ISSN 1512-1232

BINARY TEXT CLASSIFICATION USING AN ENSEMBLE OF NAÏVE BAYES AND SUPPORT VECTOR MACHINES

Abikoye Oluwakemi Christiana[1], Omokanye Samuel Oladeji[2], Aro Taye Oladele[3]

[1]Department of Computer Science, University of Ilorin, Ilorin, Nigeria
Email: kemi_adeoye@yahoo.com
[2]Department of Computer Science, University of Ilorin, Ilorin, Nigeria
Email: oladejiomokanye@yahoo.com
[3]Department of Computer Science, University of Ilorin, Ilorin, Nigeria
Email: taiwo_aro@yahoo.com

Abstract

Text classification is being done by classifiers over the years, combining classifiers together can result in better classification and thus Naïve Bayes algorithm is combined with Support vector machine through stacking and the results shows that the ensemble results in an increase in the classification accuracy though at the expense of the time taken by the ensemble to build its classification model.

Keywords: *Naïve Bayes, Support vector machine, text classification, ensemble, binary.*

1.0 Introduction

Research on Text classification (TC) focuses on finding more appropriate ways to represent documents, index such documents and constructing of classifiers to assign each document to its correct category based on the standard in consideration [1]. TC is a supervised machine learning task because the algorithms learn from examples to be able to perform its tasks as compared to unsupervised machine learning where there are no examples to learn from. TC task can be in two dimensions, the first being to classify documents to only a single category while the other is classification in which a document can belong to more than one category [2]. Text classification consists of document representation, feature transformation and/or feature selection, construction of a vector space model, application of data mining algorithm and finally an evaluation of the applied data mining algorithm [3]. Text classification is the task of classifying a document under a pre-established category. More formally, if di is a document of the entire set of documents D and $\{c_1, c_2, c_3, \ldots c_n\}$ is the set of all the categories, then text classification assigns one category c_j to a document di [4]. Classification finds a model that separates classes or data concepts in order to predict the classes of unknown objects, take for instance a school will want to determine which of its final year student can be graduated, we have two categories, “graduate” and “spill” for the final year student data, the two categories can be represented by discrete values and the way it is ordered is irrelevant to the classification. Such is called supervised learning due to the fact that the classes in the training data have been labelled already. A machine learning algorithm builds a classifier in two stages. (1) Training builds a classification model by analysing training data that has class labels and (2) testing examines a classifier (using the test data) for accuracy and classify unknown objects into their respective classes. A machine learning algorithms first builds the model to be used for classification by analysing a training data which has class labels in it, then the classifier’s model is evaluated by using a testing data, such evaluation will be for accuracy in its ability to classify unknown data to its proper class, after which the classifier can then be deployed for real world use.

ISSN 1512-1232

2.0 Text Classification

Text classification is the task of classifying a document under a pre-established category. More formally, if di is a document of the entire set of documents D and $\{c_1, c_2, c_3, ...c_n\}$ is the set of all the categories, then text classification assigns one category c_j to a document di [4]. The documents, based on their characteristics can be labelled as belonging to a single class or for multiple classes in possible situations. If a document can only be assigned to one class, it is called a single-labelled classification and if the document can be assigned to more than one class, it is called a multi-labelled classification [5]. A single-labelled text classification problem can be further classified as a binary class problem if a data item can only be assigned into one of two classes and becomes a multi-class problem if there are more than two classes in which a data item can be assigned to.

The processes of text classification found in literature and as discussed by [2], [6] as illustrated in figure 1.0 are

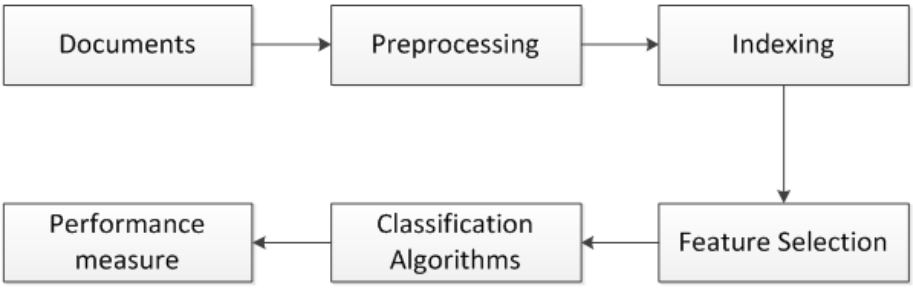

Figure 1: Text classification process (Source: Bhumika *et al*, 2013)

1. **Document collection**

   Documents to be classified are collected which can be in different formats like html, doc, pdf etc.

2. **Pre-processing**

   Because of the high dimensionality of text data, pre-processing is done to reduce it so as to present the data in a clear word format to allow for efficient representation and manipulation of data[7]. Common steps taken in pre-processing are:
   Tokenization: The document is treated as a string, and thus separated into a list of tokens
   Removing Stop words: Stop words are words that occur frequently but are insignificant for analysis such as “the”, “a”, “or”.
   Stemming word: Words are stemmed so as to convert different word form into the same canonical form. This step conflates tokens to their root form, e.g. modelling to model, associating to associate etc.

3. **Indexing**

   The complexity of documents are reduced by indexing them during pre-processing and thus making the documents easier to work on. The full text is converted to a document vector. A widely used document representation is called vector space model [8], it represents documents by creating a vector of the words in the document. Bag of words/Vector space model (BOW/VSM) document representation scheme also has its limitations, some of which are: high dimensionality of the data representation, loss of correlation with adjacent words and loss of semantic relationship that exist among the terms in a document. A way to

ISSN 1512-1232

overcoming such problems is by using term weighting methods which assigns proper weights to the term.

4. Feature selection

After pre-processing and indexing, feature selection is an important next step in classifying text [9]. Feature selection (FS) is the process of detecting relevant features while removing irrelevant, redundant or noisy data so as to speed up the data mining algorithms, improve predictive accuracy and increase comprehensibility.[10] FS is important because in many cases it's not all the data available in a dataset that are important in classifying such data. In many real world problems FS is a must due to the abundance of noisy, irrelevant or misleading features [11]. According to [11], FS has many advantages which are; it reduces the dimensionality of the feature space, removes irrelevant, redundant or noisy data, improves the quality of data, speeds up the running time of the learning algorithm, increases the accuracy of the resulting model, performance improvement to gain in predictive accuracy and better understanding of the data. There are different FS algorithms each motivated by a certain evaluation of which attribute is relevant and which is not. Some of the FS algorithms are TF-IDF, Chi-squared, Principal component analysis, T-test, Euclidean distance, information gain amongst many others.

5. Classification

The elements of the given data are classified into predefined categories, the machine learning algorithms can learn in three ways, unsupervised learning, supervised learning, and semi-supervised learning. Supervised learning is such that labelled data is used for training the machine learning algorithm i.e. data that has been assigned to predefined categories are used, so the algorithm will learn the way such data is classified and use what it learns to assign unlabelled data into categories. Semi-supervised learning is such that the training data contains both labelled and unlabelled data while unsupervised learning is such that none of the data is labelled and the algorithm is expected to assign them to their different classes. In recent times, the task of automatic text classification is being extensively studied and rapid progress is being recorded in this area, including the machine learning approaches such as Bayesian classifier, Decision Tree, K-nearest neighbour (KNN), Support Vector Machines(SVM), Neural Networks(NN), Rocchio's [2].

3.0 Review of Related works

This section reviews some relevant researches that have been done on text classification tasks, and recent works on combining algorithms shows that combining algorithms yield better results when compared to using individual algorithms.

[12] proposed a model which combines NB with modified maximum entropy classifier the two algorithms can be combined linearly by using its average, maximum or harmonic mean for classification of documents. They reported that the combination of the algorithms performs better than the individual algorithms.

[13] applied SVM on reuter datasets using different combinations of training and test sets and discovered that the higher the number of training data the better the classification accuracy gotten.

[14] hybridized KNN and SVM in order to reduce parameters considered in classification as inappropriate parameter values are known to degrade classification accuracy. SVM is used to reduce the training samples to their support vector which is then used as training data for KNN. The proposed method improved the classification accuracy but increased the classification time.

SVM and NB classifiers for text categorization with wikitology as knowledge enrichment was compare by [15]. Using the 20 Newsgroup dataset, the authors evaluated the two algorithms using micro-average f-measure and macro-average f-measure. Compared to baseline results, SVM shows

ISSN 1512-1232
an improvement of +6.36% while NB shows an improvement of +28.78%, this shows that both classifiers are improved when information extracted from wikitology is integrated and that NB classifier performs better when the documents are enriched from an external database.

[16] presented a study which builds a classification model by combining constrained one pass clustering algorithm and KNN text categorization. The datasets used for their experiment are Reuters-21578, Fudan university text categorization corpus and Ling-Spam corpus. They used the clustering algorithm to compress and discover complex distribution of the training texts and the text documents are now classified based on the cluster vectors instead of original text samples by using KNN. This improved model is more effective and efficient than KNN and has significant performance and good generalization ability when compared with NB, and SVM, it can also be incrementally updated which increases its applicability.

[5] proposed a method of finding multi-label categorization using SVM with membership function, Data mapping was performed to transform data from a high-dimensional space to a lower-dimensional space with paired SVM output values, thus lowering the complexity of the computation. A pairwise comparison approach was applied to set the membership function in each predicted class to judge all possible classified classes. They compared their proposed model with several multi-label approaches which are Naïve Bayes, Multi-Label Mixture, Jaccard Kernel and Bp-MLL with their proposed method found to be better than these other ones in terms of overall performance indices.

[17] performed experiments on text categorization and compared SVM with KNN and NB on binary classification tasks and concluded that SVM is not a clear winner in terms of performance as KNN compares favourably with suitable pre-processing and that NB also achieves good performance.

4.0 CLASSIFICATION ALGORITHMS

Naïve Bayes

Naive Bayes classifier is a simple Statistical Bayesian Classifier (Duda & Hart, 1973). Referred to as Naïve Bayes because it assumes that all variables combine towards classification and are mutually correlated. This assumption is called class conditional independence (Friedman, 1997). It is also called Idiot's Bayes, Simple Bayes, and Independence Bayes. They can predict class membership probabilities, such as the probability that a given data item belongs to a particular class label. A Naive Bayes classifier considers that the presence (or absence) of a particular feature (attribute) of a class is unrelated to the presence (or absence) of any other feature when the class variable is given.

The Naive Bayes Classifier is based on Bayesian Theorem and used when the input dimensionality is high. Bayes Theorem is stated below: Let X be a data sample whose class label is not known and let H be some hypothesis, such that the data sample X may belong to a specified class C. Bayes theorem is used for calculating the posterior probability.

Where

$P(C/X)$, is the posterior probability of target class.

$P(C)$, is called the prior probability of class.

$P(X/C)$. is the likelihood which is the probability of predictor of given class.

$P(X)$, is the prior probability of predictor of class.

$$P(C|X) = \frac{P(X|C).P(C)}{P(X)}$$

Multinomial Naïve Bayes is a model of NB that not only captures the presence or absence of words as in the ordinary NB classifier, but also captures the frequency of a word in a document [20], Multinomial Naïve Baye is more suitable for text classification as it performs better when the vocabulary size is relatively large as is usually the case of text datasets [20].

ISSN 1512-1232

Support Vector Machines

Support vector machines classification method which bases its theory on the Structural Risk Minimization principle from computational learning [21]. What Structural Risk Minimization does is to find a hypothesis that guarantees the lowest true error. SVM needs both positive and negative training set which is not common for other classification methods. SVM uses the positive and negative training set to seek for the decision surface that best separates the positive from the negative data in the n-dimensional space, so called the hyper plane. The support vectors are the documents representatives closest to the decision surface[7]. SVM classification performance is not affected if documents not belonging to the support vectors are removed from the data used for training [22]

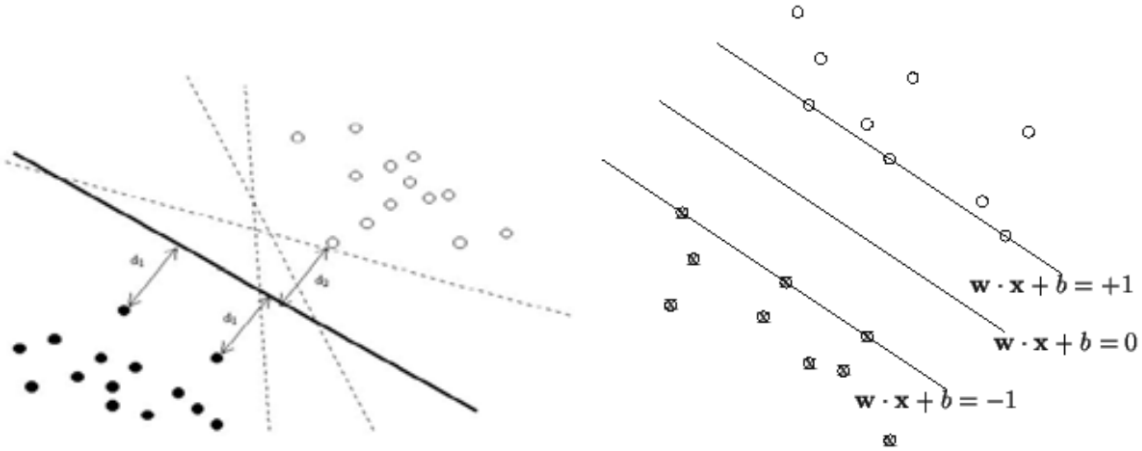

Figure 2: Optimal separating hyper plane, hyper planes and support vectors (Baharudin *et al*., 2010)

SVM can deal with documents having input space of high-dimensionality and remove irrelevant features, although some of the drawbacks with SVM is its relatively complex training and categorizing algorithms, also its memory consumption and time usage while training and during classification is also high [7].

5.0 Experimental Evaluation

A. Datasets

The text datasets used in this research are both publicly available in the University of California, Irvine repository [http://kdd.ics.uci.edu].

1. Sentiment Labelled Instances

The dataset contains 3000 instances of texts that have been labelled as either positive or negative sentiment, the texts are extracted from movies, products and restaurants reviews. The sentences are gotten from three websites which are imdb.com, amazon.com and yelp.com, each of the websites contributes 500 positive sentences and 500 negative sentences respectively. In collating the dataset, it was ensured that the statements are clearly positive or negative so as not to have neutral statements.

2. SMS Spam Collection

The SMS Spam Collection is a public set of SMS labelled messages that have been collected for mobile phone spam research. The dataset consists of SMS messages which are classified as either spam or ham. The dataset contains 5574 sentences out of which 4827 are ham messages and 747 are spam messages.

B. Preprocessing and Parameters Tuning

1. Pre-Processing

All the datasets were converted to “arff” format; this is one of the required format that WEKA software can operate on. In tokenizing the documents the following characters were removed

ISSN 1512-1232
\r\t.,;:\'\"()?!@#$\%^&*()_+ |\\?><-=[]{}[/\'~`\"- and so were not part of the letters making up the words considered by the algorithms in making classification decisions. The documents were normalized and tokenized into words and all documents were converted to lowercases.

2. Parameters Tuning

SVM: Experiments were conducted on all datasets and the linear kernel was found to give a better result as compared to other kernels available in WEKA on all datasets except the mini-newsgroup dataset in which the radial basis function (RBF) kernel had a slightly higher accuracy but still the same with the linear kernel at a 1% level of statistical significance. Also from literature, Linear kernel is said to be the best when there is a large number of instances and features as we have in text data [23], [24].

MNB: No parameter was tuned for the Multinomial Naïve Baye algorithm.

C. METHODOLOGY

The system is an ensemble of SVM and NB, The ensemble method used is stacking and linear regression was used to combine the two algorithms. The document representation used was Bag of Words model and the feature selection technique used is Term frequency and inverse document frequency (TF-IDF) which was chosen because of its efficiency in effectively selecting important words that help in better classification. The data mining software used for carrying out this research is "WEKA" – (Waikato Environment for Knowledge Analysis) tool. The classification process consists of the following stages:

1. Pre-process the data
2. Feature selection
3. Apply individual classification algorithms
4. Combine classification algorithms
5. Evaluate results

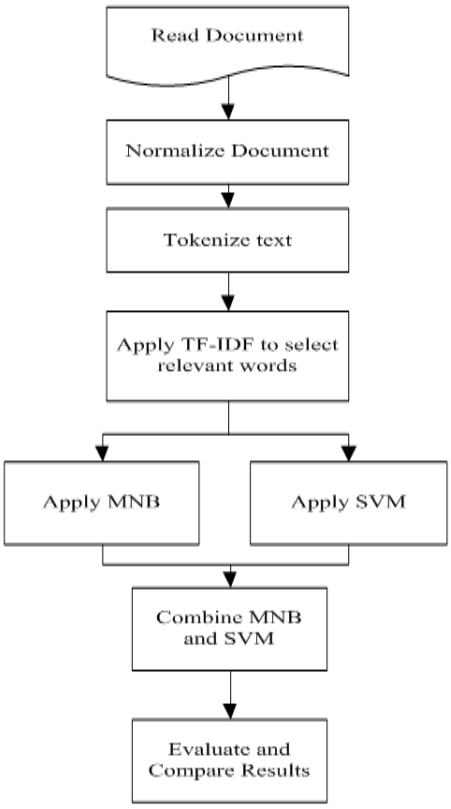

Figure 3: System Architecture

ISSN 1512-1232

D. Results and Discussions

Table 1: Performance Evaluation on datasets

	Sentiment Labelled (3000 instances)			SMS Spam (4474 instances)		
	MNB	SVM	Ensemble	MNB	SVM	Ensemble
Correctly Classified Instances	2521	2368	2535	5481	5482	5494
Incorrectly classified instances	479	632	465	93	92	80
Kappa Statistics	0.68	0.58	0.69	0.93	0.93	0.94
Time taken to build model (seconds)	0.01	10.78	82.01	0.02	6.92	53.14

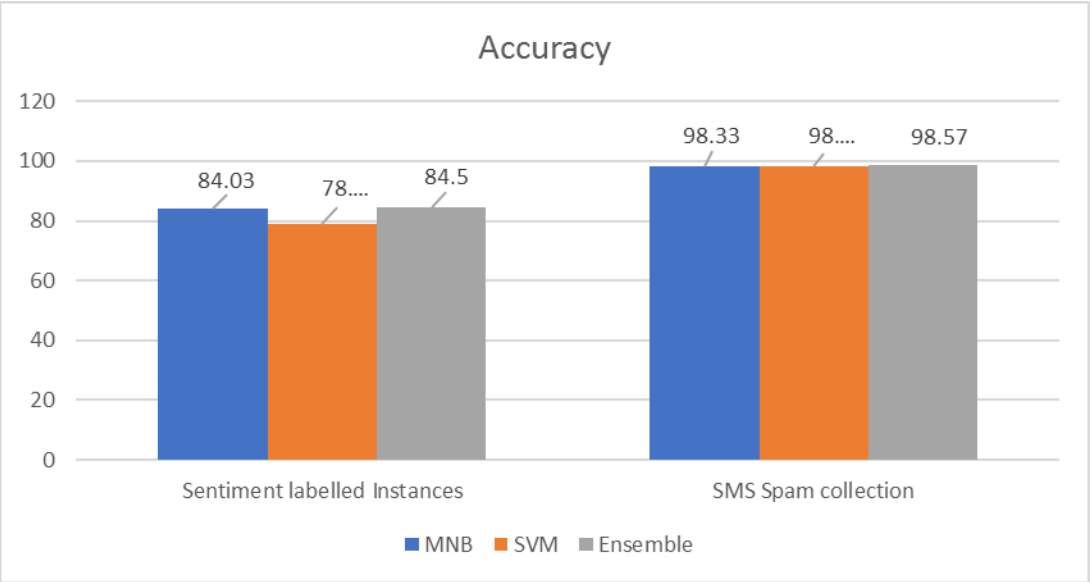

Figure 4: Accuracy on the two datasets

While MNB is better than SVM on the sentiment labelled instances and SVM performed slightly better than MNB by classifying one instance correctly than MNB in the SMS spam collection, combining both MNB and SVM ensured more instances are classified correctly than each of the algorithms individually, thus increasing the accuracy though slightly and at the expense of the time taken to build classification model. Their kappa statistics shows that on the Sentiment labelled instances, all algorithms performed substantially better than chance with The ensemble having the best performance while on the SMS spam dataset, they all performed almost perfectly better than chance with the ensemble also having the best performance which shows that the classification of the ensemble is more confident than the individual classifiers.

ISSN 1512-1232

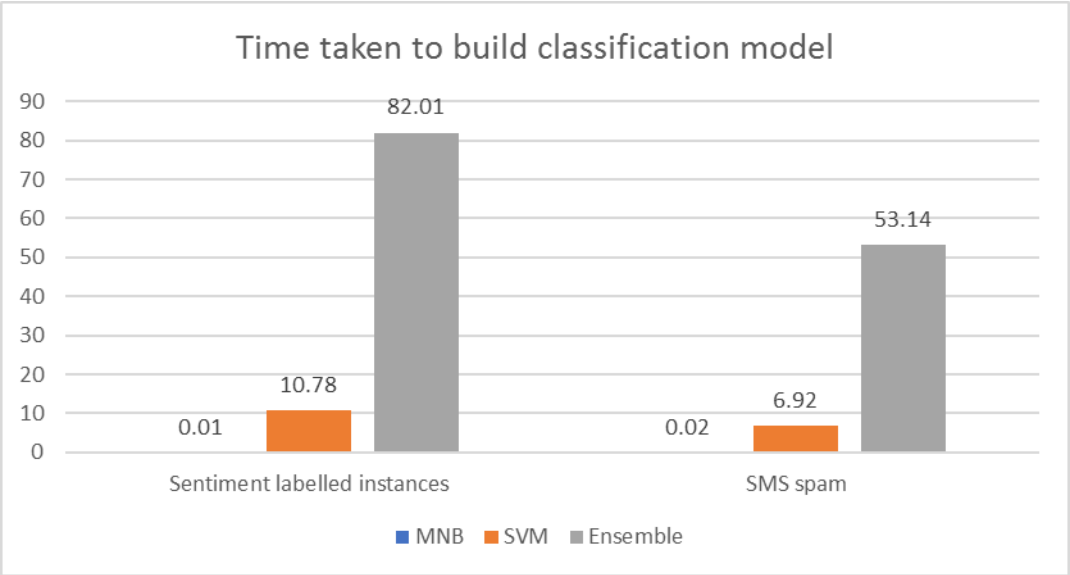

Figure 5: Time taken to build classification model

As shown in figure 4, MNB takes the fastest time in building its classification model as compared with SVM, but the ensemble model in both cases takes much more time in building its classification model in comparison with the individual algorithms.

6.0 CONCLUSION

This research shows that the combination of MNB and SVM algorithms produces a higher accuracy with more confidence in classification than using individual algorithms on performing binary text classification tasks. Observation also shows that MNB is very fast in building its classification model and would be preferred in real time binary classification situations but in cases where any increase in accuracy is very important and the added time taken by ensemble model can be overlooked, using an ensemble of algorithms is preferred.

References

[1] A. Zelaia, I. Alegria, O. Arregi, and B. Sierra, "A multiclass/multilabel document categorization system: Combining multiple classifiers in a reduced dimension," *Appl. Soft Comput. J.*, vol. 11, no. 8, pp. 4981–4990, 2011.

[2] V. Korde and C. N. Mahender, "Text Classification and Classifiers: A Survey," *Int. J. Artif. Intell. Appl.*, vol. 3, no. 2, pp. 85–99, 2012.

[3] R. Jindal, "Techniques for text classification : Literature review and current trends," *Webology*, vol. 12, no. 2, pp. 1–28, 2015.

[4] M. Ikonomakis, S. Kotsiantis, and V. Tampakas, "Text classification using machine learning techniques," *WSEAS Trans. Comput.*, vol. 4, no. 8, pp. 966–974, 2005.

[5] T. Wang and H. Chiang, "Solving multi-label text categorization problem using support vector machine approach with membership function," *Neurocomputing*, vol. 74, no. 17, pp. 3682–3689, 2011.

[6] Bhumika, S. Sehra, and A. Nayyar, "A Review Paper on Algorithms Used for Text Classification," *Ijaiem*, vol. 2, no. 3, pp. 90–99, 2013.

[7] B. Baharudin, L. H. Lee, and K. Khan, "A Review of Machine Learning Algorithms for Text-Documents Classification," *J. Adv. Inf. Technol.*, vol. 1, no. 1, pp. 4–20, 2010.

[8] A. . Fallis, "Text Categorisation: A Survey," *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–

ISSN 1512-1232
1699, 1999.
[9] A. Dasgupta, P. Drineas, B. Harb, V. Josifovski, and M. W. Mahoney, “Feature selection methods for text classification,” *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discov. data Min.*, pp. 230–239, 2007.
[10] V. Kumar and S. Minz, “Feature Selection : A literature Review,” *Smart Comput. Rev.*, vol. 4, no. 3, pp. 211–229, 2014.
[11] L. Ladha and T. Deepa, “Feature selection methods and algorithms,” *Int. J. Comput. Sci. Eng.*, vol. 3, no. 5, pp. 1787–1797, 2011.
[12] A. Jain and R. D. Mishra, “Text Categorization: By Combining Naive Bayes and Modified Maximum Entropy Classifiers,” *Int. J. Adv. Electron. Comput. Sci.*, pp. 122–126, 2016.
[13] F. Shugufta and B. Srinivasu, “Text Document Categorization using Support Vector Machine,” *Int. Res. J. Eng. Technol.*, vol. 4, no. 2, pp. 141–147, 2017.
[14] M. Sivakumar, C. Karthika, and P. Renuga, “A Hybrid Text Classification Approach Using KNN And SVM,” *Int. J. Innov. Res. Sci. Eng. Technol.*, vol. 3, no. 3, pp. 1987–1991, 2014.
[15] S. Hassan, M. Rafi, and M. S. Shaikh, “Comparing SVM and Naive Bayes classifiers for text categorization with Wikitology as knowledge enrichment,” in *Proceedings of the 14th IEEE International Multitopic Conference 2011, INMIC 2011*, 2011, pp. 31–34.
[16] S. Jiang, G. Pang, M. Wu, and L. Kuang, “An improved K-nearest-neighbor algorithm for text categorization,” *Expert Syst. Appl.*, vol. 39, no. 1, pp. 1503–1509, 2012.
[17] F. Colas and P. Brazdil, “Comparison of SVM and some older classification algorithms in text classification tasks,” *IFIP Int. Fed. Inf. Process.*, vol. 217, pp. 169–178, 2006.
[18] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley and Sons Inc., 1973.
[19] J. H. Friedman, “On bias, variance, 0/1 - loss, and the curse-of-dimensionality. Data Mining and Knowledge Discovery,” pp. 55–77, 1997.
[20] A. McCallum and K. Nigam, “A Comparison of Event Models for Naive Bayes Text Classification,” *AAAI/ICML-98 Work. Learn. Text Categ.*, pp. 41–48, 1998.
[21] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 2000.
[22] B. Heide, K. Gerhard, and M. Marc-andré, “Document Classification Methods for Organizing Explicit Knowledge,” in *Proceedings of the Third European Conference on Organizational Knowledge, Learning and Capabilities*, 2002, pp. 1–26.
[23] P. D. Shahare and R. N. Giri, “Comparative Analysis of Artificial Neural Network and Support Vector Machine Classification for Breast Cancer Detection,” *Int. Res. J. Eng. Technol.*, pp. 2114–2119, 2015.
[24] R. Mccue, “A Comparison of the Accuracy of Support Vector Machine and Nave Bayes Algorithms In Spam Classification,” Santa Cruz, CA, 2009.

Article received: 2017-09-07