International Journal of Computer Applications (0975 - 8887)
Volume 180 - No.31, April 2018

A Comparative Study on using Principle Component Analysis with different Text Classifiers

D. A. Eisa
Mathematics Department
Faculty of Science, Assiut University
New Valley, Egypt

Ahmed I. Taloba
Information System Department
Faculty of Computers and Information,
Assiut University
Assiut, Egypt

Safaa S. I. Ismail
Mathematics Department
Faculty of Science, Assiut University
New Valley, Egypt

ABSTRACT

Text categorization (TC) is the task of automatically organizing a set of documents into a set of pre-defined categories. Over the last few years, increased attention has been paid to the use of documents in digital form and this makes text categorization becomes a challenging issue. The most significant problem of text categorization is its huge number of features. Most of these features are redundant, noisy and irrelevant that cause over fitting with most of the classifiers. Hence, feature extraction is an important step to improve the overall accuracy and the performance of the text classifiers. In this paper, we will provide an overview of using principle component analysis (PCA) as a feature extraction with various classifiers. It was observed that the performance rate of the classifiers after using PCA to reduce the dimension of data improved. Experiments are conducted on three UCI data sets, Classic03, CNAE-9 and DB-World e-mails. We compare the classification performance results of using PCA with popular and well-known text classifiers. Results show that using PCA encouragingly enhances classification performance on most of the classifiers.

Keywords

Text Categorization, Dimension Reduction, Feature Extraction, Principle Component Analysis, Classifiers

1. INTRODUCTION

Most text analysis, such as text categorization (TC), includes an essential step of feature extraction to find the best set of features that assimilate each text. Text categorization is one of the central problems in text mining and information retrieval, where it is the task of classifying documents by the words of which the documents include. Several machine learning algorithms have been developed for text classification, e.g.: decision tree (J-48) [1], k-nearest neighbor (KNN) [2], support vector machine(SVM) [3] and random forests (RF) [4]. Thus, these text classifiers give acceptable accuracy with high dimensional data such as text.

There are many applications of text categorization such as topic detection [5], phishing email detection[6], author identification [7]and etc.

In text categorization, a text or a document is always represented as a bag of words. The high dimensionality of the feature space emerged as a critical problem due to this representation. This huge number of features in the feature vector results in time complexity [8] and poor performance of the classifier, so the number of input variables have to be reduced before applying a text categorization algorithm. The reduction of the feature space makes the training faster, improves the accuracy of the classifier by removing the noisy features and avoid overfitting.

The dimensionality reduction in text categorization can be made in two different ways: feature selection and feature extraction. In feature selection techniques, the most relevant variables are kept from the original data set, where as in feature extraction techniques, the original vector space is transformed into a new one with some special characteristics and the reduction is made in a new vector space. From the most popular feature extraction techniques are principle component analysis (PCA) [9][10], latent semantic indexing[11], clustering methods[12][13] and etc.

Among these many methods, PCA has attracted a lot of attention. PCA is a tool to reduce feature vector to lower dimension while retaining most of the information. It has been used since the early 90s in text processing tasks [12][14]. PCAs key advantages are its low noise sensitivity, reduce the need for capacity and memory, does not need large computations and increased efficiency given the classifiers taking place in a smaller dimensions [15].

In the current study, first, the documents are processed with the following steps; initially the documents are collected, followed by pre-processing, indexing, feature extraction, classification algorithms and performance measure. After the documents are processed, the documents are converted to a huge matrix which we have to reduce and that we do after indexing, where we use PCA as a feature extraction. However, PCA obtain a good feature subset in a less time cost. Then, after feature extraction step, the extracted features will be passed to different classifiers such as random forest (RF), support vector machine (SVM), decision tree (J-48) and k-nearest neighbours (KNN). Finally, the effectiveness of each classifier is computed; however, sensitivity, specificity and

accuracy are mostly used. Experiments are conducted on three UCL data sets [16], which are Classic03, CNAE-9 and DBWorld e-mails collection for text classification.

2. PREVIOUS WORKS

In [17] they try to speed feature extraction by using a method that folds together Unicode conversion, forced lowercasing, word boundary detection, and string hash computation. The method they found require less computation and less memory to find the integer hash features result in classifiers with equivalent statistical performance to those built using string word features.

In [18] they modulate the classifier performance. They use two-stage feature selection and feature extraction methods. In the first stage, according to the importance for classification each term within the document is ranked using the information gain (IG) method. In the second stage, the dimension reduction step, they use genetic algorithm and principle component analysis for feature selection and feature extraction after ranking the terms in descending order of importance.They apply their model on two UCI data sets, Reuters-21,578 and Classic03 which are collected for text categorization. The experimental results show that their proposed model actualizes high classification effectiveness as measured by precision, recall and F-measure.

In [19] they try to solve the high dimension of the feature vector problem for text categorization. They use a multistage model to enhance the overall accuracy and the performance of classification. In the first stage, the documents are processed and each document is represented by a bag of words. In the second step each term within the documents are ranked according to their importance for classification using the information gain (IG). Then the third stage is the attribute reduction step based on rough set which is carried out on the terms which are ranked according to their importance. Finally the extracted features are then passed to naive bayes and KNN classifier. They apply their model on three UCI data sets, Reuters-21578, Classic04 and Newsgroup 20.

In [6] they try to solve a critical text classification application, phishing email detection. They use two feature selection techniques - chi-square, information gain ratio and two feature extraction techniques principal component analysis, latent semantic analysis are used for extracting the features that improve the classification accuracy. The data set used is prepared by collecting a group of e-mails from the well known publicly available corpus that most authors in this area have used. Phishing data set consisting 1,000 phishing emails received from November 2004 to August 2007 provided by Monkey web site and, 1,700 Ham email from Spam Assassin project.

In [20] a term frequency (TF) with stemmer-based feature extraction method is proposed for document classification. The classification accuracy was calculated using J-48 classification algorithm. The effectiveness of proposed method was investigated and compared against well known other feature extraction techniques.

3. THE PROPOSED METHOD

The stages in text categorization (TC) we will follow in our work is composed of the following steps:

3.1 Documents Collection

This is the initial step of any text categorization process in which documents of several format like .html, .pdf, .doc, web content and etc. are collected.

3.2 Preparing the Text for Classification

Pre-processing of documents is an essential task during text categorization process before using the classifier in order to transform documents, which typically are strings of characters, into a set of words, which is a suitable representation for the learning algorithm, and at the same time enriching their semantic meaning. Documents filtering and stemming are applied in the extracted bag of words we have to reduce the dimensionally.

3.2.1 Stop Words Removal. Words such as pro-nouns, prepositions and articles, etc. in texts does not affect on the meaning of the documents. These words are called stop words. The removing of these words from the bag of words is an essential step such that these words are useless for purposes of retrieval. The stop words are not measured as a keywords in text mining applications [21]. Example for stop words: ”the”,”a”, ”an”, ”with”, etc. Removing stop words reduces the term space such that these words make the text look heavier and less important for analysts.

3.2.2 Stemming. This step is used to identify the words to its root. For example, extracted, extracting, extracts, extraction all can be stemmed to the word ”extract” [22]. The aim of this step is to eliminate various suffixes, to decrease the number of words, to have accurately matching stems, to save time and memory space.

3.3 Indexing

After the pre-processing step, each document is represented by a bag of words. We have to find the technique that create a vector representation of each document. So this step decrease the complexity of the documents and make them easier to handle. Each word is assigned a weight based on its number of times it appears in the document as shown in the following matrix. This process is known as term weighting.

$$\begin{pmatrix} T_1 & T_2 & .. & T_{at} & C_i \\ D_1 & W_{11} & .. & W_{t1} & C_1 \\ D_2 & W_{12} & .. & W_{t2} & C_2 \\ . & . & .. & . & . \\ . & . & .. & . & . \\ D_n & W_{1n} & .. & W_{tn} & C_n \end{pmatrix}$$

Where W_{in} is the weight of word *i* in the document *n*. We have to know that there are several ways of calculating weight, such as boolean weighting, word frequency weighting, TF-IDF, entropy, etc. A commonly used term weighting method is the so-called TF-IDF (term frequency - inverse document frequency) weighting, which is a numerical statistic measure used to reflect how important a word is to a document in a certain class of collection or corpus.

The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. For calculating the TF-IDF weight of a term in a particular document, it is necessary to calculate: term frequency (TF(t,d)) which is the number that the word *t* occurred in

the document *d*, document frequency (DF(t)) is number of documents in which the term *t* occur at least once and inverse document frequency (IDF) that can be calculated from document frequency using the following formula

$$IDF = \log\left(\frac{num\ of\ documents}{num\ of\ documents\ with\ word\ i}\right) \quad (1)$$

The inverse document frequency of a word is low if it occurs in many documents and is high if the word occurs in only few documents. The measure of word important can be calculated by using the product of the term frequency and the inverse document frequency (TF * IDF).

3.4 Feature Extraction

After the preprocessing and indexing step we have a matrix of high dimension. Having too many features cause many problems such as overfitting, reduce the accuracy of the classifier and cause high time complexity. Feature extraction will reduce the matrix size and this will improve the scalability, efficiency and accuracy of the classifiers. The main idea of feature extraction is to reduce the features dimension by creating new combinations of attributes (words). Choosing right features and deciding how to encode them to be an input for the classifier can have an enormous impact on the classifier ability to extract a good model.

Principle component analysis (PCA) is one of the most popular statistical technique for feature extraction.PCA help to improve the discriminative power of the classifiers. PCA is a useful statistical technique that many applications can be used it, such as face recognition and image compression, and is a popular technique for finding patterns in data of high dimension without losing important information such as text categorization. It is used to project the original feature space onto a lower dimensional subspace. The key idea in PCA is to find a subset of variables from a larger set, based on which original variables have the highest correlations with what is called principal components [15]. The number of the principle components is less than or equal to the number of the original features. It is an acceptable choice for reducing the dimensionality of highly dimensional data.

3.5 Text Classifiers

There are many classifiers that have been developed for variety of tasks in text classification and they give acceptable accuracy. Among them we will use the following classifiers and show how the accuracy improved after using PCA as a feature extraction:

Random Forest (RF): RF is a very good, powerful, robust and versatile learning technique, however it is a promise choice for high-dimensional text data. It is introduced in 2000s [4], it is a popular classification method which builds multiple decision trees (not only one), which are used to determine the final outcome. For classification problems, the ensemble of simple trees vote for the most popular class. One of the most known forest construction procedures, proposed by Breiman, is a subspace of features which are chosen randomly at each node to grow branches of the decision trees, then bagging method is used to generate training data subsets for building individual trees, finally combination of all individual trees are formed to form random forests model [4].

Support Vector Machine (SVM): SVM has been recognized as one of the most effective text categorization method. It gives high classification accuracy especially in highly dimensional data such that it controls complexity and overfitting issued. The time taken for each process is less than the other classifiers. So that it becomes an acceptable choice for large data set as textual data. SVMs are designed to handle high-dimensional data. SVM was developed in 1995 by Cortes and Vapnik [23]. Its core idea behind SVM is to find an optimal hyper plane between sets of hyper plane that maximize hyper plane margin, which is the distance from hyper plane to the nearest point of the pattern [23] [24]. The document representatives which are closest to the decision surface are called the support vectors. SVM is primarily used to maximize the margin, which will ensure that the input pattern would be classified correctly [25].
The aim of SVM is to find out the best possible classification function in order to differentiate between members of two classes in the training data in a two-class learning task.

Decision Tree Algorithm (J-48): The decision tree reconstructs the manual categorization of training documents by making well defined true/false-queries in the form of a tree structure. In the decision tree structure, leaves represent the corresponding category of documents and branches represent conjunctions of features that lead to those categories. The tree expands until each and every text is categorized correctly or incorrectly. The well-organized decision tree can easily categorize a document by putting it in the root node of the tree and let it run through the query structure until it reaches a certain leaf which represents the goal for the classification of the document. The decision tree classification method have several advantages over other decision support tools. The main advantage of decision tree is that it is easy in understanding and interpreting, even for non-well rounded users. Also, they are robustness to noisy data and they have the ability to learn disjunctive expressions seem suitable for text categorization. The major drawback of using a decision tree is over fits the training data with the occurrence of an alternative tree that categorizes the training data worse but would categorize the documents to be categorized better. One of the most well known decision tree algorithm is J-48 that we will use in our work.

K-Nearest Neighbor (KNN): The KNN is one of the simplest lazy classification algorithm [26] [27] and it is also well known as instance-based learning. The KNN classifier is based on the assumption that the classification of an instance is most similar to the classification of other instances that are nearby in the vector space. To categorize an unknown document d, the KNN classifier ranks the documents among training set and tries to find its k-nearest neighbors, which forms a neighborhood of d. Then majority voting among the categories of documents in the neighborhood is used to decide the class label of d. KNN is an instance-based where the function is only approximated locally and all computation is adjourned until classification. KNN is used in many applications because of its effectiveness, non-parametric and it is easy to be implemented. However, when we use KNN, the classification time is very long and it is not easy to find optimal value of k. Generally, the best alternative of k to be chosen depends on the data. Also, the effect of noise on the classification is reduced by the larger values of k but make boundaries between the classes less distinct. By using various heuristic techniques, a good 'k' can be selected.

3.6 Performance Measures for Text Classification

This is the final stage in which the effectiveness of PCA with different text classifier is evaluated. There are different criteria can be used for measuring the performance evaluation of our data sets.

In our study we will use the following criteria; confusion matrix, classification accuracy, analysis of sensitivity, specificity and F-measure.

3.6.1 Confusion Matrix. The confusion matrix consist of four classification performance indices: true positive, false positive, false negative, and true negative as given in Table 1. They are also usually used in the classification problem to evaluate the performance.

Table 1. The four classification performance indices of the confusion matrix.

Pretdicted Class		
Actual class	Classified as *pos*	Classified as *neg*
pos	True Positive (*TP*)	False Negative (*FN*)
neg	False Positive (*FP*)	True Negative (*TN*)

3.6.2 Classification Accuracy. In our study, the classification accuracy for the data sets are calculated with the following equation:

$$Accuracy(\%) = \frac{TP + TN}{TP + FN + FP + TN} \times 100, \quad (2)$$

3.6.3 Analysis of Sensitivity, Specificity and F-measure. sensitivity, specificity and F-measure are widely used to evaluate text categorization system. Specificity is the proportion of correctly proposed document to the proposed document. Sensitivity (Recall) is the proportion of the correctly proposed documents to the test data that have to be proposed. For calculating the sensitivity and the specificity of each class we use the following equations.

$$Sensetivity(\%) = \frac{TP}{(TP+FN)} \times 100, \quad (3)$$

$$Specificity(\%) = \frac{TN}{(TN+FP)} \times 100, \quad (4)$$

F-measure is a measure of test's accuracy. It is a harmonic measure of both sensitivity and specificity. It is given by the following equation

$$F - measure(\%) = \frac{2 \times Sensetivity \times Specificity}{Sensetivity + Specificity} \times 100, \quad (5)$$

4. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we present a series of experiments for text categorization on three different standards and popular text collection to examine the performance of using principle components analysis (PCA) with different text classifiers; standard RF, standard SVM, the J-48 decision tree and KNN methods are used in our study such that they are simple and give acceptable performance measurements in text categorization. The outline of the text data we used is in Table 2, where the data is downloaded from UCI machine learning databases [16]. For the classification stages we select a *10 fold cross validation*. All experiments are run on a personal computer with the configuration of Windows 7 Operation system, 1.8 GHz CPU, 2 GB of RAM and 500 GB HDD space. To evaluate the performance of the different classifiers we use WEKA 3.8.1 software which is developed by the University of Waikato [28].

In our study, after pre-processing the data by eliminating the stop words which are worthless for classification and using porter

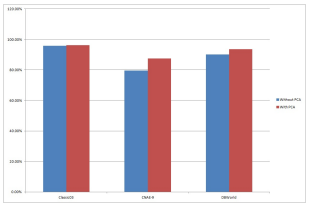

Fig. 1. Random forest classification accuracy with and without using PCA.

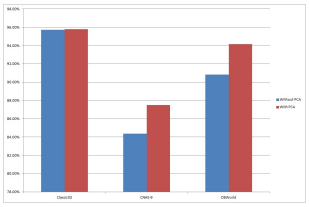

Fig. 2. Support vector machine classification accuracy with and without using PCA.

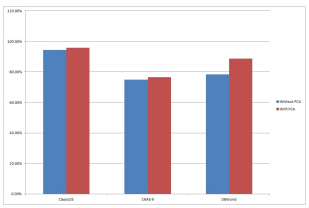

Fig. 3. Decision tree (J-48) classification accuracy with and without using PCA.

algorithm for stemming step [21], we use TF-IDF weighting scheme to reflect the important of the word to the document. This is followed by using PCA, which gives an acceptable results when dealing with text data, to reduce the huge size of the matrix we have. Then finally the extracted features are then passed to various classifiers. We first separately examine the performance of various classifiers, we have mentioned above, without using PCA for feature extraction. Then we examine these various classifiers after using PCA. The aim of using various classifiers is to compare the performance of these methods before and after using PCA.

Tables 3, 5, 7 and 9 present the results obtained for the three data sets for standard RF, SVM, J-48 and KNN classifiers respectively. However, obtained results in Tables 4, 6, 8 and 10 show that using PCA to reduce the features vector before using the standard classifiers improve the accuracy with most of the data sets compared to the standard case.

Results in Tables 3 and 4 show that when using RF as a classifier it gives acceptable and higher accuracy when compared to the rest of classifiers we use. The results in Table 4 show the improvement occur in accuracy after using PCA in the three data sets.

The results in Tables 5 and 6 show that SVM gives high accuracy with the three data sets and how the accuracy improve after using PCA with SVM.

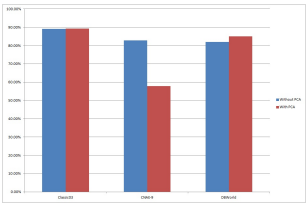

Fig. 4. K-nearest neighbor classification accuracy with and without using PCA.

Tables 7 and 8 show the performance of J-48 with our text data sets. Combining PCA and J-48 together improve the performance of J-48 with all data sets we use.
Results in Tables 9 and 10 show the performance of KNN classifier. When we compare the results in the two tables in DBworld data set we find that PCA fail to improve the performance of the KNN classifier.
With the respect to all experimental results shown above, it is seen that RF algorithm gives higher performance with text data when compared with other classifiers and its performance is improved after using PCA.

Figures 1, 2, 3, and 4 show the comparison of the classification accuracy of various classifiers we use before and after using PCA as a feature extraction technique on the different text data sets. The classification accuracy of most of the classifiers we use is improved after using PCA. Figures show that the improvement performance is much more marked with the classifier random forest (RF) when used on Classic03 data set as compared to other data sets.

5. CONCLUSION

In this study, we use principle component analysis (PCA) as a feature extraction technique to reduce the high dimensionality of the feature vector. PCA removes the irrelevant, noisy and redundant features from the feature vector and thereby improve the performance of the classifier for text categorization. First, we pre-process the documents where the feature vector is obtained through different steps like stop words removal, stemming and indexing. On the core preprocessed documents the classifiers RF, SVM, KNN and J-48 are applied, without dimension reduction, and the performance of the classifiers are observed in terms of sensitivity, specificity and F-measures. Secondly, the feature extraction method principle component analysis (PCA) is applied in the core features and the feature dimension is reduced. Then the classifiers we mentioned above are applied on the extracted features. Most of the obtained results show that the performance of most of the classifiers improved after using PCA and this seems very promising for text categorization applications.
Future scope of the work for text categorization is dimension reduction, computational time and complexity by improving different feature extraction algorithm. Also the classification performance can be improved by using different hybrid model which seems very promising for text categorization.

Acknowledgment

The authors would like to thank **Dr. Rasha Mahmoud** for many constructive technical discussions and for comments that greatly improved the manuscript. During my working on this paper, I learnt many things from her including building a system set up, provided

Table 2. Data Set Description

Data Set	No.of Doc	NO. of original Features	Classes
Classic03	3830	100	3
DBworld	64	229	2
CNAE-9	1080	856	9

Table 3. The Performance(average value of precision, sensitivity and F-measure)of RF Classifier on Three Data Sets without PCA

Data set	N. of Features	Precision	Sensitivity	F-measure	Accuracy
Classic03	100	95.9%	95.9%	95.9%	95.9%
DBworld	229	82.7%	79.7%	79.6%	79.6%
CNAE-9	856	90.4%	90.3%	90.3%	90.2%

Table 4. The Performance(average value of precision, sensitivity and F-measure)of RF Classifier with PCA

Data set	N. of Features	Precision	Sensitivity	F-measure	Accuracy
Classic03	86	96.4%	96.4%	96.4%	96.3%
DBworld	52	87.7%	87.5%	87.5%	87.5%
CNAE-9	397	93.6%	93.6%	93.6%	93.6%

Table 5. The Performance(average value of precision, sensitivity and F-measure)of SVM Classifier on Three Data Sets without Using PCA

Data set	N. of Features	Precision	Sensitivity	F-measure	Accuracy
Classic03	100	95.8%	95.7%	95.7%	95.71%
DBworld	229	84.4%	84.4%	84.4%	84.37%
CNAE-9	856	91.1%	90.8%	90.9%	90.83%

Table 6. The Performance(average value of precision, sensitivity and F-measure)of SVM Classifier with PCA

Data set	N. of Features	Precision	Sensitivity	F-measure	Accuracy
Classic03	86	95.9%	95.8%	95.8%	95.79%
DBworld	52	87.7%	87.5%	87.5%	87.5%
CNAE-9	397	94.4%	94.2%	94.2%	94.16%

Table 7. The Performance(average value of precision, sensitivity and F-measure)of J-48 Classifier on Three Data Sets without Using PCA

Data set	N. of Features	Precision	Sensitivity	F-measure	Accuracy
Classic03	100	94%	94%	94%	94.4%
DBworld	229	81.6%	75%	74.3%	75%
CNAE-9	856	78.6%	78.3%	78.4%	78.33%

Table 8. The Performance(average value of precision, sensitivity and F-measure)of J-48 Classifier with PCA

Data set	N. of Features	Precision	Sensitivity	F-measure	Accuracy
Classic03	86	95.9%	95.8%	95.8%	95.79%
DBworld	52	79.4%	76.6%	76.4%	76.56%
CNAE-9	397	90.4%	88.8%	89.1%	88.79%

Table 9. The Performance(average value of precision, sensitivity and F-measure)of KNN Classifier on Three Data Sets without Using PCA

Data set	N. of Features	Precision	Sensitivity	F-measure	Accuracy
Classic03	100	89.7%	89.2%	89.2%	89.1%
DBworld	229	86%	82.8%	82.7%	82.81%
CNAE-9	856	82.3%	81.9%	82%	81.94%

Table 10. The Performance(average value of precision, sensitivity and F-measure)of KNN Classifier with PCA

Data set	N. of Features	Precision	Sensitivity	F-measure	Accuracy
Classic03	86	90%	89.3%	89.3%	89.32%
DBworld	52	72.7%	57.8%	52.4%	57.81%
CNAE-9	379	85.4%	85%	84.8%	85%

guidance for the statistical analysis of my results, etc. I thank her very much for all her help.

6. REFERENCES

[1] N. Ur-Rahman and J. A. Harding, "Textual data mining for industrial knowledge management and text classification: A business oriented approach," *Expert Systems with Applications*, vol. 39, no. 5, pp. 4729–4739, 2012.

[2] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *Proceedings of the Fourteenth International Conference on Machine Learning*, ICML '97, (San Francisco, CA, USA), pp. 412–420, Morgan Kaufmann Publishers Inc., 1997.

[3] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," *Machine learning: ECML-98*, pp. 137–142, 1998.

[4] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[5] M. Ghiassi, M. Olschimke, B. Moon, and P. Arnaudo, "*Automated text classification using a dynamic artificial neural network model*," *Expert Systems with Applications*, vol. 39, no. 12, pp. 10967–10976, 2012.

[6] M. Zareapoor and K. Seeja, "*Feature Extraction or Feature Selection for Text Classification: A Case Study on Phishing Email Detection*," *International Journal of Information Engineering and Electronic Business*, vol. 7, no. 2, p. 60, 2015.

[7] N. Cheng, R. Chandramouli, and K. Subbalakshmi, "*Author gender identification from text*," *Digital Investigation*, vol. 8, no. 1, pp. 78–88, 2011.

[8] J. Verbeek, "*Supervised feature extraction for text categorization*," in *Tenth Belgian-Dutch Conference on Machine Learning (Benelearn'00)*, 2000.

[9] S. L. Lam and D. L. Lee, "*Feature reduction for neural network based text categorization*," in *Database Systems for Advanced Applications, 1999. Proceedings., 6th International Conference on*, pp. 195–202, IEEE, 1999.

[10] A. Selamat and S. Omatu, "*Web page feature selection and classification using neural networks*," *Information Sciences*, vol. 158, pp. 69–88, 2004.

[11] J.-T. Sun, Z. Chen, H.-J. Zeng, Y.-C. Lu, C.-Y. Shi, and W.-Y. Ma, "*Supervised latent semantic indexing for document categorization*," in *Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on*, pp. 535–538, IEEE, 2004.

[12] N. Slonim and N. Tishby, "*Document clustering using word clusters via the information bottleneck method*," in *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 208–215, ACM, 2000.

[13] A. I. Taloba, M. R. Riad, and T. H. A. Soliman, "Developing an efficient spectral clustering algorithm on large scale graphs in spark," in *Intelligent Computing and Information Systems (ICICIS), 2017 Eighth International Conference on*, pp. 292–298, IEEE, 2017.

[14] J. C. Gomez, E. Boiy, and M.-F. Moens, "*Highly discriminative statistical features for email classification*," *Knowledge and information systems*, vol. 31, no. 1, pp. 23–53, 2012.

[15] S. Karamizadeh, S. M. Abdullah, A. A. Manaf, M. Zamani, and A. Hooman, "*An overview of principal component analysis*," *Journal of Signal and Information Processing*, vol. 4, no. 03, pp. 173–175, 2013.

[16] M. Lichman, "UCI machine learning repository," 2013.

[17] G. Forman and E. Kirshenbaum, "*Extremely fast text feature extraction for classification and indexing*," in *Proceedings of the 17th ACM conference on Information and knowledge management*, pp. 1221–1230, ACM, 2008.

[18] H. Uğuz, "*A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm*," *Knowledge-Based Systems*, vol. 24, no. 7, pp. 1024–1032, 2011.

[19] H. Uğuz, "*A multistage feature selection model for document classification using information gain and rough set*," *International Journal of Advanced Research in Artificial Intelligence(IJARAI)*, vol. 3, no. 11, 2014.

[20] S. Vidhya, D. A. A. G. Singh, and E. J. Leavline, "*Feature Extraction for Document Classification*," *International Journal of Innovative Research in Science,Engineering and Technology(IJIRSET)*, vol. 4, no. 6, pp. 50–56, 2015.

[21] M. F. Porter, "*An algorithm for suffix stripping*," *Program*, vol. 14, no. 3, pp. 130–137, 1980.

[22] M. F. Porter, "*Effective PreProcessing Activities in Text Mining using Improved Porter's Stemming Algorithm*," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 2, no. 12, pp. 4536–4538, 2013.

[23] C. Cortes and V. Vapnik, "*Support-vector networks*," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[24] Z. Li, R. Yuan, and X. Guan, "*Accurate classification of the internet traffic based on the svm method*," in *Communications, 2007. ICC'07. IEEE International Conference on*, pp. 1373–1378, IEEE, 2007.

[25] C.-C. Chang and C.-J. Lin, "*LIBSVM: a library for support vector machines*," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[26] J. Wang and X. Li, "*An improved KNN algorithm for text classification*," in *Information Networking and Automation (ICINA), 2010 International Conference on*, vol. 2, pp. V2–436, IEEE, 2010.

[27] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, "*KNN model-based approach in classification*," in *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pp. 986–996, Springer, 2003.

[28] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.