A Framework for Sentiment Analysis Classification based on Comparative Study

Zahir Younis, Nidal Kafri, Wael Hasouneh

Abstract: ***A number of Feature Selection and Ensemble Methods for Sentiment Analysis Classification had been introduced in many searches. This paper presents A frame work for sentiment analysis classification based on comparative study on different classification algorithms i.e., comparison between combinations of classification algorithms: Bayes, SVM, Decision Tree. We also examined the effect of using feature selection methods (statistical, wrapper, or embedded), ensemble methods (Bagging, Boosting, Stacking, or Vote), tuning parameters of methods (SVMAttributeEval, Stacking), and the effect of merging feature subsets selected by embedded method on the classification accuracy. 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. Thus, these combinations of algorithms can be followed as a frame work for sentiment analysis.***

Keywords: Artificial Intelligence, Sentiment Analyses, Machine Learning, Ensemble Methods, Feature Selection.

I. INTRODUCTION

Nowadays there are vast number of social media and huge number of users to those media on the internet. Its importance to human life is raising due to their facilitating communication and enabling public posting and commenting their opinions of their users. People express their opinions on various topics. Many posts and comments about the news, business, politics, education, entertainments and others every day issues through the web. This huge volume of data encourages researchers and institution to find out efficient ways to analyze this data and make it useful to their interests. One of these interests is to analyze people's opinions about a particular subject which is known as sentiment analysis SA. Sentiment analysis can be automated using artificial intelligence, particularly using machine learning classification approaches. Hence, it is important/necessary to find out efficient ways to analyze this big data of text to figure out people's interests' opinions regarding product, policies, products, services, and many other issues. Sentiment Analysis SA classifies expressions as either positive or negative opinions regarding the subject of interest. This can be achieved after identifying the sentiment expressions, determining their polarity, and its relationship to the subject [1]-[10]. Recently, opinion mining (OM) is an interesting topic for researchers using the availability of huge data provided by the Internet and World Wide Web (WWW). Since people often tend to be biased when analyzing data according to their personal preferences. Hence, developing and building an efficient, accurate sentiment analysis SA algorithm and model in unbiased manner the systems became a necessity to help decision makers to make the right decisions.

II. RELATED WORK

Researchers have studied SA, using various methodologies, algorithms and datasets. H. Zin et al. [1] discussed several pre-processing approaches (such as removing stop words, meaningless, numbers) that affect the classification performance of the online movie reviews. They claimed that Support Vector Machine (SVM) achieved high performance results for 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 highier 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, which is a 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] utilized ensemble technique (Bagging) to improve the classification accuracy. They used various decision Trees as base classifiers. S. Pant and K. Jain [5] conducted a survey on the 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 that automatically classifies the tweets as positive, negative or neutral respecting the query term.

Manuscript received on 12 August 2021.
Revised Manuscript received on 05 April 2022.
Manuscript published on 30 May 2022.
* Correspondence Author

Zahir Younis, Department of Computer Science, Al-Quds University, Al-Quds, Palestine. Email: zaheray@gmail.com

Nidal Kafri*, Department of Computer Science, Al-Quds University, Al-Quds, Palestine. Email: nkafri@staff.alquds.edu

Wael Hasouneh, Department of Computer Science, Al-Quds University, Al-Quds, Palestine. Email: hassouneh@staff.alquds.edu

Retrieval Number: 100.1/ijsce.A35240911121
DOI: 10.35940/ijsce.A3524.0512222
Journal Website: www.ijsce.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)
© Copyright: All rights reserved.

Their approach utilizes the pos-tagging and the Tree kernel to avoid the need for feature engineering. However, 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) which is called feature vector. After that, they select the feature vector list. Thereafter, they applied machine learning based classification algorithms namely: Naïve Bayes, maximum entropy (ME) and SVM. These algorithms used along with the semantic orientation based wordnet which extracts synonyms and similarity for the content feature. Their results showed that the Naïve Bayes technique when subjected to unigram model performed better than the ME and SVM. Also, the accuracy was again improved when the semantic analysis wordnet was followed up, which raises it from 88.9% to 89.9%.

III. BACKGROUND

SA is one application of Machine Learning techniques in Data Mining. Recall that it is the process of analyzing opinions and emotions to infer the tendencies shown in the analyzed data, and classify them into negative, positive or neutral [8][9]. Therefore, the text data under analysis need preprocessing before clasification.

A. Data Preprocessing

Data preprocessing is a critical and time-consuming step in SA. Consequently, feature space can be reduced by this step. It should be noted that preprocessing mean tokenization the text, stop word removal, and case normalization [1] [11].

B. Feature Selection

Extracting the correct features from unstructured text is crucial to SA. In another words, feature is relevant if its existence improves the classification performance and accuracy. On contrast, a feature is irrelevant if its existence decreases the classification performance. Thus, it is important to recognize and follow the right way to extract features [12]. To classify features as relevant, irrelevant, or redundant, we need to calculate Entropy and Information Gain (IG) of features to identify the ranks and weights of the features. Then, we can decide which feature has max IG [13].

C. Feature Types

N-gram (i.e., unigrams, bigrams, trigrams, etc.) is defined as a sequence of n contiguous terms of text. These terms can be letters, words, phonemes, and syllables. N-gram is a set of words that appear in a specific frame, in another words: n=1 is unigram, n=2 is bigram, n=3 is trigram and so on [14].

D. Feature Weighting

Feature weighting and ranking is necessary to find optimal relative weights of features to improve 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 assigned weights different methods such as statistical and machine learning method [15][16].

- *Weight by Correlation:* Correlation is one of statistical techniques by which features can be assigned weights with respect to a class. This weighting approach is based on correlation and it returns the absolute or squared value of correlation as feature weight [17]. Thereafter, upon the calculated weight the features with N top ranks are selected to be in the feature subset. Computing correlation helps in evaluating the worth of a feature. This can be done by measuring the Pearson's correlation between it and the class. It gives ranking of the features from higher to lower ranks. The result is the weight of features without support of any machine learning algorithm like J48, Naive Bayes (NB), SVM and others. It is known that Pearson correlation is the most used correlation statistic to measure the strength and relationship between linearly related features [17].

- *Weight by Machine Learning:* weight of features can be calculated using classifier in machine learning with respect to the class. By machine learning we train a model using subset of features. Then add or remove features from subset depending on results of the previous model. The ranking of features and the selected subsets of features from a dataset are depending on the used machine learning classifier [16].

E. Classification

Classification is a form of data analysis that builds models. These models (i.e., classifiers) describe important data categories. The classifier predicate the class label of unknown records and categorizes the feature in one of several predefined categories. This section introduces some classification algorithms such as Bayes, SVM, Decision Tree. Also, it introduces utilization of ensemble methods with these algorithms. There are vast number of supervised machine learning approaches and algorithms in the literature. Nest, we introduce the most popular and well-known classification algorithms.

- Bayes (Naïve Bayes Multinomial, Naïve Bayes).
- SVM Classifier (LibLINEAR, LibSVM, SMO).
- Decision Tree (J48, REP Tree, Decision Stump, Hoeffding Tree, Random Tree, Logistic Model Tree (LMT).

F. Ensemble Methods

Using single classifier for analyzing big data may not achieve high accuracy. However, combined classifiers (such as ensemble methods) may produce high accuracy. Ensemble method helps in reducing noise and variance that cause errors in learning [17]. The goal of utilizing ensemble learning (EL) in this research was to improve classification performance. This can be done by applying multiple machine learning algorithms. In turn, these algorithms 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 in parallel.
- *Boosting:* is similar to bagging, several training data sets are generated by random sampling. Each model is assigned different training data set. These models are processed sequentially. In boosting, weights are assigned to each model and the output is obtained by average weights of the models.
- *Stacking:* Stacking approach is used to combine different classifiers in two steps i.e., base learner and meta learner. In base learner many different models are used to learn from a dataset. As a result, new dataset is created by collecting the outputs of the models. After that, the produced dataset is used by a stacking model learner meta to produce the final output [17].
- *Voting:* is similar to stacking, but vote is used to combine different classifiers without learner meta.

IV. RESEARCH METHODOLOGY

This section presents the methodology and steps carried out in this work as shown in Figure 1. It includes preprocessing, feature selection methods, tuning parameter and comparison carried out amongst different classification learning algorithms with using different ensemble methods. Methodology and flow of this work was as follows:

- *Preprocessing:* the purpose of this step was to remove noise i.e. redundant features, irrelevant features, numbers, stop word, missing value. Then, convert uppercase letters to lowercase letters. This process was accomplished using TF-IDF to know frequency of terms in document and in corpus.
- *Feature Selection Methods*: features subsets were selected using several methods (statistical, machine learning, embedded). This work introduces approach to select features in the following steps:
  - In the first step statistical method (Correlation) was used to measure weights and ranks of the features and correlation between feature and class. This step was necessary to obtain subset of features with high ranks and weights.
  - Second step, in this step we selected features using machine learning method (Wrapper) with genetic search.
  - Third step, using embedded method to improve features selection from the features subset that generated from the first and second steps.
  - Fourth step, this step merges the features subsets obtained in the third step.
  - Fifth step, tuning parameter of feature selection method was carried out in this step.
- *Comparative study:* In this phase a comparative study on performance evaluation of machine learning algorithms was carried out with using ensemble methods (Bagging, Boosting, Stacking, Voting) for the following algorithms: Bayes (Naïve Bayes Multinomial, Naïve Bayes), SVM (LibLINEAR, LibSVM, SMO), Decision Tree (J48, REPTree, DecisionStump, HoeffdingTree, RandomTree, LMT).

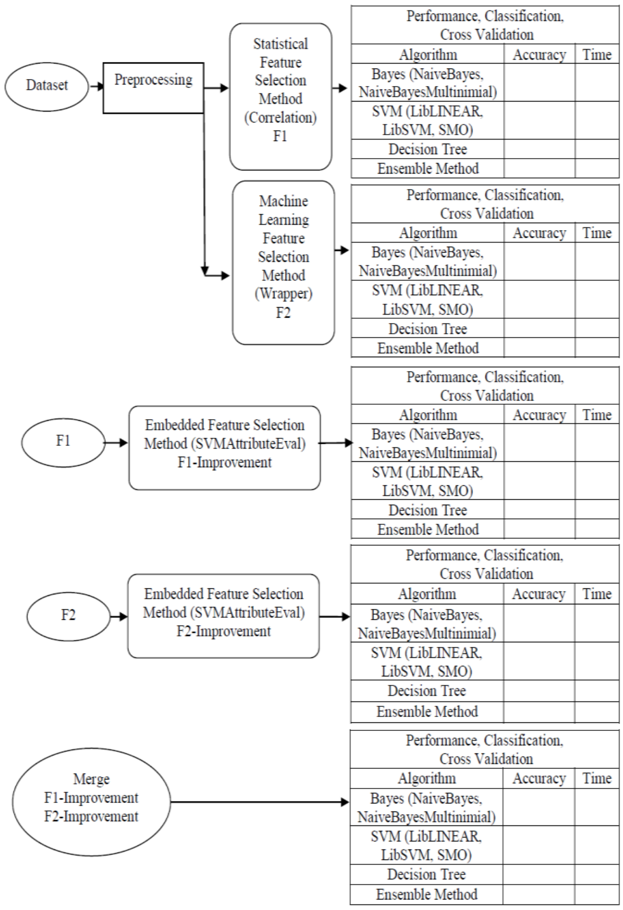

Dataset
Preprocessing
Statistical Feature Selection Method (Correlation) F1
Machine Learning Feature Selection Method (Wrapper) F2
F1
Embedded Feature Selection Method (SVMAttributeEval) F1-Improvement
F2
Embedded Feature Selection Method (SVMAttributeEval) F2-Improvement
Merge F1-Improvement F2-Improvement
Performance, Classification, Cross Validation
Algorithm
Accuracy
Time
Bayes (NaiveBayes, NaiveBayesMultinimial)
SVM (LibLINEAR, LibSVM, SMO)
Decision Tree
Ensemble Method

Figure 1: Research Methodology of This Work

V. EXPERIMENTAL RESULTS AND EVALUATION

The experiments carried out according the methodology explained in section IV. and depicted in Figure 1. Also, the experiments were implemented using Weka [1] software tool version 3.9.3 on MS Windows 10 Pro 64-bit operating system running on a laptop with Intel® Core™ i7-8550U CPU @ 1.80GHz 1.99 GHz and 8.00 GB RAM. The outcomes of these experiments show the classification accuracy of compared feature selection and ensemble methods using different classification algorithms. Recall that To select feature subsets we applied different methods. These methods were statistical (i.e., Correlation), machine learning (i.e., Wrapper), improved embedded, and improvement by merge features subsets. Furthermore, machine learning algorithms were used in the comparison. These machine learning algorithms were Bayes (NaiveBayes, NaiveBayesMultinimial), SVM (LibLINEAR, LibSVM, SMO), Decision Tree (J48, REPTree, DecisionStump, HoeffdingTree, RandomTree, LMT), and using ensemble methods (Bagging, Boosting, Stacking, Vote) along with these algorithms.

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

A. Research Scope and Dataset

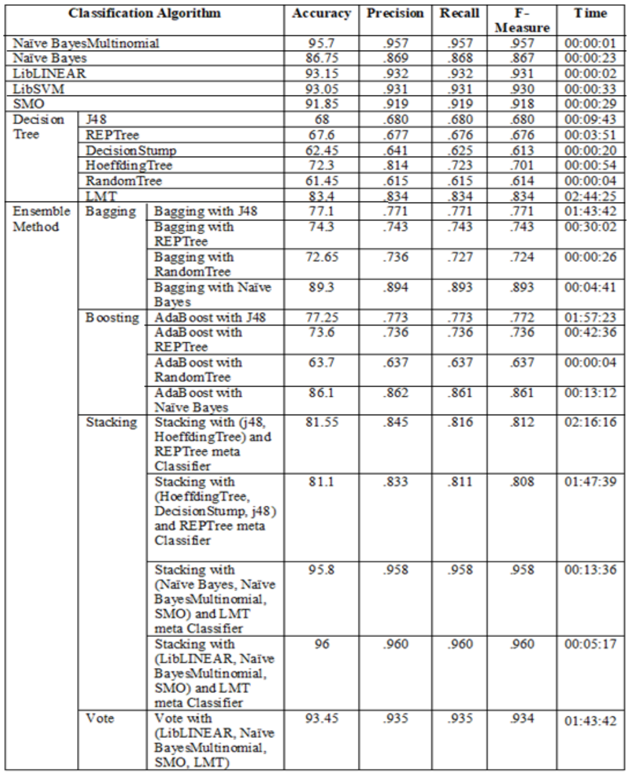
Particularly, the focus and scope of this work was to analyze people's opinions and their sentiment towards movies. For this purpose, we selected a Dataset that contains people's reviews and comments in English. The utilized Dataset (Movie Review Data) from Cornell university. The Polarity dataset v2.0 (3.0Mb) contains 1000 positive and 1000 negative processed reviews. This Dataset was introduced in Pang/Lee ACL 2004. Released June 2004.

B. Methods for Feature Selection

This section introduces the compared classification algorithms in this work.

- *Statistical Method (Correlation)*: Using statistical method in Weka tool [18] is called CorrelationAttributeEval. By Correlation usefulness of each feature for the classification process can be found. 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 of experiments, we found feature subset of 3500 feature. These features had the best ranking which produced high accuracy when running classification algorithms. These results are in the Table 1. This table shows the obtained accuracy as results of the experiments. The first column in the table shows the algorithms and their combination. The next columns the obtained result of accuracy, Precision, Recall and F-Measure. While the last column presents the elapsed time (i.e., cost) for each implemented algorithm.

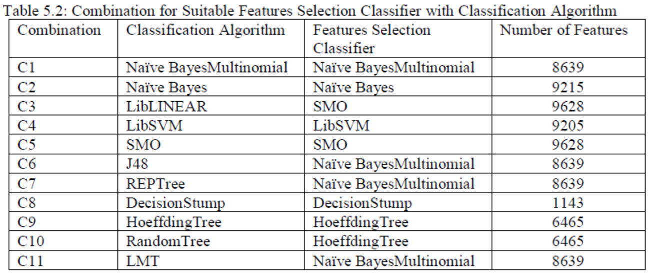
Table 1 Accuracy, Precision, Recall, F-measure and Time for Statistical Method (Correlation) by each Method.



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
		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

- *Machine Learning Method (Wrapper)*: Again, this method in Weka tool is called WrapperSubsetEval. Also, the search method is called genetic method. In our particular case we utilized search method to select features. The differences amongst the generated feature subsets depend on the used classifiers. Thus, after testing the performance of the well-known classifiers, we selected one with the highest obtained accuracy. The set of tested classifiers were Bayes (Naïve BayesMultinomial, Naïve Bayes), SVM (LibLINEAR, LibSVM, SMO), and Decision Tree (J48, REPTree, DecisionStump, HoeffdingTree, RandomTree, LMT). Table 2. Shows the used combination of classification algorithms with features selection classifier that produced highest accuracy (i.e., best number of features). It also shows the obtained number of features.

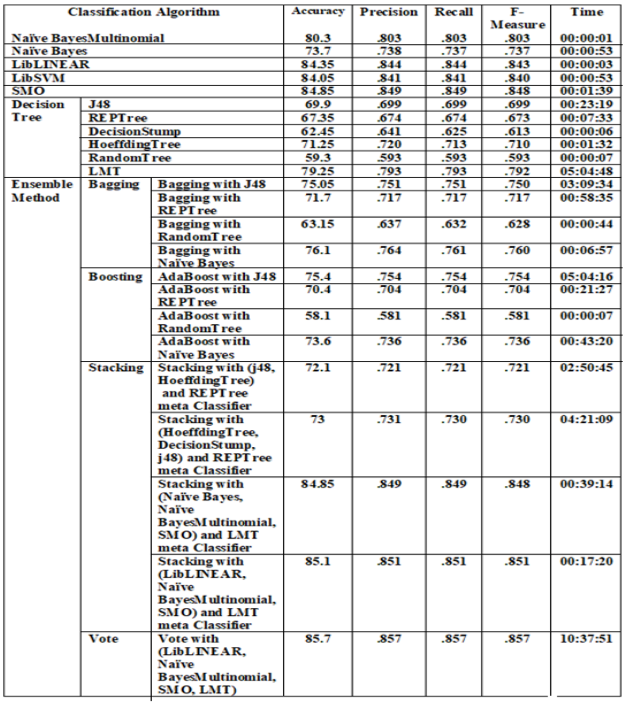
Table 5.2: Combination for Suitable Features Selection Classifier with Classification Algorithm



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

This step a suitable feature subset can be identified to be used for classification algorithm that achieves high accuracy. These results are depicted in Table 3. This table shows the classification algorithms along with the obtained results for accuracy, Recall, F-measure and the elapsed time spend using Wrapper machine learning method. For example, the highest accuracy (i.e., 85.7%) was achieved using vote with (LibLINeAR, Naïve BayesMultinomial, SMO, LMT) as ensemble methods in the last raw.

Table 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

Retrieval Number: 100.1/ijsce.A35240911121
DOI: 10.35940/ijsce.A3524.0512222
Journal Website: www.ijsce.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

- *Embedded Method*: Embedded method in Weka tool i.e., SVMAttributeEval uses SVM. this method enabled us to decrease dataset dimensionality, and extract necessary features from dataset, to obtain high classification accuracy. This method were applied on feature subset which obtained using correlation, and on feature subset which obtained using Wrapper machine learning method.

Classification accuracy can be improved by reselecting features from feature subsets obtained using different methods as follows:

- *Subset of Statistical Method*: We utilized the feature subset of 3500 features obtained by using the statistical method (Correlation) to extract more relevant features from this subset. This selection was done by applying SVM Attribute Eval. Consequently, the classification accuracy was improved as shown in Table 4. It shows that all accuracy results were improved. For example, the accuracy for vote with (LibLINeAR, Naïve Bayes Multinomial, SMO, LMT) increased from 85.7% to 99.75%. Similarly, all the accuracy for other methods were increased.

Table 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

- *Subset of Machine Learning Method*: for this subset, different classifiers were used to extract features. It should be noted that each classifier produced different subset of features. Consequently, the resulting accuracy was also different. To optimize the accuracy in this stage the embedded Method (SVMAttributeEval) was used to extract more relevant features from best feature subset that obtained by SMO classifier as shown in Table 5. It is clear that many algorithms achieved accuracy more than 99%.

C. Merge Feature Subsets

Moreover, classification algorithms accuracy can be improved using merging feature subsets. Thus, the two feature subsets (i.e., the obtained subsets using embedded method on statistical feature subset and the hat obtained using machine learning). After merging of two feature subsets a new feature subset was obtained consists of 828 features. Consequently, the classification accuracy was improved as shown in Table 6. In addition to the accuracy the table shows the achieved Precision, Recall, F-measure values and time cost for Merging Feature Subsets.

Table 5.: 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

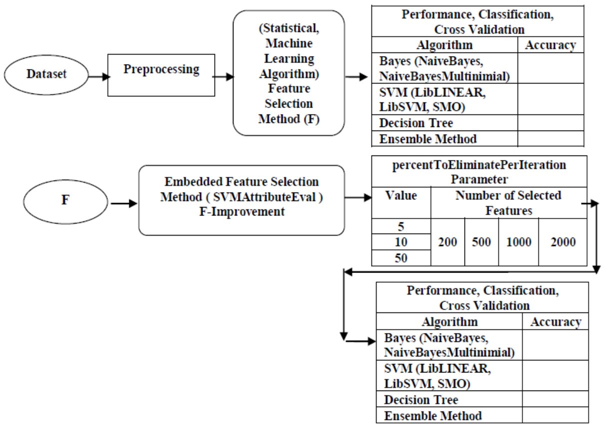

Dataset
Preprocessing
(Statistical, Machine Learning Algorithm) Feature Selection Method (F)
Performance, Classification, Cross Validation
Algorithm
Accuracy
Bayes (NaiveBayes, NaiveBayesMultinimial)
SVM (LibLINEAR, LibSVM, SMO)
Decision Tree
Ensemble Method
F
Embedded Feature Selection Method (SVMAttributeEval) F-Improvement
percentToEliminatePerIteration Parameter
Value
Number of Selected Features
5
10
50
200
500
1000
2000
Performance, Classification, Cross Validation
Algorithm
Accuracy
Bayes (NaiveBayes, NaiveBayesMultinimial)
SVM (LibLINEAR, LibSVM, SMO)
Decision Tree
Ensemble Method

Figure 2.: Research Methodology with Tuned Parameter Diagram

D. Tuning Parameter (percen To Eliminate Per Iteration)

One more important factor that affect accuracy is the parameter in SVMAttributeEval method. This parameter is required to be properly tuned manually [19]. This tuning affects 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 upgrade the previous approach in Figure 1. by adding "parameter tuning" in features selection step as shown in Figure 2. This figure shows and illustrates modified process flow of our framework. As a result, we obtained better feature subset, less time needed to select this subset, and better achieved classification accuracy [20-25].

Table 6.: 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
		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

Table 7. and Table 8. Show the effect of tuning parameter on classification accuracy:

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.

Also, tuning of the parameter improves the classification algorithms accuracy in wrapper method. It reached 100% when using SMO, and Vote algorithms with value of the parameter was 10 and number of selected features were 1000. Accuracy reached 100% when using LibLINEAR, SMO, Stacking, and Vote algorithms with value of parameter was 5 and number of selected features 2000. moreover, the accuracy reached 99.95% when using LibLINEAR, Stacking, and Vote algorithms with value of parameter 10 and number of selected features 2000.

- *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 (percen To Eliminate Per Iteration) properly to reduce the time needed in the feature's selection step.

Table 9. shows the time needed to select features when value of parameter percen To Eliminate Per Iteration was 5, 10, 50 and number of selected features 2000.

Table 7 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	928	99.65	**99.95**	98.65	92.15	97.5	99.25	98.55

We conclude that the bigger the value of the parameter the smaller the elapsed time to select features.

E. Tuning Parameter (num Folds in Stacking)

Recall that stacking is a type of ensemble method to combine outputs from multiple classifiers [17]. 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. Thus, tuning numFolds affects the result of classification accuracy. Table 10 shows the impact of this parameter on accuracy when using merge feature subsets method:We conclude that tuning numFolds parameter affects the result of classification accuracy. The obtained results shows that classification accuracy reached 99.9% when numFolds was 11.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

F. Comparison

Table 11 shows the obtained accuracy by our experiments and the obtained results of the related works that used the same movie data:

Table 8: 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

Table 9.: 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

Table 10: 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

Furthermore, the comparison of obtained accuracy of the classification algorithms used with feature selection methods, and merging two embedded feature subsets presented in Table 12:

Table 11: Comparison on Dataset = 2000 Review

Reference	Year	Approach	Features	Accuracy
[20]	2013	The performance of base and hybrid classifierNB-GA method, using TF-IDF, and feature selection by best first search method	Not mentioned	93.8
[21]	2014	Tuning of hyperparameters in random forest Classifier, Unigrams.	1942	91
[22]	2015	Feature extraction method that uses thedependency relation between words to extract featuresfrom text,using mRMR to select important features, present a concept extraction algorithm based on a novel conceptparser scheme to extract semantic features, Unigrams, SVM	Not mentioned	90.1
[23]	2015	Lexicon pooled Naïve Bayes has high accuracy, POS Feature lexicon-based	Not mentioned	83.7
[24]	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 NaïveBayes
[25]	2017	Obtaining a high-quality minimal feature subset (Unigram, CHI, IG) by SVM (POS, CHI, IG) by NB	2311	91.33 by SVM
			16669	94.13 by NB
This Work		Feature selection by using statisticalmethod (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 12: Accuracy Comparison on Statistical, Machine Learning, Embedded, and Merged 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

VI. CONCLUSION AND FUTURE WORK

The aim of this work was to find a frame work for sentiment analysis (SA) classification based on a comparative study on feature selection and ensemble classification method and algorithms. In this work experiments with different combinations of several feature selection methods and several classification algorithms were carried out. We found that using some combinations of these methods and algorithms perform and produce classification accuracy better than other combinations. Feature selection using statistical (Correlation), machine learning (Wrapper), and embedded (SVMAttributeEval) methods were tested and evaluated. An improvement on accuracy using improved statistical and machine learning methods by applying embedded method were achieved. moreover, the accuracy by using ensemble methods, merging two embedded feature subsets, and by changing the tuning parameter in SVMAttributeEval method were increased. Furthermore, by changing the tuning parameter, the time needed to select features subset was reduced.

The results of our experiments 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 these 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. Thus, the results of this work can be considered as a frame work for sentiment analysis. 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 can motivate the 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. 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.
12. 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.
13. Wijayasekara D., Manic M., and Mcqueen M., "Information gain based dimensionality selection for classifying text documents," 2013 IEEE Congress on Evolutionary Computation, 2013.
14. Saif H., et al., "Semantic Sentiment Analysis of Twitter," The Semantic Web – ISWC 2012 Lecture Notes in Computer Science, pp. 508–524, 2012.
15. O'Keefe T. and Koprinska I.," Feature Selection and Weighting Methods in Sentiment Analysis," Proceedings of the 14th Australasian Document Computing Symposium, 2009.
16. 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.
17. 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.
18. 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.
19. Rijn J. and Hutter F., "Hyperparameter Importance Across Datasets," Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018.
20. Govindarajan M., "Sentiment Analysis of Movie Reviews using Hybrid Method of Naive Bayes and Genetic Algorithm," International Journal of Computer Research, 2013.
21. Parmar H., et al., "Sentiment Mining of Movie Reviews using Random Forest with Tuned Hyperparameters," 2014.
22. 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.
23. 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.
24. 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.
25. 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.

AUTHORS PROFILE

Zahir M. A. Younis, was born in Tulkarm, Palestine and received his education at the An-Najah National University, where he obtained his B.Sc. degree in Computer Science in 2000. He worked at Palestinian Central Bureau of Statistics as a programmer for four years. Then he worked at Al-Quds Open University as an administrative employee and he is still. He obtained his MSc. degree in Computer Science in 2020 at the Al-Quds University. Email: zaheray@gmail.com.

Nidal M. S. Kafri, was born in Attil, Palestine and received his education at the Technical University of Plzen in Czech Republic, where he obtained his MSc. degree in Control Systems and Installation Management in 1982. He worked at Al-Quds University as a lecturer at the department of Computer Science for fifteen years. Then he moved to the Czech Technical University in Prague in 1997, where he obtained his Ph.D. degree in Distributed and Parallel computing in 2002. Since 2002, he worked as an assistant professor at the Department of Computer Science and IT, Al-Quds University, Palestine. His research interest is in Distributed and Parallel Computing, digital signal processing, algorithms and, simulation, AI and security. Email: nkafri@staff.alquds.edu.

Wael A. Y. Hassouneh, was born in Nablus, Palestine and received his education at the Odessa Polytechnic Institute in Odessa Ukraine, where he obtained his MSc. degree in computer Engineering, Specialization: Computers, systems, complexes and networks. in 1992. Then he continues his PH. D in Computer Engineering. Specialization: Computers, computing systems and networks, elements and devices of computers, equipment and control systems, at Odessa State Polytechnic University, Ukraine. His Ph.D. dissertation on: “Methods of Abridged Execution of Operations and their Hardware Check in Arithmetic Devices”. Since 1997, he worked as an assistant professor at the Department of Computer Science and IT, Al-Quds University, Palestine. His research interest is in Distributed and Parallel Computing, superscalar and Multicore processors, Control and embedded systems. Email: hassouneh@staff.alquds.edu.

Retrieval Number: 100.1/ijsce.A35240911121
DOI: 10.35940/ijsce.A3524.0512222
Journal Website: www.ijsce.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)