffding bounds are used to decide the number of instances to be run in order to achieve a certain level of confidence.
    - **Random Tree**: is a supervised learner, it is an ensemble learning (EL) algorithm that generates lots of individual learners. It employs a bagging idea to construct a random set of data for constructing a decision tree [39].

- **Logistic Model Tree (LMT)**: is a classification model with an associated supervised training algorithm that combines logistic regression (LR) and decision tree learning [36].

2.5.2 Ensemble Methods

When analyzing big data, a single classifier may not give high accuracy, but combined classifiers (ensemble methods) may produce high accuracy. Ensemble method helps to reduce noise, variance that causes error in learning [40].

The aim of utilizing ensemble learning (EL) in this work is to improve classification performance. This can be achieved by having multiple machine learning algorithms, which use multiple trained models, by combining the output of these models we can get low bias and low variance. The result of ensemble is improving classification accuracy and flexible model. Some of these ensemble methods are:

- **Bagging**: Bagging or Bootstrap Aggregation is an effective ensemble method. It is desired with learners have high variance (unstable learner), this method generate several training data sets by random sampling. Each of these data sets is used to train a different model. The outputs of the models are combined by averaging or voting (i.e., the result of majority) to create a single output. These models are built by parallel in bagging method as shown below in Figure 2.3 [40].

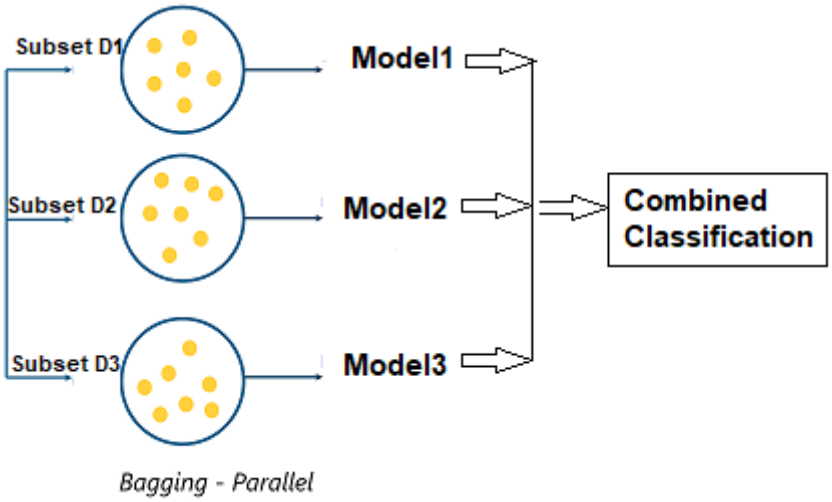


Figure 2.3: Ensemble Using Bagging Method

- **Boosting**: is similar to bagging, several training data sets are generated by random sampling. Each of these data sets is used to train a different model. These models are processed sequentially. In boosting weights are assigned to each model and the output is obtained by weighted average of models, as shown in Figure 2.4 [40].

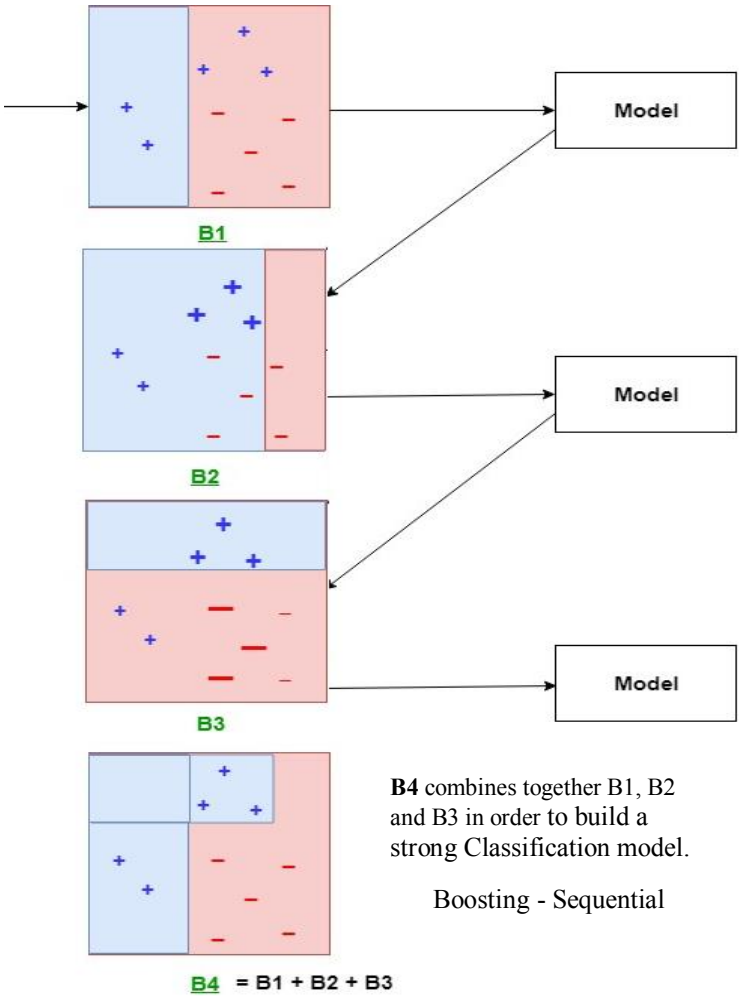


Figure 2.4: Ensemble Using Boosting Method

- **Stacking**: Stacking is used to combine different classifiers. It consists of two stages i.e., base learner and meta learner. In base learner many different models are used to learn from a dataset. Consequently, new dataset is created by collecting outputs of each model. Then the resulting dataset is used by stacking model learner meta to result the final output, as shown in Figure 2.5 [40].

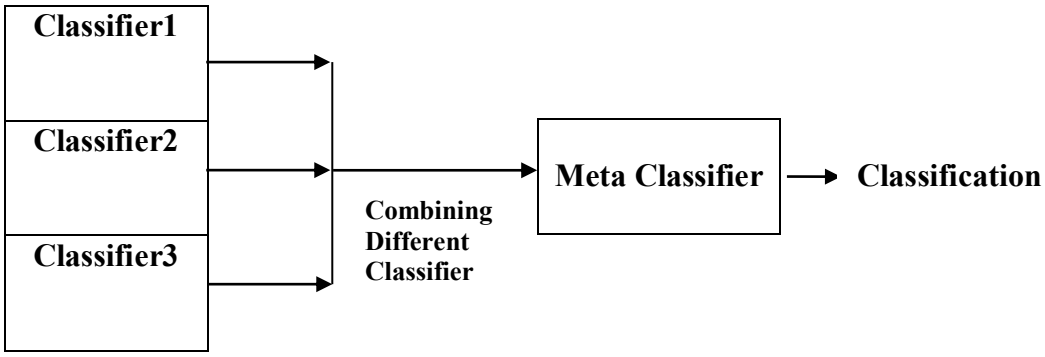


Figure 2.5: Ensemble Using Stacking Method

- **Vote**: is similar to stacking, but vote is used to combine different classifiers without learner meta, as shown in Figure 2.6 [40].

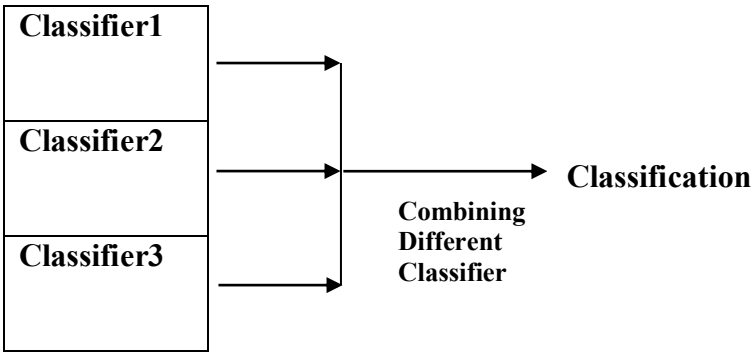


Figure 2.6: Ensemble Using Vote Method

2.5.3 Clustering

Data clustering is the process of placing data in similar clusters. It is a branch of data mining. Clustering belongs to the type of unsupervised learning. Its purpose is to partition unlabeled data into clusters. The clustering algorithm divides a dataset into several clusters, where the similarity between the points within a given cluster is greater than the similarity between two points within two different clusters. We deal with a large amount of data. We tend to summarize the huge amount of data into a small number of groups or categories in order to facilitate the analysis process. There are many algorithms used in the data clustering process, such as the k-means clustering algorithm [41].

2.6 Validation and Evaluation

2.6.1 Cross-validation

The purpose of cross-validation to make the performance evaluation unbiased by fitting and evaluating each candidate model on separate data sets. K-fold cross-validation is the method to evaluate classification. In this method the data set is randomly divided into k subsets of approximately equal sizes and the model generated from k-1 folds is tested against the remaining fold in turn. The performance of the model is the average of the k accuracies that resulted from the k-fold cross validation [42]. In this research we are using 10-fold cross validation.

2.6.2 Evaluation Measures

Confusion Matrix see Table 2.3 [43] is a performance measurement for machine learning classification. The value of precision and recall are based on computing confusion matrix as follows:

Table 2.3: Confusion Matrix

	Predicted (P)	Predicted (N)
Actual (P)	True Positive	False Negative
Actual (N)	False Positive	True Negative

The following are some definitions that we need in classification evaluation:
True-Positive (TP) is the number of positive reviews correctly classified as positive. True-Negative (TN) is the number of negative reviews correctly classified as negative. False-Positive (FP) is the number of negative reviews that were misclassified as positive. False-Negative (FN) is the number of positive reviews that were misclassified as negative.

There are different evaluation measures used to evaluate the classification performance [44]. In this work we focus on some of these measures namely:

- **Accuracy**: Accuracy (a) as a measure is the number of reviews that are correctly classified. It is one of the most important measures of classification, which is defined as a ratio of the correctly/truly classified reviews (both positive and negative) to the total number of reviews [44]:

  $a = (TP+TN)/(\mathrm{TP+TN+FP+FN})$. (2.8)

  It measures the percentage of correct predictions taking into account positive and negative inputs. It relies on the distribution of a dataset that can easily leads to wrong conclusions about system performance.

- **Precision**: Precision (p) is the ratio between correctly classified reviews (as true positive) and all classified reviews (true positive and false positive) [44]:

  $p = TP/(T\mathrm{P+FP})$. (2.9)

- **Recall**: Recall (r) is the ratio of the correctly classified positive reviews to all positive reviews [44]:

  $r = TP/(\mathrm{TP}+FN)$. (2.10)

- **F-Measure**: F-measure is the harmonic mean of precision and recall which defined as [44]:

  $F-Measure = (2*p*r)/(p+r)$. (2.11)

  This measure is used in performance evaluation of text classification.

2.7 Summary

In this chapter, we explained concepts about (SA), data preprocessing, VSM, feature selection, features types, feature weighting, TF-IDF, machine learning algorithms, classification, ensemble methods, clustering, validation, and evaluation.

Chapter 3

Literature Review

This chapter presents some previous works and researches relevant to our work. It introduces works on preprocessing algorithms, feature selection in sentiment analysis. Also, it presents statistical, machine learning, embedded methods, classification algorithms and comparison of these algorithms.

3.1 Introduction

There are many researches in (SA) in order to aid decision makers, take the right decisions, the approaches of this type of research focus on classifying text as positive or negative. In this chapter we review some related works. It presents some approaches and methods provided in the literature. Mainly those approaches that introduce, illustrate and explores how to select features of dataset, and which method are utilized to select features such as statistical, machine learning and hybridized methods.

3.2 Pre-processing

H. Zin et al. [1] introduced the effect of the pre-processing strategies in the (SA) of online movie reviews. Because data collected from the Web is usually full of errors and unnecessary words, this information needs to be pre-processed, including several steps: data tokenization, and removing stop-word, meaningless words, numbers, and word less than 3 characters, after pre-processing the authors found improving of classification accuracy.

S. Kotsiantis et al. [13] presented the most well know algorithms for each step of data pre-processing (cleaning, normalization, transformation, feature extraction and selection).

3.3 Feature Selection in Sentiment Analysis

S. Gnanambal et al. [23] presented a comparative evaluation study of classification algorithms before and after attribute selection using (WEKA), authors showed classification performance will improve after attribute selection.

V. Vaghela and B. Jadav [45] presented classification approaches (using lexicon-based approach and machine learning based approach), and feature selection methods (using TF-IDF, Information Gain, Chi-square) that give better accuracy to classification of (SA).

3.4 Feature Selection Methods

P. Kumbhar and M. Mali [3] introduced the important challenge of the (SA) of text classification is the accuracy and the high dimensions of the feature space. These issues can be overcome by using the feature selection. Feature selection is a process to select a subset of the best features from the original features set. Feature selection is a strategy designed to make text document more efficient and accurate. There are several methods of feature selection: statistic, machine learning and embedded methods, by these methods we can reduce time, and improve classification accuracy.

B. Agarwal and N. Mittal [46] used feature selection methods to extract best features Information Gain (IG) and Minimum Redundancy Maximum Relevancy (mRMR), using unigram and bi-grams feature set to extract features, using classification algorithms Boolean Multinomial Naïve Bayes (BMNB) and (SVM). The result of experiment was BMNB classifier performs better than of SVM classifier.

M. Doshi and S. Chaturvedi [47] used CHI-SQUARE, INFOGAIN and GAINRATIO to select relevant features. Results showed an increase in accuracy and a decrease in time.

H. Arafat et al. [48] used feature selection methods and showed that mRMR is performs better than IG for sentiment classification. Also, they claimed that hybrid feature selection method based on the Rough set theory and Information Gain (IG) is performs better than the previous methods.

T. Phyu and N. Oo [49] used feature subset selection algorithm based on conditional mutual information (MI), this approach was proposed to select an optimal feature subset.

A. Manek et al. [50] used feature selection method based on Gini Index with Support Vector Machine (SVM) classifier, this approach was proposed for sentiment classification of a large movie review data set.

H. Hamidi and A. Daraei [51] applied a hybrid feature selection approach. First step they selected feature-based weight by SVM, then they applied GA on the selected features, after that classification algorithms were applied to predict the occurrence of Myocardial

Infarction. Their results showed that the Multi-layer Perceptron and Sequential Minimal Optimization achieved better accuracy.

D. Gamal et al. [52] used Arabic dataset for (SA), they applied many steps in the preprocessing stage, then different machine learning algorithms were applied, the results showed that ridge regression algorithm has the highest accuracy.

3.4.1 Statistical (Filter), Machine Learning (Wrapper), Embedded Methods

P. Kumbhar and M. Mali [3] introduced many feature selection methods that are categorized into three types:

- **Statistical Method (Filter)**: The features that has high ranking will be selected, the process of selection is independent to classifiers, by this method no dependency between features, time is needed to select features are low, (such as statistical method Correlation criteria, Chi-square test, Information Gain).
- **Machine Learning Method (Wrapper)**: This method needs learning algorithm to select the relevant feature, it needs search method to find subset of features, time is needed to find subset are high, examples of these search method are genetic, heuristic, grid search method.
- **Embedded method**: The process of feature selection is performed by a classifier as part of training process, it is called hybrid model of filter and wrapper. In the hybrid method, the filter generates subsets, while wrapper evaluates these subsets to find a feature subset.

3.5 Classification Algorithms Comparing

X. Wu et al. [53] presented a description and comparison between 10 top algorithms, C4.5, k-Means, SVM, Apriori, EM, PageRank, AdaBoost, KNN, NB, and CART. The authors used these algorithms in data mining for classification, clustering, and statistical analysis.

The authors made comparative study on machine learning algorithms for sentiment classification, they presented (SA) for customers on two datasets, using six different machine learning algorithms Naïve Bayes (Multinomial), Logistic Regression, SGD (Stochastic Gradient Descent), Linear SVM (Support Vector Machine) and RF (Random Forest), they compare the differences between these algorithms according accuracy, and showed the result Linear SVM is the best for sentiment classification .

P. Khare and K. Burse [54] showed the performance of different classification methods on clinical data, using five classification algorithms are Bayesnet, SMO, Simple Logistic, ONE-R, ZERO- R. The result was Bayesnet classifier is the best.

3.6 Genetic Algorithm (GA)

P. Khare and K. Burse [54] used (GA) to select relevant features before applying classification algorithm. M. Govindarajan [55] used the hybrid NB-GA in (SA) of movie reviews, and showed high percentage of classification accuracy, the idea of the algorithm is based on Darwin's theory of evolution and its natural selection process.

3.7 Ensemble Methods

N. Joshi and S. Srivastava [4] presented improving accuracy by using bagging with different classification algorithm decision tree (BF Tree, J48, Decision stump, CART). P. Pujari and J. Gupta [56] presented improving classification accuracy by selecting more important feature and by combining multiple classifiers three decision tree classifiers CART, CHAID and QUEST.

I. Syarif et al. [57] presented ensemble algorithms (Bagging, Boosting and Stacking) to improve the performance of network intrusion detection systems. Output of several weak classifier is combined into single composite classification to give high accuracy instead of individual weak classifier.

Chapter 4

The Proposed Approach

This chapter introduces the concepts and our methodology used in this work. Also, it explores our approach used to evaluate the performance for sentiment analysis (SA) methods and algorithms. Moreover, it presents the utilized experiments environment and settings for this work.

4.1 Introduction

Recalling that web sites are widely used and help in connecting people through social networks that provides commenting on public posted material. Often, people express their opinions on various topics. The availability of these information raises the interest of many institutions, organizations and businesses of these information. Hence it is necessary to process and analyze this big data of text to have knowledge about the opinions of people about product, policy, services, and others issues. SA classifies expressions as positive or negative opinions towards the subject of interest after identifying the sentiment expressions, determining their polarity, and relationship to the subject [10].

Researchers developed approaches for feature selection in SA, each approach has advantages and disadvantages, we develop approach that is based on improved methods such as feature selection and ensemble methods for SA classification using different classification algorithms, merge features subsets to improve accuracy, and tuning parameter of classification algorithm to improve accuracy and reducing time needed to select features subset.

4.2 Feature Types

In this work we used unigram feature type as one of n-gram feature types (see section 2.5), each single word is considered as a feature as words are separated by space. In this research we experimented based on type of feature is unigram as many researchers used this type and got in their experiments high accuracy using unigram features better than bigram features [58][59].

4.3 Preprocessing

The preprocessing stage is necessary to clean data, reduce dimensional space of data. This stage includes:

- Tokenization: breaking up text of each review into words using space as separate between words.
- Stop-words removal: removing the English language most common words using a list of stop words, such as “the”, “a”, “an”, “in” etc.
- Case Normalization: Converting words into lowercase letters (i.e., replacing uppercase letter by lowercase one in the words).
- Using TF-IDF: Researchers use weighting techniques to select features such as Term Frequency (TF), which means the term has more occurrences in a document indicates that this term is important. Document Frequency (DF) which means the term has more occurrences in more documents indicates that term is important. Inverse Document Frequency (IDF) used is the opposite of DF where the term is more relevant if it occurs in less documents. Term Frequency-Inverse Document Frequency (TF-IDF) which means term loses its importance if it appears more frequently in the corpus (set of documents). Entropy defines relevant terms as terms that have high occurrence frequency in less documents. We used in this research Tokenization, Stop-words removal, TF-IDF.

4.4 Classification

Machine learning-based (SA) go through multiple phases that are: preprocessing, feature selection, machine learning algorithm as shown in Figure 4.1. Sentiment classification is usually formulated as a two-class classification problem, positive and negative, training and testing data used are normally product reviews.

The dataset is split into training and testing sets using sampling and 10-fold cross validation, and for comparing between classification algorithms. In this work we used many machine learning algorithms are: Bayes (Naïve Bayes Multinomial, Naïve Bayes), SVM (LibLINEAR, LibSVM, SMO), Decision Tree (J48, REPTree, DecisionStump, HoeffdingTree, RandomTree, LMT), and using ensemble methods (Bagging, Boosting, Stacking, Vote).

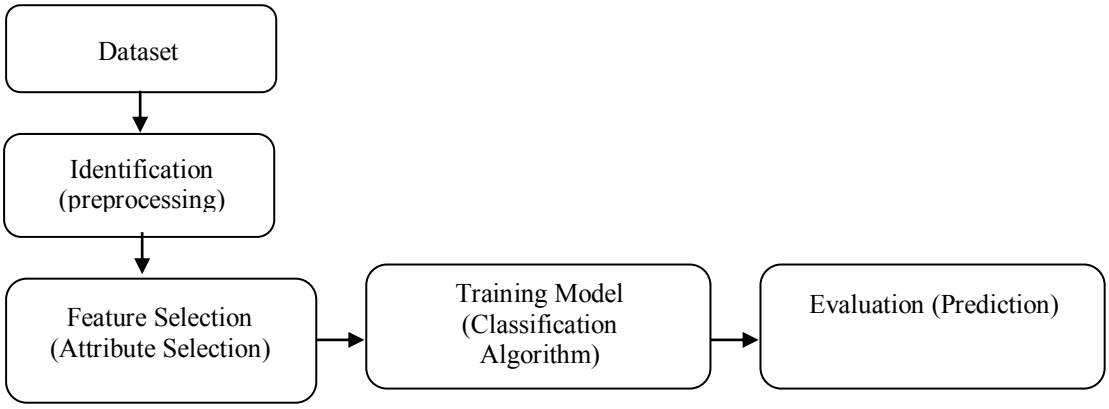


Figure 4.1: Multiple Phases in Sentiment Analysis

4.5 Search Method of Feature Subset

Using in this research Genetic Algorithm (GA) as search strategy of a subset of features, the main concept of these algorithm is the survival of the fittest.

The idea of the algorithm is based on Darwin's theory of evolution and its natural selection process. The principle of algorithm (survival of the fittest or strongest), where the solution group is the feature group, the quality of the solution (fitness function) is a measure of the strength or weakness of a feature. Components of the (GA) are: reproduction, crossover, and mutation. Where reproduction selects good subset of features, crossover combines good features to generate better offspring's, and mutation change feature locally.

Mating is a process that we will create to share parts of solutions and will be called crossover. The chance of living a feature as a result of luck will make it in multiple ways to choose solutions from generation to generation and call it selection. Further, we add the mutation part as it can cause a change in a gene to solve to produce a better solution called mutation. The selection process that depends on the fitness function, and on the parameters of the (GA) continues to generate successive genes until the optimal solution of features is found [60].

4.6 Ensemble Learning (EL)

We select subsets of features by different methods, combine different classification algorithms and using different ensemble methods that can be applied on these different subsets of features. So, we evaluate ways to use different ensemble methods (Bagging, Boosting, Stacking, Vote) in our experiments by comparing the results of ensemble methods (learners) with other results using single learner. There is no learner is absolutely

better than the other learner, but combining the learners with different (ensemble methods) may work well in different parts of the data, and it is possible to get very strong learner because by applying ensemble methods we get combining output of these methods that lead to models with low bias error and low variance error [4][40].

4.7 Research Methodology

This section presents our methodology and steps in carrying out this work as shown below in Figure 4.2. It includes preprocessing, feature selection methods, tuning parameter, comparison is carried out amongst different classification learning algorithms with using different ensemble methods, methodology of this work are as follows:

1. **Preprocessing**: By removing noise i.e. redundant features, irrelevant features, numbers, stop word, missing value. Then, convert uppercase letters to lowercase letters. This can be accomplished using TF-IDF to know frequency of terms in document and in corpus.
2. **Feature Selection Methods:** We select features subset by many methods (statistical, machine learning, embedded), we introduce approach to select features in the following steps:
    - First, using statistical method (Correlation) to measure weight, rank of feature, and correlation between feature and class, to obtain subset of features that have high rank and weight.
    - Second, using machine learning method (Wrapper) with genetic search to select features.
    - Third, using embedded method to improve features selection from previous features subset that we obtained in first, and second steps.
    - Fourth, we merge features subsets that we obtained in third step.
    - Fifth, Tuning parameter of feature selection method.

3. **Comparative study:** In this phase a comparative study on performance evaluation of machine learning algorithms is carried out with using ensemble methods (Bagging, Boosting, Stacking, Vote) for the following algorithms: Bayes (Naïve Bayes Multinomial, Naïve Bayes), SVM (LibLINEAR, LibSVM, SMO), Decision Tree (J48, REPTree, DecisionStump, HoeffdingTree, RandomTree, LMT).

4.7.1 Research Methodology Diagram

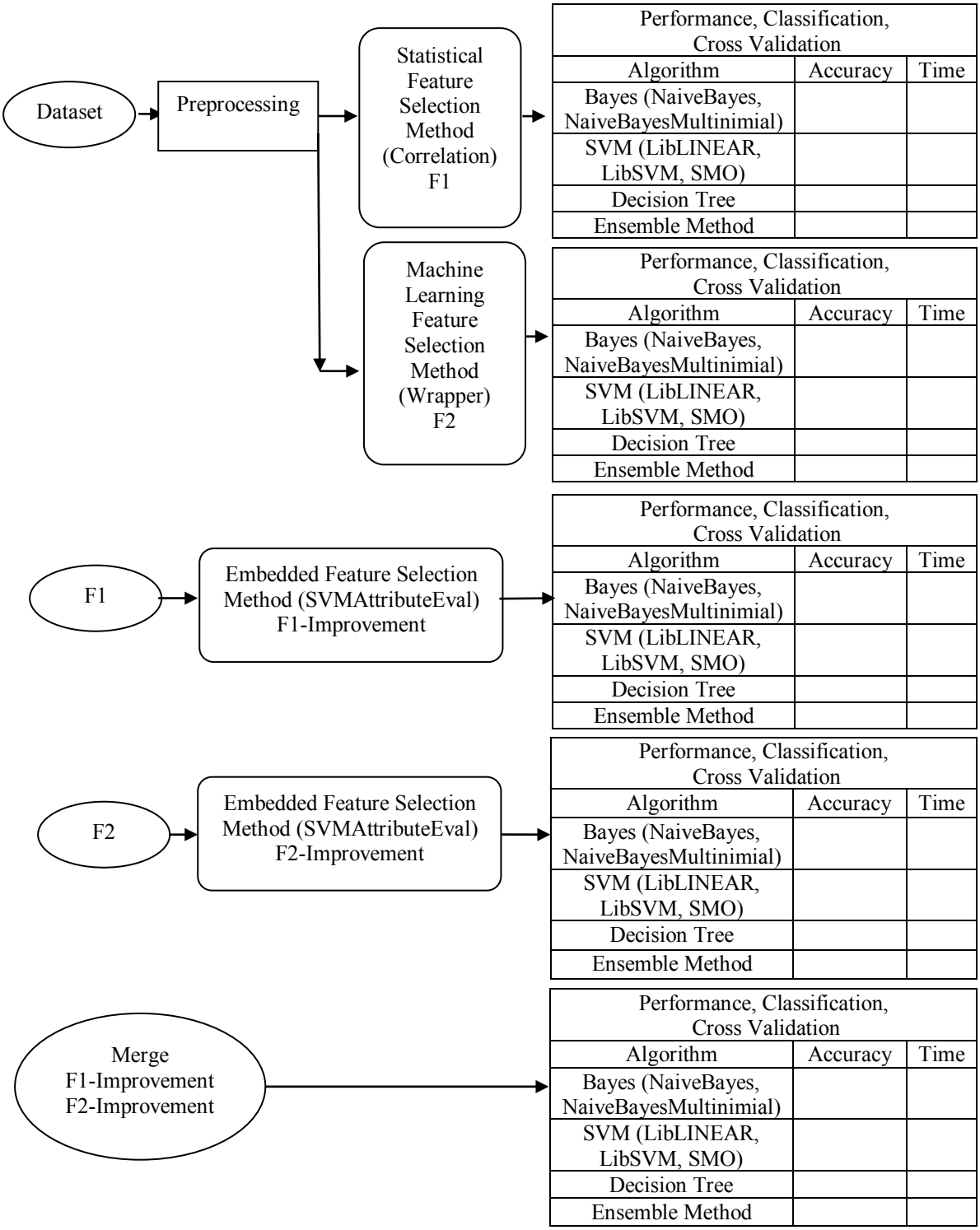


Figure 4.2: Research Methodology of This Work

Chapter 5

Experimental Results and Discussion

This chapter introduces the utilized experiments' environment and settings for this work. Moreover, it presents the experimental results and discussion. The experiments show classification accuracy of compared feature selection and ensemble methods using different classification algorithms. In our experiments we applied methods to select features subset using statistical method (i.e., Correlation), machine learning (i.e., Wrapper), improved embedded method, and improvement by merge features subsets. Furthermore, we select machine learning algorithms to be used in the comparison such as Bayes (NaiveBayes, NaiveBayesMultinimial), SVM (LibLINEAR, LibSVM, SMO), Decision Tree (J48, REPTree, DecisionStump, HoeffdingTree, RandomTree, LMT), and using ensemble methods (Bagging, Boosting, Stacking, Vote) with these algorithms.

5.1 Hardware and Software Specifications

In the course of our experiments the following hardware and software are used. The computation hardware is a laptop with Intel® Core™ i7-8550U CPU @ 1.80GHz 1.99 GHz and with installed memory (RAM) 8.00 GB. On the other hand, we utilize Weka [1] software tool version 3.9.3 on MS Windows 10 Pro 64-bit operating system).

5.2 Research Scope and Dataset

The scope of this work is to analyze people's opinions and their sentiment towards movies. More specifically, we select a Dataset that contains people's reviews and comments in English. The utilized Dataset (Movie Review Data) from Cornell university.

The Polarity dataset v2.0 (3.0Mb)[2] contains 1000 positive and 1000 negative processed reviews. This Dataset was introduced in Pang/Lee ACL 2004. Released June 2004.

[1] http://www.cs.waikato.ac.nz/ml/weka/

[2] http://www.cs.cornell.edu/people/pabo/movie-review-data/review_polarity.tar.gz

5.3 Methods for Feature Selection

This section introduces the compared classification algorithms in this work.

5.3.1 Statistical Method (Correlation)

Using statistical method in Weka tool [61] is called CorrelationAttributeEval, by Correlation we find usefulness of each feature for the classification process. The features are relevant if they have low correlation with each other and high correlation to the class label. On the other hand, the features are irrelevant if they have low correlation to class label. In this stage, by experiments we found feature subset of 3500 feature with best ranking that give high accuracy when running classification algorithms as shown in the Table 5.1. This table shows the Precision, Recall, F-measure, and Training Time, but we focus on the accuracy and training.

Table 5.1: Accuracy, Precision, Recall, F-measure and Time for Statistical Method (Correlation).

Classification Algorithm			Accuracy	Precision	Recall	F-Measure	Time
Naïve BayesMultinomial			95.7	.957	.957	.957	00:00:01
Naïve Bayes			86.75	.869	.868	.867	00:00:23
LibLINEAR			93.15	.932	.932	.931	00:00:02
LibSVM			93.05	.931	.931	.930	00:00:33
SMO			91.85	.919	.919	.918	00:00:29
Decision Tree	J48		68	.680	.680	.680	00:09:43
	REPTree		67.6	.677	.676	.676	00:03:51
	DecisionStump		62.45	.641	.625	.613	00:00:20
	HoeffdingTree		72.3	.814	.723	.701	00:00:54
	RandomTree		61.45	.615	.615	.614	00:00:04
	LMT		83.4	.834	.834	.834	02:44:25
Ensemble Method	Bagging	Bagging with J48	77.1	.771	.771	.771	01:43:42
		Bagging with REPTree	74.3	.743	.743	.743	00:30:02
		Bagging with RandomTree	72.65	.736	.727	.724	00:00:26
		Bagging with Naïve Bayes	89.3	.894	.893	.893	00:04:41
	Boosting	AdaBoost with J48	77.25	.773	.773	.772	01:57:23
		AdaBoost with REPTree	73.6	.736	.736	.736	00:42:36
		AdaBoost with RandomTree	63.7	.637	.637	.637	00:00:04
		AdaBoost with Naïve Bayes	86.1	.862	.861	.861	00:13:12
	Stacking	Stacking with (j48, HoeffdingTree) and REPTree meta Classifier	81.55	.845	.816	.812	02:16:16
		Stacking with (HoeffdingTree, DecisionStump, j48) and REPTree meta Classifier	81.1	.833	.811	.808	01:47:39

Classification Algorithm			Accuracy	Precision	Recall	F-Measure	Time
		Stacking with (Naïve Bayes, Naïve BayesMultinomial, SMO) and LMT meta Classifier	95.8	.958	.958	.958	00:13:36
		Stacking with (LibLINEAR, Naïve BayesMultinomial, SMO) and LMT meta Classifier	96	.960	.960	.960	00:05:17
	Vote	Vote with (LibLINEAR, Naïve BayesMultinomial, SMO, LMT)	93.45	.935	.935	.934	01:43:42

As a result, we found that: The highest accuracy that can be obtained by Bayes classification algorithms is by using Naïve BayesMultinomial. Also, the highest accuracy of SVM classification algorithms can be obtained by using LibLINEAR. Furthermore, the high accuracy of decision tree classification algorithms is by using LMT, but this algorithm takes more training time. Moreover, we found that bagging with these algorithms gives better accuracy than the accuracy of original algorithm without any bagging, but this takes more time too. Also, boosting with algorithms give accuracy better than accuracy of original algorithm without any boosting. Finaly, stacking with algorithms give accuracy better than accuracy of the original algorithm as Stacking with (LibLINEAR, Naïve BayesMultinomial, SMO) and LMT meta classifier give high accuracy. Vote comparing to stacking with the same classification algorithm, stacking give better accuracy because of the existence of meta classifier in stacking. Also, the needed time in stacking is better than vote (i.e., less time).

Precisely, we obtained high accuracy (96%) by using Stacking with (LibLINEAR, Naïve BayesMultinomial, SMO) and LMT meta classifier. The time needed is 00:05:17 (hrs: mins: secs). The least time (00:00:01) needed is when using Naïve BayesMultinomial with accuracy is 95.7%.

Figure 5.1 and Figure 5.2 depict the obtained results of this experiment. Figure 5.1 shows the accuracy obtained by machine learning classification algorithms using correlation statistical method for feature selection. While Figure 5.2 shows the training time for these algorithms.

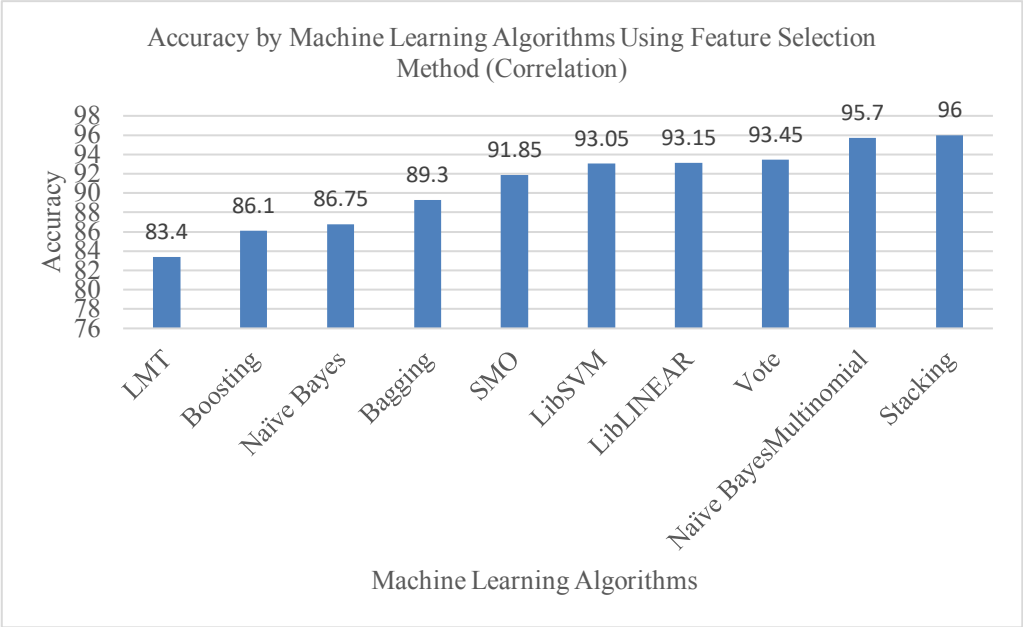


Figure 5.1: Accuracy by Machine Learning Algorithms Using Feature Selection Method (Correlation)

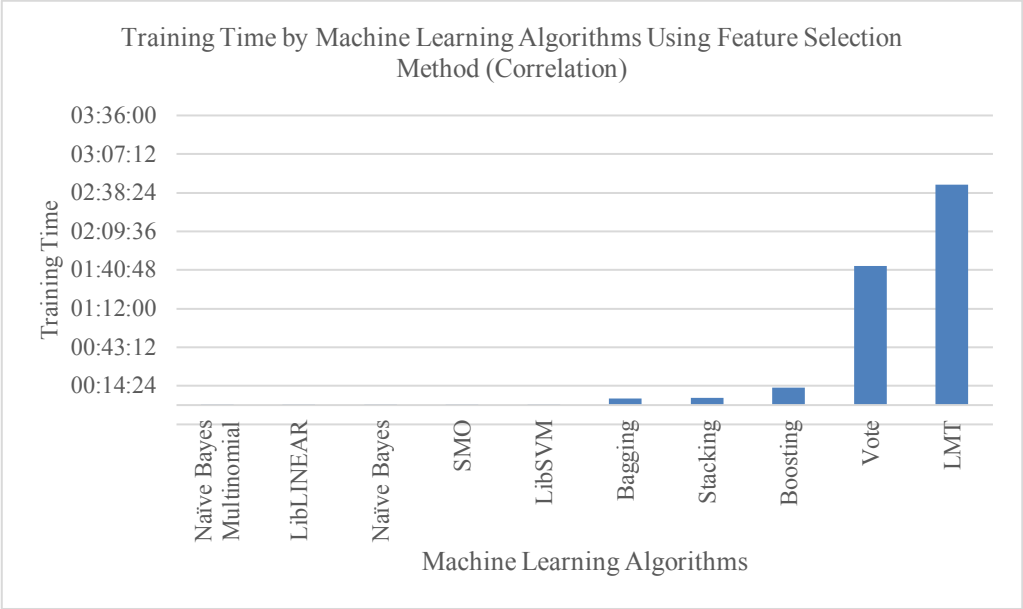


Figure 5.2: Training Time by Machine Learning Algorithms Using Feature Selection Method (Correlation)

5.3.2 Machine Learning Method (Wrapper)

In this subsection we present the used machine learning method and the search method in Weka tool. The used machine learning method in Weka tool is called WrapperSubsetEval and the search method is genetic method. In this case we use classifier to select features by search method. Feature subsets differ according to the used type of classifier to select features. Thus, we select one classifier from well-known classifiers to select features for each classification algorithm which produces best features subset and higher accuracy. The

set of classifiers are Bayes (Naïve BayesMultinomial, Naïve Bayes), SVM (LibLINEAR, LibSVM, SMO), and Decision Tree (J48, REPTree, DecisionStump, HoeffdingTree, RandomTree, LMT). The combination of classification algorithm with suitable features selection classifier that gives highest accuracy (i.e., best number of features) along with the obtained number of features as shown in Table 5.2 and Figure 5.3.

Table 5.2: Combination for Suitable Features Selection Classifier with Classification Algorithm

Combination	Classification Algorithm	Features Selection Classifier	Number of Features
C1	Naïve BayesMultinomial	Naïve BayesMultinomial	8639
C2	Naïve Bayes	Naïve Bayes	9215
C3	LibLINEAR	SMO	9628
C4	LibSVM	LibSVM	9205
C5	SMO	SMO	9628
C6	J48	Naïve BayesMultinomial	8639
C7	REPTree	Naïve BayesMultinomial	8639
C8	DecisionStump	DecisionStump	1143
C9	HoeffdingTree	HoeffdingTree	6465
C10	RandomTree	HoeffdingTree	6465
C11	LMT	Naïve BayesMultinomial	8639

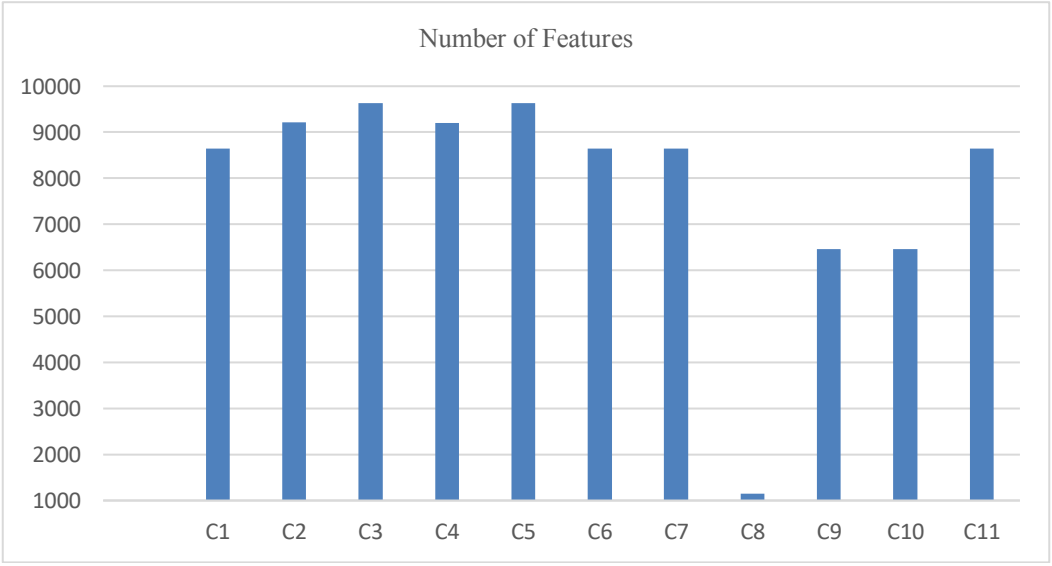


Figure 5.3: Number of Selected Features by Combination of Features Selection Classifier and Classification Algorithm

In this stage we found suitable feature subset for classification algorithm that achieves high accuracy when running a classification algorithm as shown in the Table 5.3.

Table 5.3: Accuracy, Precision, Recall, F-measure and Time for Machine Learning Method (Wrapper).

Classification Algorithm			Accuracy	Precision	Recall	F-Measure	Time
Naïve BayesMultinomial			80.3	.803	.803	.803	00:00:01
Naïve Bayes			73.7	.738	.737	.737	00:00:53
LibLINEAR			84.35	.844	.844	.843	00:00:03
LibSVM			84.05	.841	.841	.840	00:00:53
SMO			84.85	.849	.849	.848	00:01:39
Decision Tree	J48		69.9	.699	.699	.699	00:23:19
	REPTree		67.35	.674	.674	.673	00:07:33
	DecisionStump		62.45	.641	.625	.613	00:00:06
	HoeffdingTree		71.25	.720	.713	.710	00:01:32
	RandomTree		59.3	.593	.593	.593	00:00:07
	LMT		79.25	.793	.793	.792	05:04:48
Ensemble Method	Bagging	Bagging with J48	75.05	.751	.751	.750	03:09:34
		Bagging with REPTree	71.7	.717	.717	.717	00:58:35
		Bagging with RandomTree	63.15	.637	.632	.628	00:00:44
		Bagging with Naïve Bayes	76.1	.764	.761	.760	00:06:57
	Boosting	AdaBoost with J48	75.4	.754	.754	.754	05:04:16
		AdaBoost with REPTree	70.4	.704	.704	.704	00:21:27
		AdaBoost with RandomTree	58.1	.581	.581	.581	00:00:07
		AdaBoost with Naïve Bayes	73.6	.736	.736	.736	00:43:20
	Stacking	Stacking with (j48, HoeffdingTree) and REPTree meta Classifier	72.1	.721	.721	.721	02:50:45
		Stacking with (HoeffdingTree, DecisionStump, j48) and REPTree meta Classifier	73	.731	.730	.730	04:21:09
		Stacking with (Naïve Bayes, Naïve BayesMultinomial, SMO) and LMT meta Classifier	84.85	.849	.849	.848	00:39:14
		Stacking with (LibLINEAR, Naïve BayesMultinomial, SMO) and LMT meta Classifier	85.1	.851	.851	.851	00:17:20
	Vote	Vote with (LibLINEAR, Naïve BayesMultinomial, SMO, LMT)	85.7	.857	.857	.857	10:37:51

We can summarize the obtained high accuracy results by a specific algorithm of each category as follows:

High accuracy of SVM classification algorithms category obtained by using SMO algorithm. While the highest accuracy of Bayes classification algorithms is obtained by using Naïve BayesMultinomial. From the decision tree classification algorithms, the highest accuracy obtained by using LMT, but this algorithm takes more computation time. Similar to statistical method, bagging with algorithms give accuracy better than accuracy of original algorithm without any bagging, but also this takes more time. Also, boosting with these algorithms gives accuracy better than accuracy of original algorithms without any boosting. Again, this takes more time. Stacking with algorithms gives accuracy better than accuracy of the original algorithms as Stacking with (LibLINEAR, Naïve BayesMultinomial, SMO) and LMT meta classifier. Vote comparing to stacking with the same classification algorithms, gives accuracy slightly bettering. Also, more time is taking when using vote. So staking is better than vote.

The highest obtained accuracy is 85.1%, when using Stacking with (LibLINEAR, Naïve BayesMultinomial, SMO) and LMT meta classifier. The needed time in this case is 00:17:20 (hrs: mins: secs). On the other hand, the least computing time is 00:00:01 when using Naïve BayesMultinomial while the achieved accuracy is 80.3%.

Figures 5.4 and Figure 5.5 depict the obtained results of these experiments. Figure 5.4 shows the obtained accuracy when using feature selection method (Wrapper). Figure 5.5 shows training time for these algorithms.

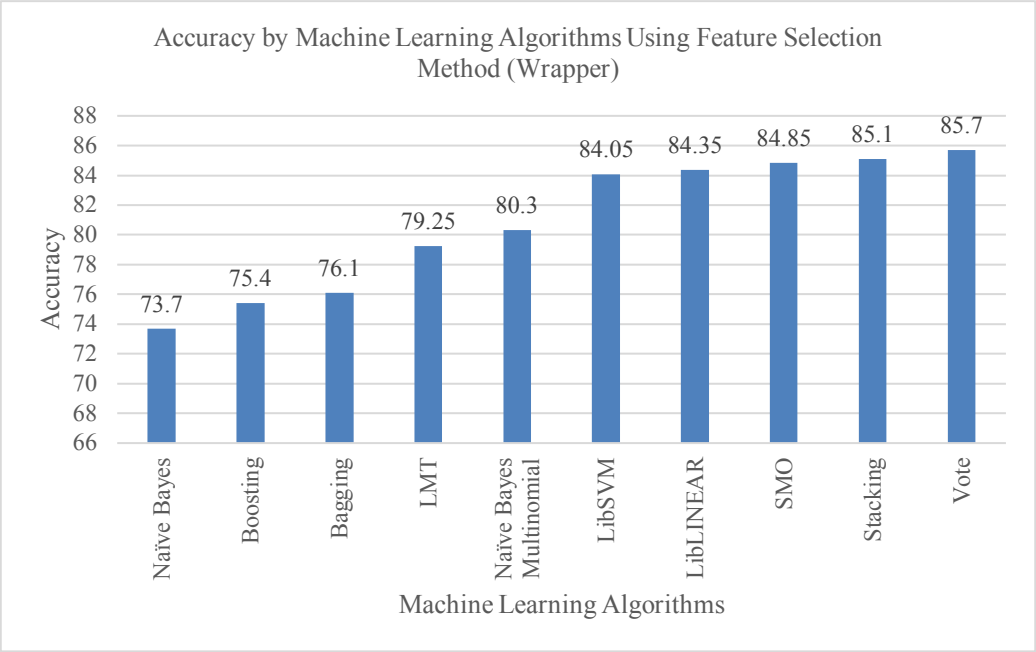


Figure 5.4: Accuracy by Machine Learning Algorithms Using Feature Selection Method (Wrapper)

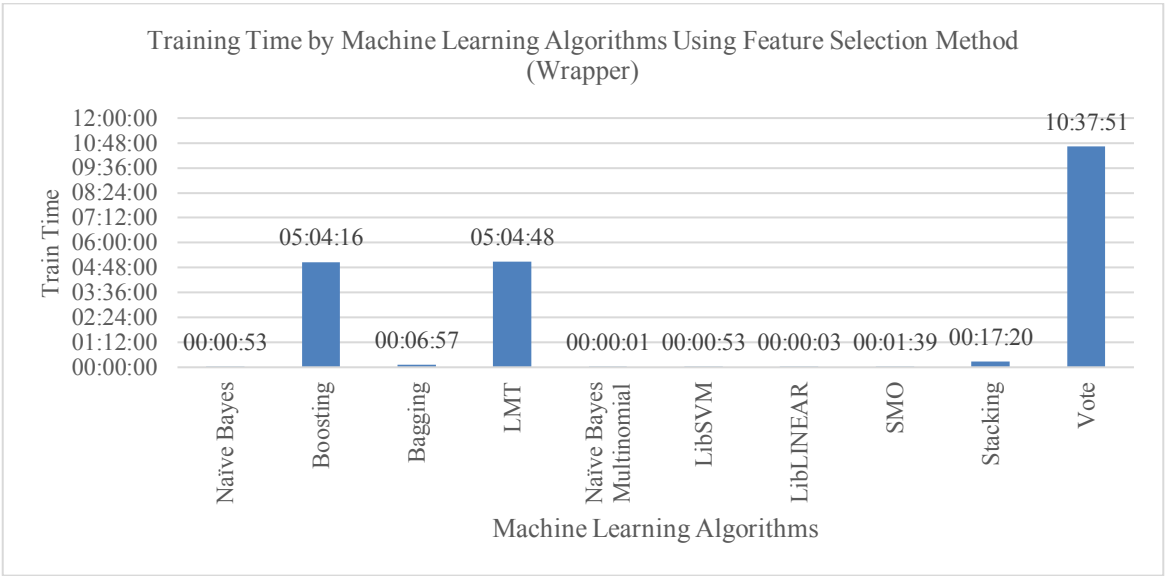


Figure 5.5: Training Time by Machine Learning Algorithms Using Feature Selection Method (Wrapper)

5.3.3 Embedded Method

Using Embedded method in Weka tool is called SVMAttributeEval that uses SVM. By this method we can decrease dataset dimensionality, and extract necessary features from dataset, to obtain high classification accuracy. We applied this method on features subset which obtained by statistical method (Correlation), and on feature subset which obtained by machine learning method (Wrapper).

- **Classification Accuracy Improvement by Reselecting Features from Feature Subset Obtained by Statistical Method**

We use the previous feature subset 3500 obtained by using the statistical method (Correlation) to extract more relevant features from this subset. This selection is carried out by applying SVMAttributeEval to obtain new feature subset. Consequently, the classification accuracy is improved as shown in Table 5.4.

Table 5.4: Accuracy for Embedded Method (Improved Statistical Method)

Classification Algorithm	**Accuracy**
Naïve BayesMultinomial	96.4
Naïve Bayes	86.35
LibLINEAR	99.75
LibSVM	97.65
SMO	99.65
J48	71.05
REPTree	69.15
DecisionStump	57.05
HoeffdingTree	80.9
RandomTree	66.5
LMT	92.6
Bagging with J48	79.25
Bagging with REPTree	77.15
Bagging with RandomTree	79.6
Bagging with Naïve Bayes	88.3
AdaBoost with J48	81.75
AdaBoost with REPTree	79.25
AdaBoost with RandomTree	67.35
AdaBoost with Naïve Bayes	87.6
Stacking with (j48, HoeffdingTree) and REPTree meta Classifier	80.85
Stacking with HoeffdingTree, DecisionStump, j48) and REPTree meta Classifier	80.25
Stacking with Naïve Bayes, Naïve BayesMultinomial, SMO) and LMT meta Classifier	99.6
Stacking with (LibLINEAR, Naïve BayesMultinomial, SMO) and LMT meta Classifier	99.75
Vote with (LibLINEAR, Naïve BayesMultinomial, SMO, LMT)	99.75

We summarize that improvement of classification algorithm accuracy is better compared with statistical method. Highest accuracy of SVM classification algorithms is when using LibLINEAR, SMO, LibSVM. Also, the highest accuracy is achieved when using Naïve BayesMultinomial of Bayes classification algorithm. Further, the highest accuracy is achieved by using LMT as one of decision tree classification algorithms. Bagging with algorithms give better accuracy than the original algorithm without any bagging. Moreover, boosting with algorithms give accuracy better than accuracy of original algorithm without any boosting. Also, stacking with algorithms give better accuracy than the original algorithm as Stacking with (LibLINEAR, Naïve BayesMultinomial, SMO) and LMT meta Classifier give highest accuracy. The training time of classification algorithms is not high because number of features is not big and more time elapsed by using LMT.

Figure 5.6 shows the obtained accuracy by classification algorithms using embedded feature selection method. Table 5.5 and Figure 5.7 depict the obtained accuracy by using statistical method compared with embedded method.

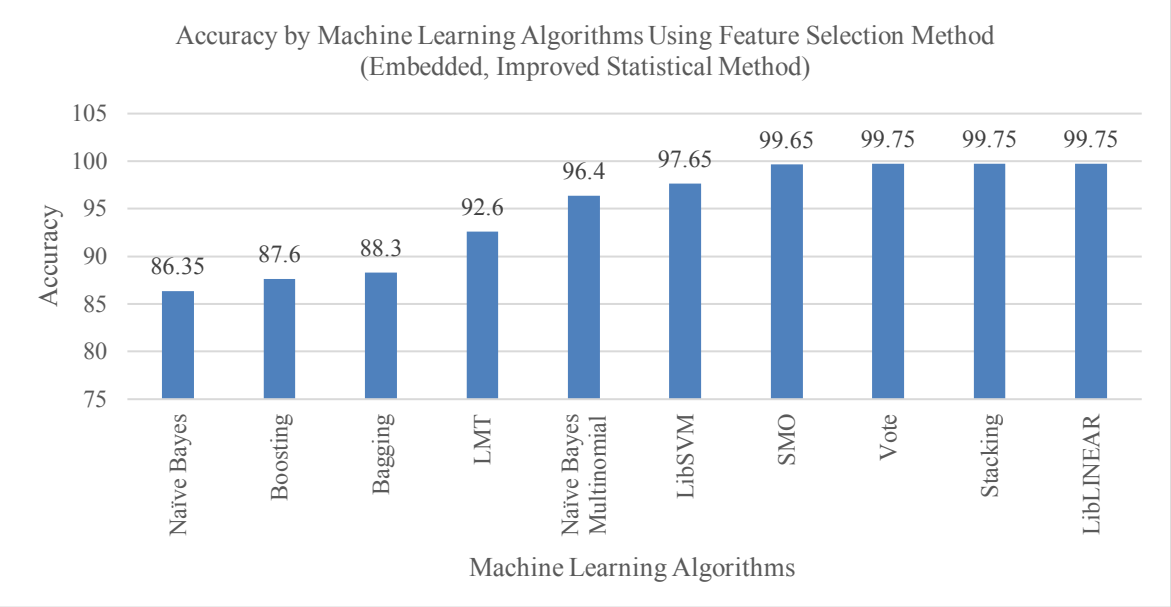


Figure 5.6: Accuracy by Machine Learning Algorithms Using Feature Selection Method (Embedded, Improved Statistical Method)

Table 5.5: Accuracy by Using Statistical Method Compared with Embedded Method

Classification Algorithm	**Accuracy Using Statistical Method**	**Accuracy Using Embedded Method (Improved Statistical Method)**
Naïve BayesMultinomial	95.7	96.4
Naïve Bayes	86.75	86.35
LibLINEAR	93.15	99.75
LibSVM	93.05	97.65
SMO	91.85	99.65
J48	68	71.05
REPTree	67.6	69.15
DecisionStump	62.45	57.05
HoeffdingTree	72.3	80.9
RandomTree	61.45	66.5
LMT	83.4	92.6
Bagging withJ48	77.1	79.25
Bagging with REPTree	74.3	77.15
Bagging with RandomTree	72.65	79.6
Bagging with Naïve Bayes	89.3	88.3
AdaBoost withJ48	77.25	81.75
AdaBoost with REPTree	73.6	79.25
AdaBoost with RandomTree	63.7	67.35
AdaBoost with Naïve Bayes	86.1	87.6
Stacking with (j48, HoeffdingTree) and REPTree meta Classifier	81.55	80.85
Stacking with HoeffdingTree, DecisionStump, j48) and REPTree meta Classifier	81.1	80.25
Stacking with Naïve Bayes, Naïve BayesMultinomial, SMO) and LMT meta Classifier	95.8	99.6
Stacking with (LibLINEAR, Naïve BayesMultinomial, SMO) and LMT meta Classifier	96	99.75
Vote with (LibLINEAR, Naïve BayesMultinomial, SMO, LMT)	93.45	99.75

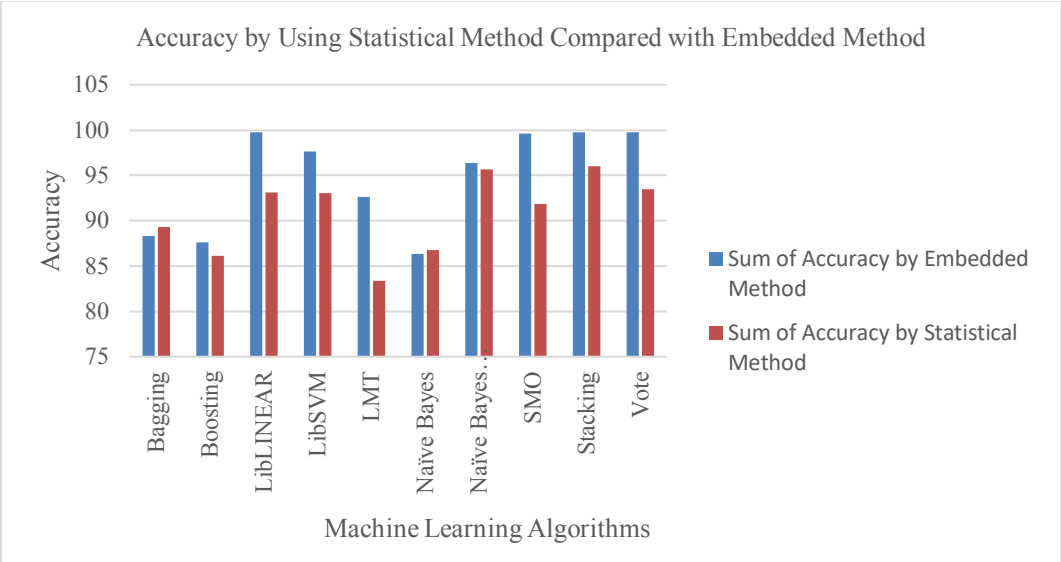


Figure 5.7: Accuracy by Using Statistical Method Compared with Embedded Method

- **Classification Accuracy Improvement by Reselecting Features from Feature Subset Obtained by Machine Learning Method**

In machine learning method we used classifiers to extract features. Each classifier produces different subset of features. Consequently, the resulting accuracy is also different. To optimize the accuracy in this stage we use embedded Method (SVMAttributeEval) to extract more relevant features from best feature subset that obtained by SMO classifier as shown in Table 5.6.

Table 5.6: Accuracy for Embedded Method (Improved Machine Learning Method)

Classification Algorithm	**Accuracy**
Naïve BayesMultinomial	92.8
Naïve Bayes	82.8
LibLINEAR	99.15
LibSVM	95.9
SMO	99.35
J48	69.15
REPTree	68.45
DecisionStump	62.45
HoeffdingTree	76.6
RandomTree	66.55
LMT	90.35
Bagging with J48	76.4
Bagging with REPTree	74.6
Bagging with RandomTree	77.05
Bagging with Naïve Bayes	83.65
AdaBoost with J48	78.05
AdaBoost with REPTree	75.8
AdaBoost with RandomTree	67.6
AdaBoost with Naïve Bayes	84.05
Stacking with (j48, HoeffdingTree) and REPTree meta Classifier	76.05
Stacking with HoeffdingTree, DecisionStump, j48) and REPTree meta Classifier	75.35
Stacking with Naïve Bayes, Naïve BayesMultinomial, SMO) and LMT meta Classifier	99.35
Stacking with (LibLINEAR, Naïve BayesMultinomial, SMO) and LMT meta Classifier	99.5
Vote with (LibLINEAR, Naïve BayesMultinomial, SMO, LMT)	99.45

We achieved high accuracies (ordered from highest to lowest accuracy) when using SMO, LibLINEAR, and LibSVM of SVM classification algorithms. While high accuracy is achieved by Bayes classification algorithm is Naïve BayesMultinomial. On the other hand, high accuracy obtained from decision tree classification algorithm is LMT. Bagging with algorithms give accuracy better than accuracy of original algorithm without any bagging. Boosting with algorithms give accuracy better than accuracy of original algorithm without any boosting. Stacking with algorithms outcome better accuracy than using of the original

algorithm such as Stacking with (LibLINEAR, Naïve BayesMultinomial, SMO) and LMT meta classifier give high accuracy. The training time of classification algorithms is not high because the number of features is relatively not big and most of them are LMT.

Figure 5.8 shows the obtained accuracy by classification algorithms using embedded method. Table 5.7 and Figure 5.9 depict the obtained accuracy by using machine learning method compared with embedded method.

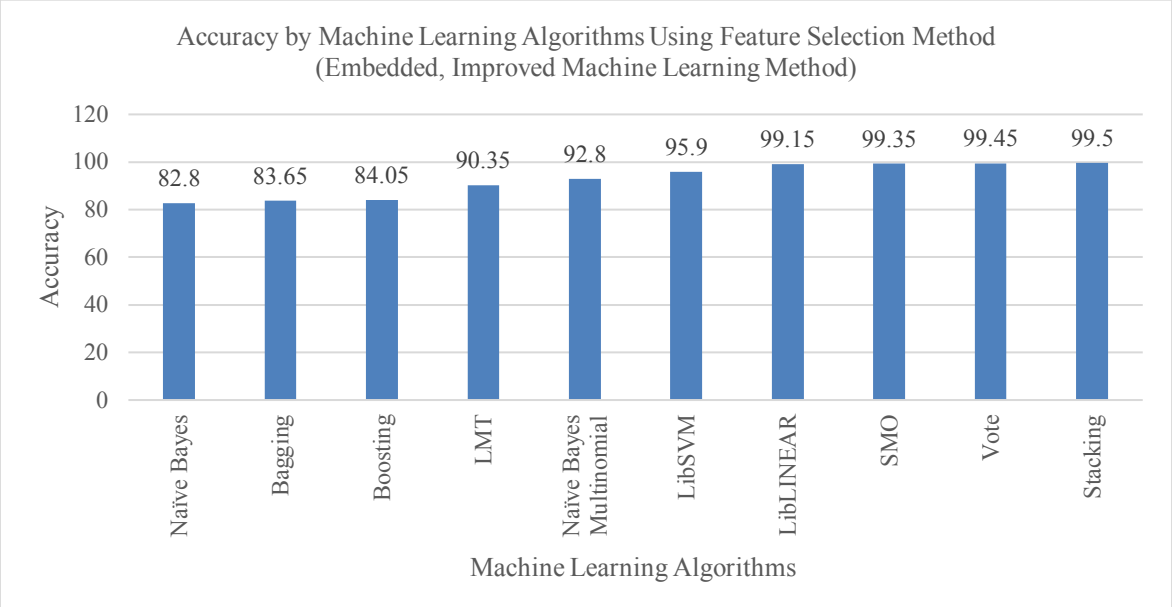


Figure 5.8: Accuracy by Machine Learning Algorithms Using Feature Selection Method (Embedded, Improved Machine Learning Method)

Table 5.7: Accuracy by Using Machine Learning Method Compared with Embedded Method

Classification Algorithm	**Accuracy Using Machine Learning Method**	**Accuracy Using Embedded Method** **(Improved Machine Learning Method)**
Naïve BayesMultinomial	80.3	92.8
Naïve Bayes	73.7	82.8
LibLINEAR	84.35	99.15
LibSVM	84.05	95.9
SMO	84.85	99.35
J48	69.9	69.15
REPTree	67.35	68.45
DecisionStump	62.45	62.45
HoeffdingTree	71.25	76.6
RandomTree	59.3	66.55
LMT	79.25	90.35
Bagging withJ48	75.05	76.4
Bagging with REPTree	71.7	74.6
Bagging with RandomTree	63.15	77.05
Bagging with Naïve Bayes	76.1	83.65
AdaBoost withJ48	75.4	78.05
AdaBoost with REPTree	70.4	75.8
AdaBoost with RandomTree	58.1	67.6
AdaBoost with Naïve Bayes	73.6	84.05
Stacking with (j48, HoeffdingTree) and REPTree meta Classifier	72.1	76.05
Stacking with HoeffdingTree, DecisionStump, j48) and REPTree meta Classifier	73	75.35
Stacking with Naïve Bayes, Naïve BayesMultinomial, SMO) and LMT meta Classifier	84.85	99.35
Stacking with (LibLINEAR, Naïve BayesMultinomial, SMO) and LMT meta Classifier	85.1	99.5
Vote with (LibLINEAR, Naïve BayesMultinomial, SMO, LMT)	85.7	99.45

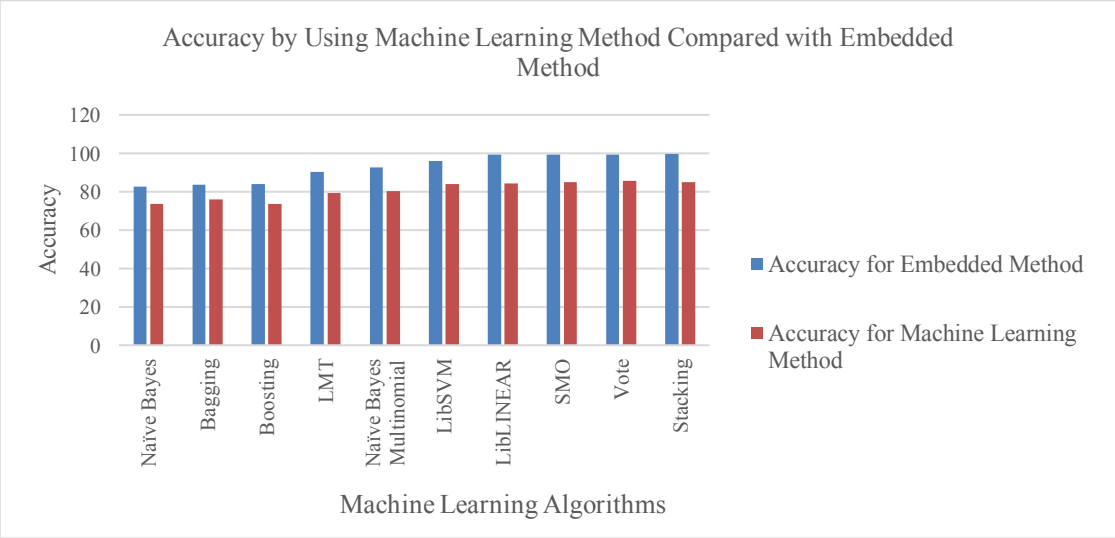


Figure 5.9: Accuracy by Using Machine Learning Method Compared with Embedded Method

5.4 Merge Feature Subsets

We can improve classification algorithms accuracy by merging feature subsets. In this case we merge two feature subsets (first subset that obtained after using embedded method on statistical feature subset with second feature subset that obtained after using embedded method on machine learning feature subset). After merging of two feature subsets a new feature subset is obtained consists of 828 features. Consequently, the classification accuracy is improved as shown in Table 5.8.

Table 5.8: Accuracy, Precision, Recall, F-measure and Time for Merging Feature Subsets Method

Classification Algorithm			Accuracy	Precision	Recall	F-Measure	Time
Naïve BayesMultinomial			96.95	.970	.970	.969	< 1 sec
Naïve Bayes			86.85	.869	.869	.868	00:00:04
LibLINEAR			99.8	.998	.998	.998	00:00:01
LibSVM			98.5	.985	.985	.985	00:00:10
SMO			99.65	.997	.997	.996	00:00:18
Decision Tree	J48		70.9	.709	.709	.709	00:01:47
	REPTree		67	.671	.670	.669	00:01:07
	DecisionStump		62.45	.641	.625	.613	00:00:05
	HoeffdingTree		83.55	.841	.836	.835	00:00:15
	RandomTree		64.7	.647	.647	.647	00:00:03
	LMT		89.05	.891	.891	.890	00:42:24
Ensemble Method	Bagging	Bagging with J48	77.85	.779	.779	.778	00:20:27
		Bagging with REPTree	75.45	.755	.755	.754	00:04:53
		Bagging with RandomTree	78.4	.792	.784	.782	00:00:11
		Bagging with Naïve Bayes	87.95	.880	.880	.879	00:00:35
	Boosting	AdaBoost with J48	80.75	.808	.808	.807	00:20:52
		AdaBoost with REPTree	77.95	.780	.780	.779	00:06:59
		AdaBoost with RandomTree	65.5	.655	.655	.655	00:00:03
		AdaBoost with Naïve Bayes	87.95	.882	.880	.879	00:04:53
	Stacking	Stacking with (j48, HoeffdingTree) and REPTree meta Classifier	83.45	.835	.835	.834	00:31:49
		Stacking with HoeffdingTree, DecisionStump, j48) and REPTree meta Classifier	83.3	.833	.833	.833	00:27:04

Classification Algorithm			Accuracy	Precision	Recall	F-Measure	Time
		Stacking with Naïve Bayes, Naïve Bayes Multinomial, SMO) and LMT meta Classifier	99.65	.997	.997	.996	00:02:34
		Stacking with LibLINEAR, Naïve Bayes Multinomial, SMO) and LMT meta Classifier	99.85	.999	.999	.998	00:04:17
	Vote	Vote with LibLINEAR, Naïve Bayes Multinomial, SMO, LMT)	99.85	.999	.999	.998	00:26:36

Figure 5.10 and Figure 5.11 depict the obtained results of this experiment. Figure 5.10 shows the obtained accuracy of classification algorithms when using merge feature subsets. Figure 5.11 shows training time for these algorithms.

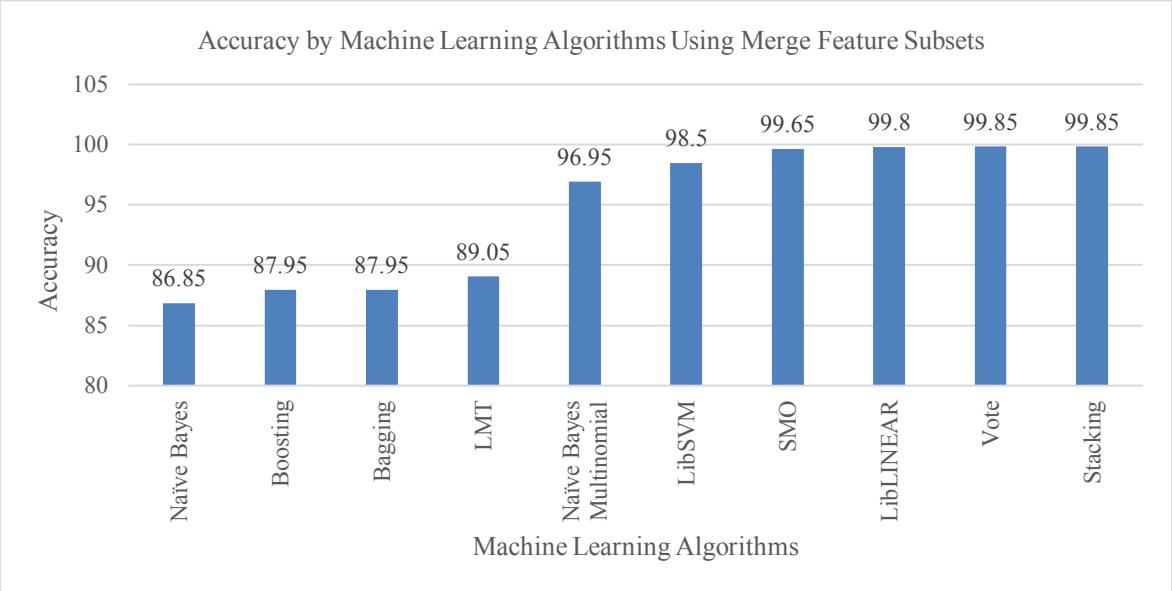


Figure 5.10: Accuracy by Machine Learning Algorithms Using Merge Feature Subsets

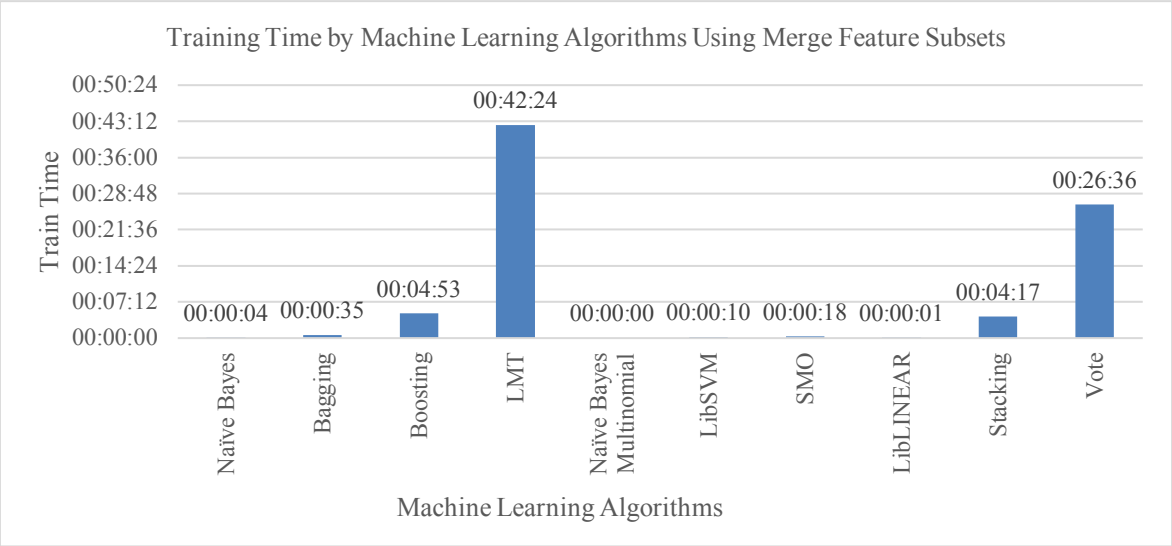


Figure 5.11: Training Time by Machine Learning Algorithms Using Merge Feature Subsets

We summarize the improvement of the accuracy using classification algorithm. Highest accuracy is obtained by using SVM classification algorithms LibLINEAR, SMO, and LibSVM. While in the case of using Bayes classification algorithm the high accuracy is obtained using Naïve BayesMultinomial. Also, high accuracy algorithm of decision tree classification algorithms is LMT. Again, bagging with these algorithms give accuracy better than the accuracy of original algorithm without any bagging. Moreover, boosting with algorithms give accuracy better than accuracy of original algorithm without any boosting. Also, stacking with algorithms give accuracy better than accuracy of the original algorithm as Stacking with (LibLINEAR, Naïve BayesMultinomial, SMO) and LMT meta classifier give high accuracy. The training time of classification algorithms is not high because number of features is not big and most of them are LMT. Vote comparing to stacking with the same classification algorithms give the same accuracy but vote need more training time. Thus, staking is better than vote.

The highest achieved accuracy is 99.85% when using Stacking with (LibLINEAR, Naïve BayesMultinomial, SMO) and LMT meta classifier. Where the elapsed time is 00:04:17 (hrs: mins: secs). The least time is less than one second in Naïve BayesMultinomial and accuracy is 96.95%. Also, the algorithm LibLINEAR give accuracy 99.8% with training time 00:00:01 (hrs: mins: secs).

5.5 Tuning Parameter (percentToEliminatePerIteration):

We studied the effect of using SVMAttributeEval method as improved method to select features in both cases from statistical features subset, and from wrapper features subset.

Then, we applied some classification algorithms on these features' subset obtained by SVMAttributeEval and found that the accuracy is improved.

Also, tuning parameter of SVMAttributeEval method instead of default values will increase performance of classification algorithms. So, it is important to know which parameters of this method and their proper values that increase the performance of classification model.

The parameter in SVMAttributeEval method is required to be properly tuned manually, because this tuning will affect the result of classification algorithm accuracy and time needed to select optimal feature subset. This parameter is used to determine percent rate of attribute elimination, and number of features reduced by value in each iteration. Consequently, we can modify the previous approach in Figure 4.2 by adding "parameter tuning" requirement in features selection step as shown in Figure 5.12. This figure depicts and illustrates this modified process flow of our methodology [62]. The result will be the better feature subset, less time needed to select this subset, and better achieved classification algorithm accuracy.

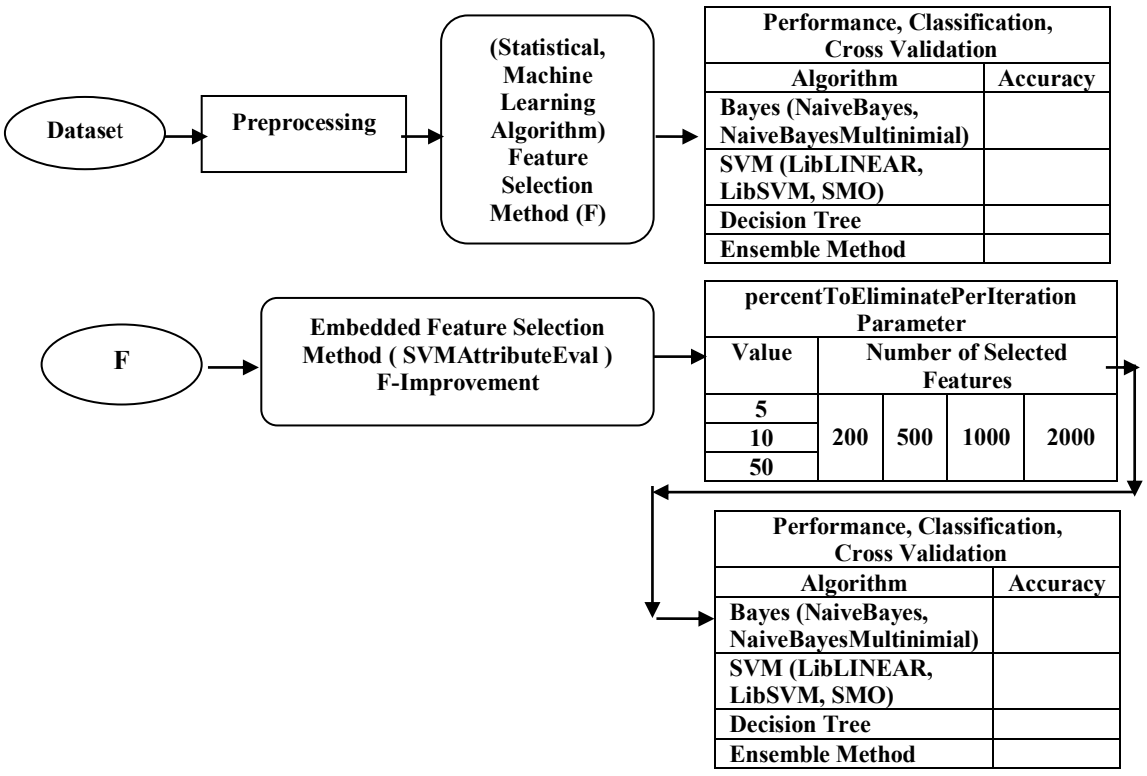


Figure 5.12: Research Methodology with Tuned Parameter Diagram

Tables 5.9, and 5.10 as shown depict the effect of tuning parameter on classification algorithms accuracy:

Table 5.9: Accuracy by Using Embedded Method (Improved Statistical Method), Tuning Parameter

Classification algorithm	**Percent Rate of Attribute Elimination = 5**				**Percent Rate of Attribute Elimination = 10**				**Percent Rate of Attribute Elimination = 50**			
	Number of Selected Features				**Number of Selected Features**				**Number of Selected Features**			
	200	500	1000	2000	200	500	1000	2000	200	500	1000	2000
Naïve BayesMultinomial	91.4	96.4	98.1	98.05	91.15	96	97.85	97.8	90.3	95.4	97.6	97.75
Naïve Bayes	83	86.35	87.35	87.8	82.45	86.15	87.45	87.8	82.9	85.4	87.7	87.55
LibLINEAR	93.1	99.75	99.75	98.6	91.65	99.45	99.9	98.5	91.2	96.7	98.75	98.2
LibSVM	91.6	97.65	98.2	95.8	91.75	97.1	97.8	95.45	91.15	96.1	97.7	95.85
SMO	94.05	99.65	99.8	98.2	92.8	99.7	99.8	98.15	92.25	97.6	98.9	98.1
Stacking with (LibLINEAR, Naïve BayesMultinomial, SMO) and LMT meta Classifie	94.2	99.75	99.75	98.85	92.7	99.7	99.9	98.7	91.65	97.6	99.3	98.65
Vote with (LibLINEAR, Naïve BayesMultinomial, SMO, LMT)	94.4	99.75	99.8	98.65	92.8	99.65	**99.95**	98.65	92.15	97.5	99.25	98.55

Table 5.10: Accuracy by Using Embedded Method (Improved Wrapper Method), Tuning Parameter

Classification algorithm	**Percent Rate of Attribute Elimination = 5**				**Percent Rate of Attribute Elimination = 10**				**Percent Rate of Attribute Elimination = 50**			
	Number of Selected Features				**Number of Selected Features**				**Number of Selected Features**			
	200	500	1000	2000	200	500	1000	2000	200	500	1000	2000
Naïve Bayes Multinomial	88.75	92.8	93.4	94.75	88.3	91.9	94.05	94.65	87.8	92.35	94.25	95
Naïve Bayes	80.6	82.8	83.35	79.45	79.6	81.7	82.45	87.8	80.45	81.6	82.3	87.8
LibLINEAR	91.1	99.15	99.9	**100**	90.7	99	99.85	**99.95**	89.5	97.6	99.75	99.8
LibSVM	90.25	95.9	96	95.1	89.35	94.55	96.1	95.05	88.65	94.45	95.8	95.15
SMO	92.45	99.35	**100**	**100**	91.9	99.45	**100**	99.9	90.2	97.65	99.9	99.7
Stacking with (LibLINEAR, Naïve BayesMultinomial, SMO) and LMT meta Classifier	92.6	99.5	99.9	**100**	91.5	99.35	**99.95**	**99.95**	90.05	97.8	99.9	99.75
Vote with (LibLINEAR, Naïve BayesMultinomial, SMO, LMT)	92.4	99.45	99.9	**100**	91.3	99.3	**100**	**99.95**	89.8	97.7	99.9	99.85

We conclude that the improvement of classification algorithms accuracy in statistical method reaches 99.95% when using vote algorithm, where the value of parameter is 10, and number of selected features is 1000.

Tuning of parameter affects the result of accuracy, and improvement of classification algorithms accuracy in wrapper method reaches 100% when using SMO, and Vote algorithms with value of the parameter is 10 and number of selected features are 1000. Accuracy reaches 100% when using LibLINEAR, SMO, Stacking, and Vote algorithms with value of parameter is 5 and number of selected features is 2000. Also, accuracy reaches 99.95% when using LibLINEAR, Stacking, and Vote algorithms with value of parameter is 10 and number of selected features is 2000.

5.5.1 The Time Needed to Select Feature Subset by Using Embedded Method (SVMAttributeEval)

In stage of feature subset selection by using embedded method SVMAttributeEval we need to tune parameter (percenToEliminatePerIteration) properly to reduce the time needed in the feature's selection step.

Table 5.11 as shown depicts and illustrates time needed to select features when value of parameter percenToEliminatePerIteration is 5, 10, 50 and number of selected features is 2000.

Table 5.11: The Time Needed by Using Embedded Method (SVMAttributeEval, Tuning Parameter)

	Percent Rate of Attribute Elimination = 5	**Percent Rate of Attribute Elimination = 10**	**Percent Rate of Attribute Elimination = 50**
	Number of Selected Features = 2000	**Number of Selected Features = 2000**	**Number of Selected Features = 2000**
Time (hrs: mins: secs).	31:49:15	07:23:34	00:03:45

We conclude if the value of parameter is big then time needed to select features will be small.

5.6 Tuning Parameter (numFolds in Stacking)

Stacking is a type of ensemble method to combine outputs from multiple classifiers [40]. Parameter numFolds in Stacking means inner cross-validation that determines number of folds used for cross-validation. For every partition of the outer cross-validation the inner cross-validation is repeated to obtain better performance of these classification algorithms. So tuning numFolds parameter affects the result of classification accuracy. Table 5.12 shows the effect of this parameter numFolds on accuracy when using merge feature subsets method:

Table 5.12: Accuracy by Using Merge Feature Subsets Method, Tuning Parameter (numFolds in Stacking)

Classification algorithm	**Merge Two Embedded feature subsets**			
	numFolds			
	12	11	10 default	9
Stacking with (LibLINEAR, Naïve BayesMultinomial, SMO) and LMT meta classifier	99.85	99.9	99.85	99.75

We conclude that tuning parameter numFolds affects the result of classification algorithms accuracy, and improvement of classification algorithm accuracy reaches 99.9% when numFolds is 11.

5.7 Comparison

Table 5.13 shows a comparison between this work and others work who used the same movie data:

Table 5.13: Comparison on Dataset = 2000 Review

Reference	Year	Approach	Features	Accuracy
[55]	2013	The performance of base and hybrid classifier NB-GA method, using TF-IDF, and feature selection by best first search method	Not mentioned	93.8
[58]	2014	Tuning of hyperparameters in random forest Classifier, Unigrams.	1942	91
[59]	2015	Feature extraction method that uses the dependency relation between words to extract features from text, using mRMR to select important features, present a concept extraction algorithm based on a novel concept parser scheme to extract semantic features, Unigrams, SVM	Not mentioned	90.1
[65]	2015	Lexicon pooled Naïve Bayes has high accuracy, POS Feature lexicon-based	Not mentioned	83.7
[63]	2014	Experimental results show that composite feature of prominent unigrams and prominent bi-tagged features perform better than other features for movie review sentiment classification, Information gain, NB, SVM.	2244	89.4 by SVM
				86.2 by Naïve Bayes
[64]	2017	Obtaining a high-quality minimal feature subset (Unigram, CHI, IG) by SVM	2311	91.33 by SVM
		(POS, CHI, IG) by NB	16669	94.13 by NB
This Work		Feature selection by using statistical method (Correlation)	3500	96
		Feature selection by using machine learning method (Wrapper), genetic algorithm	9628	85.1
		Feature selection by using embedded method (SVMAttributeEval), improved statistical method.	500	99.75
		Feature selection by using embedded method (SVMAttributeEval), improved machine learning method.	500	99.5
		Merge two embedded feature subsets (improvement on selection feature subset)	828	99.85
		Tuning parameter (percentToEliminatePerIteration) of SVMAttributeEval method	1000, 2000	99.95 when using embedded method on statistic feature subset 100 when using embedded method on wrapper feature subset
		Tuning parameter (numFolds in Stacking)	828	99.9

Table 5.14, and Figure 5.13 as shown depict classification algorithms accuracy compared with (feature selection methods, and merging two embedded feature subsets):

Table 5.14: Accuracy Comparison on Statistical, Machine Learning, Embedded, Merge Two Embedded Feature Subsets

Classification Algorithm	**Statistical**	**Machine Learning**	**Embedded (Improved Statistical Method)**	**Embedded (Improved Machine Learning Method)**	**Merge Two Embedded Feature Subsets**
Naïve BayesMultinomial	95.7	80.3	96.4	92.8	96.95
Naïve Bayes	86.75	73.7	86.35	82.8	86.85
LibLINEAR	93.15	84.35	99.75	99.15	99.8
LibSVM	93.05	84.05	97.65	95.9	98.5
SMO	91.85	84.85	99.65	99.35	99.65
J48	68	69.9	71.05	69.15	70.9
REPTree	67.6	67.35	69.15	68.45	67
DecisionStump	62.45	62.45	57.05	62.45	62.45
HoeffdingTree	72.3	71.25	80.9	76.6	83.55
RandomTree	61.45	59.3	66.5	66.55	64.7
LMT	83.4	79.25	92.6	90.35	89.05
Bagging with J48	77.1	75.05	79.25	76.4	77.85
Bagging with REPTree	74.3	71.7	77.15	74.6	75.45
Bagging with RandomTree	72.65	63.15	79.6	77.05	78.4
Bagging with Naïve Bayes	89.3	76.1	88.3	83.65	87.95
AdaBoost withJ48	77.25	75.4	81.75	78.05	80.75
AdaBoost with REPTree	73.6	70.4	79.25	75.8	77.95
AdaBoost with RandomTree	63.7	58.1	67.35	67.6	65.5
AdaBoost with Naïve Bayes	86.1	73.6	87.6	84.05	87.95
Stacking with (j48, HoeffdingTree) and REPTree meta Classifier	81.55	72.1	80.85	76.05	83.45
Stacking with HoeffdingTree, DecisionStump, j48) and REPTree meta Classifier	81.1	73	80.25	75.35	83.3
Stacking with Naïve Bayes, Naïve BayesMultinomial, SMO) and LMT meta Classifier	95.8	84.85	99.6	99.35	99.65
Stacking with (LibLINEAR, Naïve BayesMultinomial, SMO) and LMT meta Classifier	96	85.1	99.75	99.5	99.85
Vote with (LibLINEAR, Naïve BayesMultinomial, SMO, LMT)	93.45	85.7	99.75	99.45	99.85

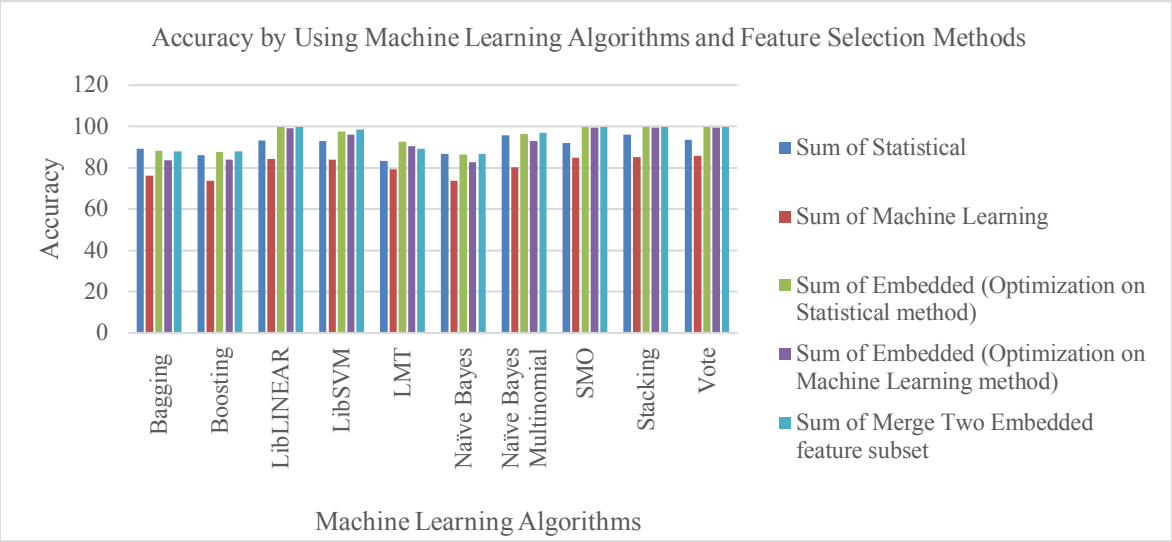


Figure 5.13: Accuracy by Using Machine Learning Algorithms and Feature Selection Methods

Figures 5.14, 5.15, 5.16, and 5.17 as shown depict ensemble methods accuracy compared with feature selection methods such as statistical, machine learning, embedded, and merge two embedded feature subsets.

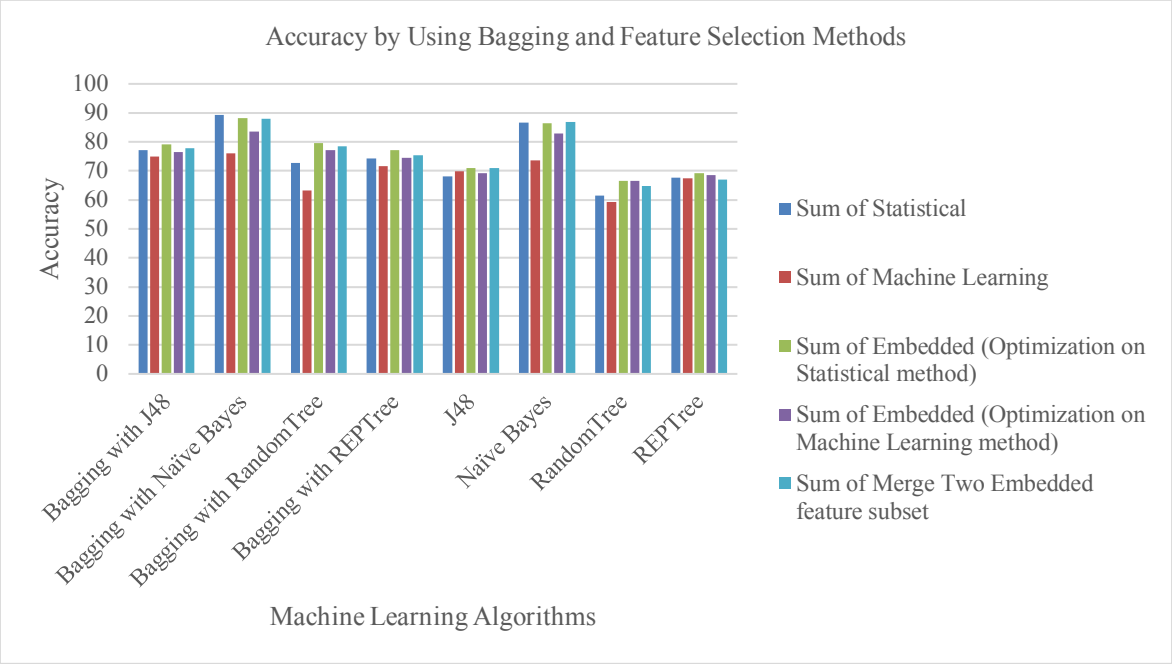


Figure 5.14: Accuracy by Using Bagging and Feature Selection Methods

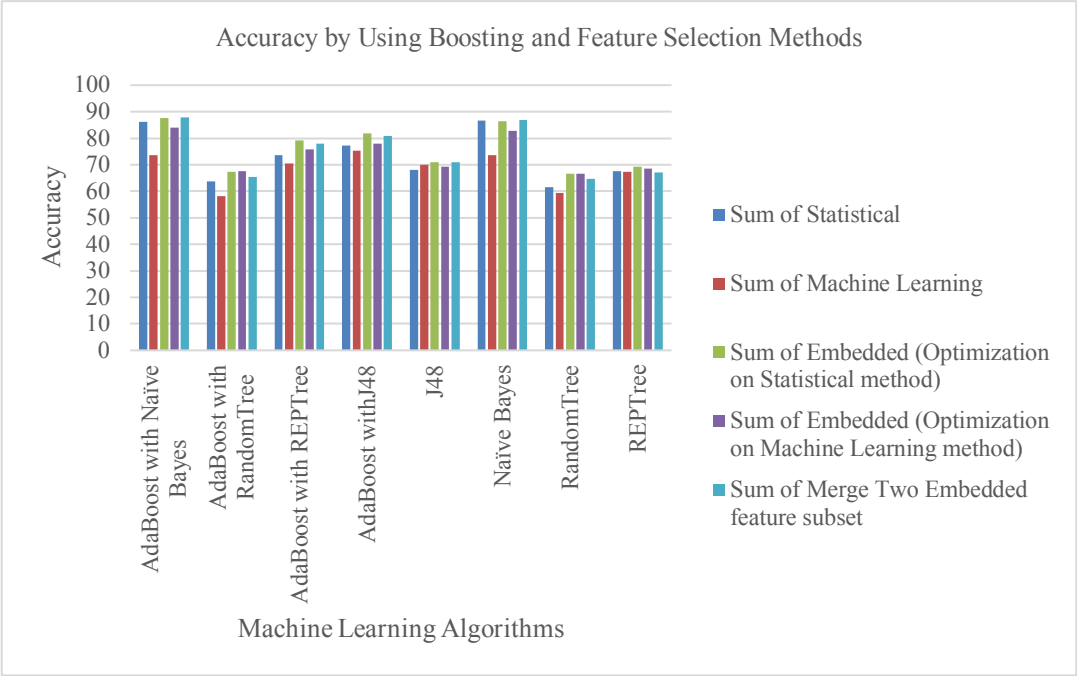


Figure 5.15: Accuracy by Using Boosting and Feature Selection Methods

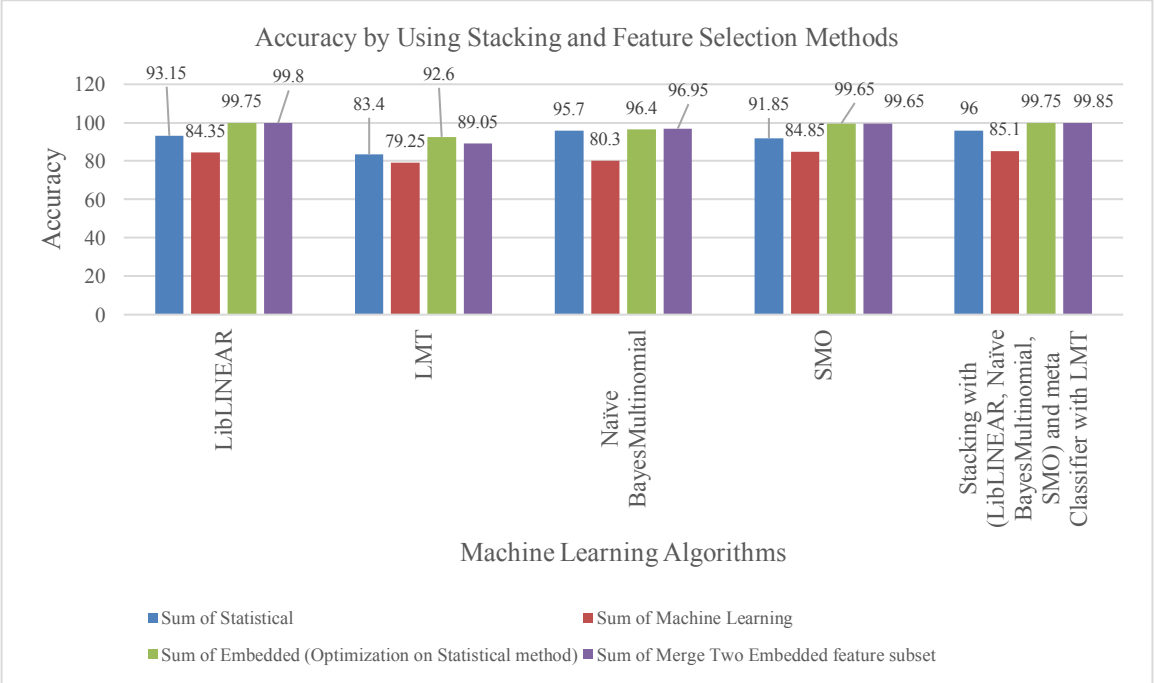


Figure 5.16: Accuracy by Using Stacking and Feature Selection Methods)

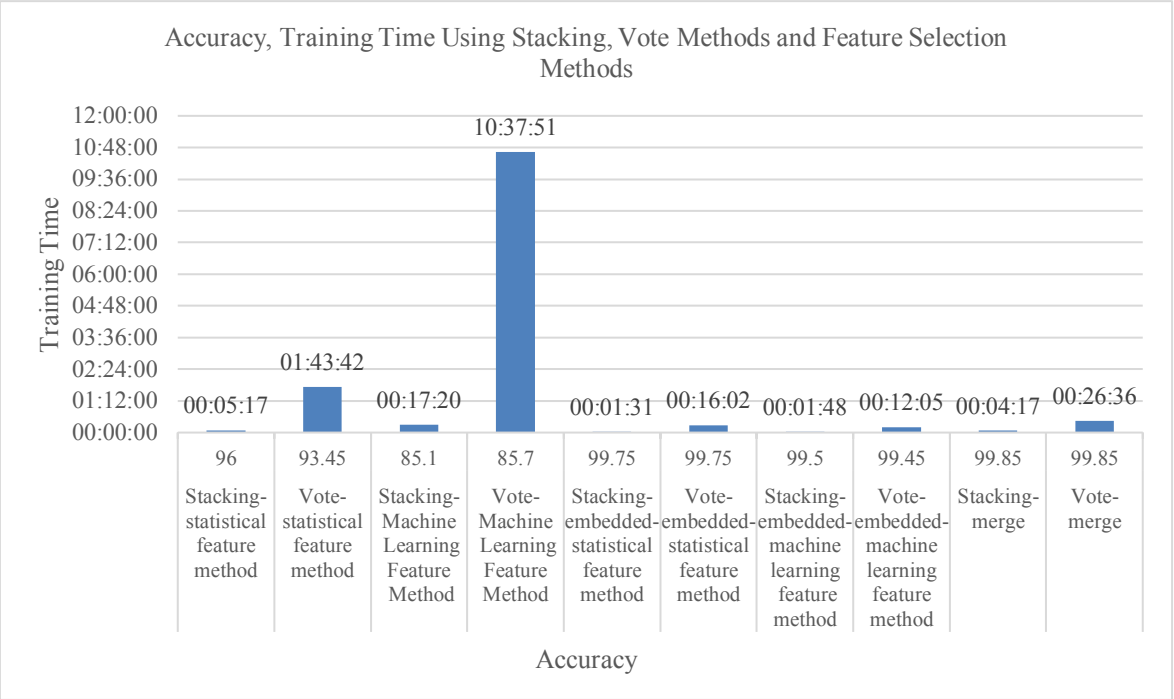


Figure 5.17: Accuracy, Training Time Using Stacking, Vote Methods and Feature Selection Methods

Table 5.15, and Table 5.16 as shown depict which classification algorithm achieved highest accuracy by using feature selection method, and which feature selection method suitable for classification algorithm.

Table 5.15: The Classification Algorithm that Give the Highest Accuracy by Using Feature Selection Method

This Work	Approach	The Classification Algorithm that Give the Highest Accuracy		Features	Accuracy
	Feature Selection by Using Statistical Method	Bayes	Naïve BayesMultinomial	3500	95.7
		SVM	LibLINEAR	3500	93.15
		Decision Tree	LMT	3500	83.4
		Ensemble Method	Bagging with Naïve Bayes	3500	89.3
			AdaBoost with Naïve Bayes	3500	86.1
			Stacking with (LibLINEAR, Naïve BayesMultinomial, SMO) and LMT meta Classifier	3500	96
	Feature Selection by Using Machine Learning Method	Bayes	Naïve BayesMultinomial	8639	80.3
		SVM	SMO	9628	84.85
		Decision Tree	LMT	8639	79.25
		Ensemble Method	Bagging with Naïve Bayes	9215	76.1
			AdaBoost withJ48	8639	75.4
			Stacking with (LibLINEAR, Naïve BayesMultinomial, SMO) and LMT meta Classifier	9628	85.1
	Feature Selection by Using Embedded Method (Improved Statistical Method)	Bayes	Naïve BayesMultinomial	500	96.4
		SVM	LibLINEAR	500	99.75
		Decision Tree	LMT	500	92.6
		Ensemble Method	Bagging with Naïve Bayes	500	88.3
			AdaBoost with Naïve Bayes	500	87.6
			Stacking with (LibLINEAR, Naïve BayesMultinomial, SMO) and LMT meta Classifier	500	99.75
	Feature Selection by Using Embedded Method (Improved Machine Learning Method)	Bayes	Naïve BayesMultinomial	500	92.8
		SVM	SMO	500	99.35
		Decision Tree	LMT	500	90.35
		Ensemble Method	Bagging with Naïve Bayes	500	83.65
			AdaBoost with Naïve Bayes	500	84.05
			Stacking with (LibLINEAR, Naïve BayesMultinomial, SMO) and LMT meta Classifier	500	99.5
	Merge Two Embedded Feature Subsets	Bayes	Naïve BayesMultinomial	828	96.95
		SVM	LibLINEAR	828	99.8
		Decision Tree	LMT	828	89.05
		Ensemble Method	Bagging with Naïve Bayes	828	87.95
			AdaBoost with Naïve Bayes	828	87.95
			Stacking with (LibLINEAR, Naïve BayesMultinomial, SMO) and LMT meta Classifier	828	99.85

Table 5.16: The Suitable Feature Selection Method for Classification Algorithm

Classification Algorithm	The Highest Accuracy	The Suitable Feature Selection Method
Naïve BayesMultinomial	96.95	Merge Two Embedded Feature Subset
Naïve Bayes	86.85	Merge Two Embedded Feature Subset
LibLINEAR	99.8	Merge Two Embedded Feature Subset
LibSVM	98.5	Merge Two Embedded Feature Subset
SMO	99.65	Feature Selection by Embedded Method (Improved Statistical Method) Merge Two Embedded Feature Subset
J48	71.05	Feature Selection by Embedded method (Improved Statistical Method)
REPTree	69.15	Feature Selection by Embedded method (Improved Statistical Method)
DecisionStump	62.45	More method gives the same accuracy
HoeffdingTree	83.55	Merge Two Embedded Feature Subset
RandomTree	66.55	Feature Selection by Embedded method (Improved Machine Learning Method)
LMT	92.6	Feature Selection by Embedded method (Improved Statistical Method)
Bagging withJ48	79.25	Feature Selection by Embedded method (Improved Statistical Method)
Bagging with REPTree	77.15	Feature Selection by Embedded method (Improved Statistical Method)
Bagging with RandomTree	79.6	Feature Selection by Embedded method (Improved Statistical Method)
Bagging with Naïve Bayes	89.3	Statistical Method
AdaBoost withJ48	81.75	Feature Selection by Embedded method (Improved Statistical Method)
AdaBoost with REPTree	79.25	Feature Selection by Embedded method (Improved Statistical Method)
AdaBoost with RandomTree	67.6	Feature Selection by Embedded method (Improved Machine Learning Method)
AdaBoost with Naïve Bayes	87.95	Merge Two Embedded Feature Subset
Stacking with Naïve Bayes, Naïve BayesMultinomial, SMO) and LMT meta Classifier	99.65	Merge Two Embedded Feature Subset
Stacking with (LibLINEAR, Naïve BayesMultinomial, SMO) and LMT meta Classifier	99.85	Merge Two Embedded Feature Subset
Vote with (LibLINEAR, Naïve BayesMultinomial, SMO) and LMT	99.85	Merge Two Embedded Feature Subset

Chapter 6

Conclusion and Future Work

6.1 Conclusion

The aim of this work is a comparative study on feature selection and ensemble methods for sentiment analysis (SA) classification using different classification algorithms. In this work we experimented with different combinations of several feature selection methods and several classification algorithms. We found that using some combinations of these methods and algorithms perform and produce classification accuracy better than other combinations. We tested and evaluated feature selection using statistical (Correlation), machine learning (Wrapper), and embedded (SVMAttributeEval) methods. We achieved an improvement on accuracy using improved statistical and machine learning methods by applying embedded method. Also, we improved the accuracy by using ensemble methods, merging two embedded feature subsets, and by changing the tuning parameter in SVMAttributeEval method. Furthermore, by changing the tuning parameter we were able to reduce the time needed to select features subset. We carried out many experiments using multiple classification algorithms to measure the classification performance.

The results of our experiment showed that the performance and the obtained accuracy depends on the feature selection method, ensemble method used, number of selected features, type of classifier, and tuning parameter of a method.

On the other hand, the time required to select features subset by SVMAttributeEval method was found to be dependent on the tuning parameters' value, so it is important to identify the best value to be used instead of using the default value.

In our experiment we achieved a high accuracy of 99.85% by merging features of two embedded methods when using ensemble method (Stacking with (LibLINEAR, Naïve BayesMultinomial, SMO) and LMT meta Classifier). This accuracy was better than the achieved accuracy of previous studies. Also, we achieved an improvement of accuracy when ensemble methods (Bagging, Boosting, Stacking, Vote) were applied. We were able to present the suitable feature selection method and training time for each classification

algorithm. Moreover, we achieved a high accuracy of up to 99.5% by tuning parameter of the stacking method, and a high accuracy of up to 99.95% and 100% by tuning parameter of the SVMAttributeEval method using statistical and machine learning approaches, respectively.

Based on the outcomes of the experimental results, we conclude that the accuracy increases when we select the best features, which is achieved by using improved embedded method. Thus, using ensemble methods, the accuracy increases. Also, the accuracy increased by modifying the tuning parameter of SVMAttributeEval and Staking methods. The time required for features selection subset decreased by changing the tuning parameter value in the SVMAttributeEval method.

6.2 Future Work

There are some other techniques that can be used for feature extraction which could improve classification accuracy other than those used in this study. For example, we can classify texts based on semantic aspects for twitter reviews.

The effect of changing the tuning parameter of an algorithm on the results of this research opens the door for researchers to use other parameters to improve classification accuracy, such as UseResampling, numIterations parameters in AdaBoost algorithm, and parameter reduced Error Pruning in J48 algorithm.

References

[1] Zin H., Mustapha.N, Murad M., and Sharef N., "The effects of pre-processing strategies in sentiment analysis of online movie reviews," AIP Conference Proceedings 1891, 020089, 2017.

[2] Isalm M. and Sultana N., "Comparative Study on Machine Learning Algorithms for Sentiment Classification," International Journal of Computer Applications, vol. 182, no. 21, pp. 1–7, 2018.

[3] Kumbhar P. and Mali M. "A Survey on Feature Selection Techniques and Classification Algorithms for Efficient Text Classification," International Journal of Science and Research (IJSR), vol. 5, no. 5, pp. 1267–1275, May 2016.

[4] Joshi N. and Srivastava S., "Improving Classification Accuracy Using Ensemble Learning Technique (Using Different Decision Trees)," 2014.

[5] Pant S. and Jain K., "Sentiment Analysis Using Feature Selection and Classification Algorithms- a Survey," International Journal of Innovative in Engineering Research and Technology [IJIERT] ISSN: 2394-3696 VOLUME 4, ISSUE 5, May 2017.

[6] Sahayak V., Shete V., and Pathan A., "Sentiment Analysis on Twitter Data," International Journal of Innovative Research in Advanced Engineering (IJIRAE) Issue 1, Vol. 2, January 2015.

[7] Gautam G. and Yadav D., "Sentiment analysis of twitter data using machine learning approaches and semantic analysis," 2014 Seventh International Conference on Contemporary Computing (IC3), 2014.

[8] Medhat W., Hassan A., and Korashy H., "Sentiment analysis algorithms and applications: A survey," Ain Shams Engineering Journal, vol. 5, no. 4, pp. 1093–1113, 2014.

[9] Binali H., Potdar V., and Wu C., "A state of the art opinion mining and its application domains," 2009 IEEE International Conference on Industrial Technology, 2009.

[10] B. Liu and L. Zhang, "A Survey of Opinion Mining and Sentiment Analysis," Mining Text Data, pp. 415–463, 2012.

[11] Greaves F., et al. "Use of Sentiment Analysis for Capturing Patient Experience from Free-Text Comments Posted Online," Journal of Medical Internet Research, vol. 15, no. 11, 2013.

[12] Behdenna S., Barigou F., and Belalem G., "Document Level Sentiment Analysis: A survey," EAI Endorsed Transactions on Context-aware Systems and Applications, vol. 4, no. 13, p. 154339, 2018.

[13] Kotsiantis S., Kanellopoulos D., and Pintelas P., “Data Preprocessing for Supervised Learning,” International Journal of Computer Science Volume 1 Number 1 ISSN 1306-4428, 2006.

[14] Saif H., et al., “On stopwords, filtering and data sparsity for sentiment analysis of twitter,” Proceedings of the 9th International Language Resources and Evaluation Conference (LREC'14). 810-817, 2014.

[15] Sabbah T., et al., “Hybridized term-weighting method for Dark Web classification,” Neurocomputing, vol. 173, pp. 1908–1926, 2016.

[16] MUNTEANU D., “Vector space model for document representation in information retrieval,” Annals of Dunarea de Jos, 2007.

[17] Hirapara Sh., et al., “Survey on Opinion Mining and Feature Selection,” International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization), https:// www.ijircce.com Vol. 5, Issue 3, March 2017.

[18] Wijayasekara D., Manic M., and Mcqueen M., “Information gain based dimensionality selection for classifying text documents,” 2013 IEEE Congress on Evolutionary Computation, 2013.

[19] Shareef S. and Hashim S., “False alarm reduction for Network Intrusion Detection System by using Decision Tree classifier,” JMAUC, vol. 10, no. 2, pp. 76-87, Jan. 2018.

[20] Saif H., et al., “Semantic Sentiment Analysis of Twitter,” The Semantic Web – ISWC 2012 Lecture Notes in Computer Science, pp. 508–524, 2012.

[21] O'Keefe T. and Koprinska I.,” Feature Selection and Weighting Methods in Sentiment Analysis,” Proceedings of the 14th Australasian Document Computing Symposium, 2009.

[22] Madasu A. and Elango S., “Efficient feature selection techniques for sentiment analysis,” Multimedia Tools and Applications, vol. 79, no. 9-10, pp. 6313–6335, 2019.

[23] Gnanambal S., et al., “Classification Algorithms with Attribute Selection: An Evaluation Study using WEKA,” International Journal of Advanced Networking and Applications, Volume: 09 Issue: 06 Pages: 3640-3644 (2018) ISSN: 0975-0290, April 2018.

[24] Jing L., et al., “Improved Feature Selection Approach TFIDF in Text Mining,” Proceedings. International Conference on Machine Learning and Cybernetics, doi:10.1109/icmlc.1174522, 2002.

[25] Oladipupo T., “Types of Machine Learning Algorithms,” New Advances in Machine Learning, 2010.

[26] Vadivukarassi M., et al., “Sentimental Analysis of Tweets Using Naive Bayes Algorithm,” World Applied Sciences Journal 35 (1): 54-59, 2017 ISSN 1818-4952 © IDOSI Publications, 2017.

[27] Wagh B., et al., “Sentimental Analysis on Twitter Data using Naive Bayes,” Ijarcce, vol. 5, no. 12, pp. 316–319, 2016.

[28] Suppala K. and Rao N., “Sentiment Analysis Using Naïve Bayes Classifier”, International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8 Issue-8 June 2019.

[29] Xu S., et al., “Bayesian Multinomial Naïve Bayes Classifier to Text Classification,” Lecture Notes in Electrical Engineering Advanced Multimedia and Ubiquitous Engineering, pp. 347–352, 2017.

[30] Abbas M., et al., “Multinomial Naive Bayes Classification Model for Sentiment Analysis,” International Journal of Computer Science and Network Security, VOL.19 No.3, March 2019.

[31] Bachhety S., et al., “Improved Multinomial Naïve Bayes Approach for Sentiment Analysis on Social Media,” Proceedings of 4th International Conference on Computers and Management (ICCM), 2018.

[32] Mohana R. and Sumathi S., “Document classification using Multinomial Naïve Bayesian Classifier,” International Journal of Science, Engineering and Technology Research (IJSETR), Volume 3, Issue 5, May 2014.

[33] Fan R., et al., “LIBLINEAR: a library for large linear classification,” Journal of Machine Learning Research. 9. 1871-1874. 10.1145/1390681.1442794, 2008.

[34] Ahmad M., et al., “SVM Optimization for Sentiment Analysis,” International Journal of Advanced Computer Science and Applications, vol. 9, no. 4, 2018.

[35] Bhargava N., et al., “Decision Tree Analysis on J48 Algorithm for Data Mining,” 2013

[36] Olayinka T.C. and Chiemeke S.C., “Predicting Paediatric Malaria Occurrence Using Classification Algorithm in Data Mining,” Journal of Advances in Mathematics and Computer Science, pp. 1–10, 2019.

[37] Oliver J. and Hand D., “Averaging over decision stumps,” Machine Learning: ECML-94 Lecture Notes in Computer Science, pp. 231–241, 1994.

[38] Srimani P. and Patil M. M., “Performance analysis of Hoeffding trees in data streams by using massive online analysis framework,” International Journal of Data Mining, Modelling and Management, vol. 7, no. 4, p. 293, 2015.

[39] Mishra A. and Ratha B., “Study of Random Tree and Random Forest Data Mining Algorithms for Microarray Data Analysis,” 2016.

[40] Re M. and Valentini G., “Ensemble methods: A review,” 2012.

[41] Amelio A. and Tagarelli A., “Data Mining: Clustering,” 10.1016/B978-0-12-809633-8.20489-5, 2017.

[42] Lei J., “Cross-Validation with Confidence,” Journal of the American Statistical Association, pp. 1–20, 2019.

[43] Visa S., et al., “Confusion Matrix-based Feature Selection,” CEUR Workshop Proceedings, January 2011.

[44] Hossin M. and Sulaiman M. “A Review on Evaluation Metrics for Data Classification Evaluations,” International Journal of Data Mining & Knowledge Management Process, vol. 5, no. 2, pp. 01–11, 2015.

[45] Vaghela V. and Jadav B., “Analysis of Various Sentiment Classification Techniques,” International Journal of Computer Applications, vol. 140, no. 3, pp. 22–27, 2016.

[46] Agarwal B. and Mittal N., “Optimal Feature Selection for Sentiment Analysis,” Computational Linguistics and Intelligent Text Processing Lecture Notes in Computer Science, pp. 13–24, 2013.

[47] Doshi M. and Chaturvedi S., “Correlation Based Feature Selection (CFS) Technique to Predict Student Perfromance,” International journal of Computer Networks & Communications, vol. 6, no. 3, pp. 197–206, 2014.

[48] Arafat H., et al.,” Different Feature Selection for Sentiment Classification,” International Journal of Information Science and Intelligent System, December 2013.

[49] Phyu T. and Oo N., “Performance Comparison of Feature Selection Methods,” MATEC Web of Conferences, vol. 42, p. 06002, 2016.

[50] Manek A., et al., “Aspect term extraction for sentiment analysis in large movie reviews using Gini Index feature selection method and SVM classifier,” World Wide Web, vol. 20, no. 2, pp. 135–154, 2016.

[51] Hamidi H. and Daraei A. “Analysis and evaluation of techniques for myocardial infarction based on genetic algorithm and weight by SVM,” Journal of Information Systems and Telecommunication, 2016.

[52] Gamal D., et al., “Twitter Benchmark Dataset for Arabic Sentiment Analysis,” International Journal of Modern Education and Computer Science, vol. 11, no. 1, pp. 33–38, 2019.

[53] Wu X., et al., “Top 10 algorithms in data mining,” Knowledge and Information Systems, vol. 14, no. 1, pp. 1–37, Apr. 2007.

[54] Khare P. and Burse K., "Feature Selection Using Genetic Algorithm and Classification using Weka for Ovarian Cancer," International Journal of Computer Science and Information Technologies, Vol. 7 (1), 2016.

[55] Govindarajan M., "Sentiment Analysis of Movie Reviews using Hybrid Method of Naive Bayes and Genetic Algorithm," International Journal of Computer Research, 2013.

[56] Pujari p. and Gupta J., "Improving Classification Accuracy by Using Feature Selection and Ensemble Model," International Journal of Soft Computing and Engineering (IJSCE), Vol. 2, Issue-2, May 2012.

[57] Syarif I., et al., "Application of Bagging, Boosting and Stacking to Intrusion Detection," Machine Learning and Data Mining in Pattern Recognition Lecture Notes in Computer Science, pp. 593–602, 2012.

[58] Parmar H., et al., "Sentiment Mining of Movie Reviews using Random Forest with Tuned Hyperparameters," 2014.

[59] Agarwal B., et al., "Concept-Level Sentiment Analysis with Dependency-Based Semantic Parsing: A Novel Approach," Cognitive Computation, vol. 7, no. 4, pp. 487–499, 2015.

[60] Goldberg D., "Genetic Algorithms in Search, Optimization, and Machine learning," Addison Wesley, 1989.

[61] Witten I., et al., (1999). "Weka: Practical machine learning tools and techniques with Java implementations," Working paper 99/11, Hamilton, New Zealand: University of Waikato, Department of Computer Science, 1999.

[62] Rijn J. and Hutter F., "Hyperparameter Importance Across Datasets," Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018.

[63] Agarwal B. and Mittal N., "Prominent feature extraction for review analysis: an empirical study," Journal of Experimental & Theoretical Artificial Intelligence, vol. 28, no. 3, pp. 485–498, 2014.

[64] Yousefpour A., Ibrahim R., and A. Hamed H., "Ordinal-based and frequency-based integration of feature selection methods for sentiment analysis," Expert Systems with Applications, vol. 75, pp. 80–93, 2017.

[65] Devaraj M., Piryani R., and Singh V., "Lexicon Ensemble and Lexicon Pooling for Sentiment Polarity Detection," IETE Technical Review, vol. 33, no. 3, pp. 332–340, 2015.

دراسة مقارنة طرق استخراج الميزات وطرق التعلم الجمعي مع خوارزميات التصنيف المختلفة لتصنيف وتحليل الآراء

إعداد: زاهر "محمد عدنان" يونس

إشراف: د. نضال كفري

ملخص:

يستخدم الناس الويب للتعبير عن آرائهم ومشاعرهم حول مواضيع مختلفة، مثل جودة المنتج والسياسة والحرب والخدمات والتعليم والعديد من المجالات الأخرى المثيرة للاهتمام. ومن هنا تأتي ضرورة تحليل هذا الكم الكبير من النصوص فيما يتعلق بموضوع معين ومعرفة ما يفكر به الناس وآرائهم.

تحليل المشاعر والآراء sentiment analysis (SA) هو حقل التنقيب عن النص، تصنف التعبيرات على أنها آراء إيجابية أو سلبية تجاه موضوع الاهتمام بعد تحديد تعبيرات المشاعر، والعلاقة مع الموضوع.

الهدف الرئيسي من هذا البحث هو دراسة مقارنة طرق استخراج الميزات من هذه النصوص وطرق التعلم الجمعي التي تم استخدامها مع خوارزميات التصنيف المختلفة لتصنيف تحليل المشاعر. أثناء مقارنة الخوارزميات المختلفة تبين ان هنالك تباين في دقة التصنيف. وذلك استنادًا إلى طرق اختيار الميزة (إحصائية، لغة الآلة، مضمنة). تبين وتوضح هذه الدراسة طريقة اختيار الميزة الأكثر ملاءمة لهذا النوع من البيانات ولخوارزميات التصنيف المختلفة. كذلك تظهر باستخدام طرق التعلم الجمعي (التعبئة، التعزيز، التراص، التصويت) والتي هي طرق الجمع بين خوارزميات التعلم المختلفة بحيث تدعم كل خوارزمية الخوارزمية الأخرى. كما تبين ان المصنف المدمج القوي بشكل أفضل من المصنف الفردي الذي يؤدي إلى دقة عالية. كما ان دمج ميزات فرعية بالطريقة المضمنة يحسن دقة التصنيف ويؤدي استخدام ضبط المعلمة لطرق تحديد الميزة إلى تحسين دقة التصنيف وكذلك تقليل الوقت المطلوب لاستخراج مجموعة فرعية من المميزات.

أظهرت النتائج أن الدقة التي تم الحصول عليها تعتمد على طريقة اختيار الميزة واستخدام طرق التعلم الجمعي وعدد الميزات المحددة ونوع المصنف وضبط المعلمة لطريقة استخراج البيانات. حيث تم التوصل الى تحقيق دقة عالية تصل إلى 99.85% عند استخدام طريقة التعلم الجمعي Stacking لمجموعة من المميزات التي تم دمجها. كما وصلت الدقة إلى 99.5% من خلال ضبط المعلمة في طريقة Stacking وكذلك حصلنا على دقة أعلى من خلال ضبط المعلمة في طريقة SVMAttributeEval وصلت الى 99.95% باستخدام الطريقة الإحصائية و 100% باستخدام لغة الآلة. أيضا باستخدام ضبط المعلمة يتم تقليل الوقت المطلوب لاستخراج مجموعة فرعية من المميزات.

ونستخلص بذلك انه للوصول الى دقة عالية في تصنيف الآراء لا يمكن الاعتماد على خوارزميات محددة ولكن علينا فحص الانسب من هذه الخوارزميات واختيار طريقة الدمج بينها لطبيعة ونوع النصوص والآراء المراد تصنيفها.