ICIC Express Letters
Volume **11**, Number **12**, December **2017**
 ISSN 1881-803X
pp. **1773–1779**

A NOVEL SVM ENSEMBLE APPROACH FOR AUTOMATIC DOCUMENT CLASSIFICATION

Gang Wang

School of Management
Hefei University of Technology
No. 193, Tunxi Road, Hefei 230009, P. R. China
wgedison@gmail.com

Received June 2017; accepted August 2017

Abstract. *Support Vector Machine (SVM) is a widely used technique in automatic document classification due to its ability to efficiently handle relatively high-dimensional and large-scale datasets without decreasing classification accuracy. However, SVM still suffers from some problems, e.g., the multi-class and kernel selection. In this paper, we propose a new SVM ensemble approach, called Bagging-RS SVM, which is based on two popular ensemble strategies, i.e., bagging and random subspace and aims at building accurate and diverse classifiers. Four benchmark datasets are selected to demonstrate the effectiveness and feasibility of the proposed methods. Experimental results reveal that Bagging-RS SVM gets the best performance among the eight methods, i.e., SVM, Bagging SVM, Random Subspace SVM, Boosting SVM, DT, NB and KNN. All these results illustrate that Bagging-RS SVM can be used as an alternative technique for automatic document classification.*
Keywords: Automatic document classification, Ensemble learning, Bagging, Random subspace, SVM

1. **Introduction.** Automatic document classification is the task of building software tools to automatically assign some labels (from a set of pre-defined class labels) to a document based on some selected features of that document [1,2]. In recent years, this has become important due to the advent of large amounts of data in digital form. Especially with the rapid development of the Web, a huge amount of textual information is now accessible online. Moreover, much more textual documents are being created through Web 2.0 platforms, e.g., blogs, wikis and forums, where millions of Web users are now active information providers. These further increase the importance of automatic document classification [1-3].

Until the late 1980s, the most popular automatic document classification method was based on the knowledge engineering approach [4]. However, the main problem of such an approach is the knowledge acquisition bottle-neck; domain experts must be available and heavily consulted in designing the classification rules. In recent years, machine learning techniques have been applied to automatic document classification, including K-Nearest-Neighbor (KNN), Naive Bayes (NB), Decision Tree (DT), Support Vector Machine (SVM) and Artificial Neural Network (ANN). However, each of the above mentioned machine learning techniques has its own unique properties and associated problems. The KNN is easy to implement and shows its effectiveness in a variety of problem domains. A major drawback of the KNN is that it is computationally intensive, especially when the size of the training set grows large [5]. The NB is a relatively accurate classifier if trained using a large data set while as in many other linear classifiers, capacity control and generalization remain an issue [6]. The DT is simple to understand and interpret while it does not work well when the number of distinguishing features between documents is large [6]. The ANN is applicable to multivariate non-linear problems and does not need to

assume an underlying data distribution [7]. However, the main disadvantage of ANN is its "black box" nature and greater computational burden, especially to automatic document classification which is a high-dimensional problem. As a relatively new machine learning technique, SVM can be used as a discriminative document classifier and has shown better performance than other techniques due to its ability to efficiently handle relatively high-dimensional and large-scale data sets without decreasing classification accuracy. However, applying SVM individually to automatic document classification has also some drawbacks, e.g., multi-class and kernel selection problems.

Consequently, researchers have attempted to further enhance SVM with ensemble techniques. The performance of SVM ensemble has been investigated empirically, and it seemed to consistently give better results. In this research, we propose a new SVM ensemble approach, called Bagging-RS SVM, for automatic document classification based on two ensemble strategies, i.e., bagging and random subspace, to enhance the accuracy of automatic document classification. Among the diverse ensemble techniques that are available, bagging and random subspace are two more often used techniques and have been found to be accurate, computationally feasible across various data domain. In addition, it has been observed that an important prerequisite for ensemble techniques to reduce the test error is that it generates a diversity of ensemble members. For bagging, the only factor encouraging diversity is the proportion of different objects in the training samples. Although the classifier techniques used in bagging are sensitive to small changes in data, the bootstrap sampling appears to lead to ensembles of low diversity compared to other ensemble methods, e.g., boosting. And at the same time, document representation using vector space model in automatic document classification has high dimensional input space and few irrelevant features. Thus, we can use random subspace strategy to select a subset of relevant features as input. As a result, we introduce random subspace strategy into Bagging SVM and get Bagging-RS SVM. As there are two different factors, i.e., bootstrap selection of instances and random selection of relevant features, encouraging diversity in Bagging-RS SVM, it would be advantageous to get more accuracy. For the testing and illustration purposes, four benchmark datasets are used to verify the effectiveness of the proposed method, i.e., Bagging-RS SVM. The experimental results reveal that Bagging-RS SVM gets the best performance among the eight methods, i.e., SVM, Bagging SVM, Random Subspace SVM (RS SVM), Boosting SVM, DT, NB and KNN. All these results illustrate that Bagging-RS SVM can be used as an alternative technique for automatic document classification.

The remainder of the paper is organized as follows. In Section 2, we propose a new approach, i.e., Bagging-RS SVM, based on the bagging and the random subspace for automatic document classification. In Section 3, we present the details of experiment design. Section 4 reports the experimental results. Based on the observations and results of these experiments, Section 5 draws conclusions and future research directions.

2. **Bagging-RS SVM for Automatic Document Classification.**

2.1. **Bagging and random subspace.**

Breiman's bagging, short for bootstrap aggregating, is one of the earliest ensemble learning algorithms [8]. It is also one of the most intuitive and simplest algorithms to implement, with a surprisingly good performance. Diversity in bagging is obtained by using bootstrapped replicas of the training dataset: different training data subsets are randomly drawn – with replacement – from the entire training dataset. Each training data subset is used to train a different base learner of the same type. The base learners' combination strategy for bagging is majority vote. Simple as it is, this strategy can reduce variance when combined with the base learner generation strategies.

The random subspace method is an ensemble construction technique proposed by Ho [9]. In the random subspace, the training dataset is also modified as in bagging. However, this modification is performed in the feature space (rather than example space). The random subspace may benefit from using both random subspaces for constructing the classifiers and aggregating the classifiers. When the dataset has many redundant attributes, one may obtain better classifiers in random subspaces than in the original feature space [9]. The combined decision of such classifiers may be superior to a single classifier constructed on the original training dataset in the complete feature space.

2.2. **Bagging-RS SVM for document classification.** Great improvement in generalization performance has been observed from ensemble learning in a wide range of numerical experiments and practical applications. On the generalization ability of ensemble learning it is usually much stronger than that of a single learner, Dietterich gave three reasons by viewing the nature of machine learning as searching a hypothesis space for the most accurate hypothesis [10]. The first reason is that, the training data might not provide sufficient information for choosing a single best learner. For example, there may be many learners performing equally well on the training set. Thus, combining these learners may be a better choice. The second reason is that, the search processes of the learning algorithms might be imperfect. For example, even if there exists a unique best hypothesis, it might be difficult to achieve since running the algorithms results in sub-optimal hypotheses. Thus, ensembles can compensate for such imperfect search processes. The third reason is that, the hypothesis space being searched might not contain the true target function, while ensembles can give some good approximation. For example, it is well-known that the classification boundaries of decision trees are linear segments parallel to coordinate axes. If the target classification boundary is a smooth diagonal line, using a single decision tree cannot lead to a good result but a good approximation can be achieved by combining a set of decision trees. Although these intuitive explanations are reasonable, they lack rigorous theoretical analyses.

In practice, to achieve a good ensemble, two necessary conditions should be satisfied: accuracy and diversity. For the first condition, accuracy, we could simply mean that the base learner should be more accurate than random guessing. In this study, we use SVM as base learner which is satisfied with the above condition. For the second condition, diversity, we mean that each base learner has its own knowledge about the problem and has a different pattern of errors compared to other base learners. Focusing on diversity, there are different methods for construction of diverse base learners. For example, just as mentioned above, bagging perturbs the distribution of the training set by resampling. And random subspace perturbs the feature space to get diversity. For bagging, however, there is the only factor encouraging diversity. Although the base learner used in bagging is sensitive to small changes in data, the bootstrap sampling appears to lead to ensembles of low diversity compared to other ensemble methods, e.g., boosting. To enforce diversity, a version of bagging called Random Forest was proposed by Breiman [11]. The ensemble consists of decision trees built again on bootstrap samples. The difference lies in the construction of the decision tree. The feature to split a node is selected as the best feature among a set of M randomly chosen features, where M is a parameter of the algorithm. This small alteration appeared to be a winning heuristic in that diversity was introduced without much compromising the accuracy of the base learners.

Following this improvement, as document representation using vector space model in automatic document classification has high dimensional input space and few irrelevant features (Joachims, 1998; Leopold & Kindermann, 2002), we can select a subset of relevant features as input for base learners in Bagging SVM. This strategy is also used in random subspace which builds each base learner on a different subset of features randomly selected from the original feature set. Thus, we introduce random subspace strategy into

Bagging SVM and get Bagging-RS SVM. As there are two different ensemble strategies, i.e., bootstrap selection of instances and random selection of relevant features, encouraging diversity in Bagging-RS SVM, it would be advantageous to get more accuracy than bagging and random subspace individually. The pseudo-code for the Bagging-RS SVM algorithm is given in Figure 1.

Input: Data set $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$;
Base learning algorithm SVM;
Number of selected features rate k;
Number of learning rounds for Bagging T_B;
Number of learning rounds for Random Subspace T_R.
Process:
For $t = 1, 2, \ldots, T_B$:
$D_t = Bootstrap(D)$; % Generate a bootstrap sample from D
For $s = 1, 2, \ldots, T_R$:
$D_t^s = RS(D_t, k)$; % Random generate a subspace sample from D_t
$h_s = SVM(D_t^s)$ % Train a decision tree h_s from the sample
end.
$H_t^{RS}(x) = \arg\max_{y \in Y} \sum_{s=1}^{T_R} 1(y = h_s(x))$ % the value of $1(\alpha)$ is 1 if α is true
end. % and 0 otherwise
Output: $H(x) = \arg\max_{y \in Y} \sum_{t=1}^{T_B} 1\left(y = H_t^{RS}(x)\right)$ % the value of $1(\alpha)$ is 1 if α is true
% and 0 otherwise

FIGURE 1. The Bagging-RS SVM algorithm

3. **Experimental Design.** Four benchmark datasets are used to evaluate the performance of our proposed method. These four text classification tasks are drawn from standard benchmarks, i.e., OHSUMED and TREC. The datasets range from 6 to 17 classes, 2000 to 31472 binary features. Dataset oshcal is from OHSUMED collection subset of MEDLINE dataset, which contains 233445 documents indexed using 14321 unique categories. We took different subsets of categories to construct these datasets provided by Han and Karypis [12]. Dataset fbis is from the Foreign Broadcast Information Service data of TREC-5. Datasets la1 and la2 are from the Los Angeles Times data of TREC-5. For all datasets, we used a stop-list to remove common words, and the words were stemmed using Porter's suffix-stripping algorithm.

The evaluation criteria of our experiments are adopted from the established standard measures in the fields of text classification. Although precision and recall are two popular performance measures for text classification, neither they make sense in isolation from each other as it is well known from the IR practice that higher levels of precision may be obtained at the price of recall, and vice versa. To combine precision and recall, we use the F_1 measure as performance evaluation criteria. Usually, F_1 measure is computed from two ways, i.e., macro-averaged and micro-averaged. Macro-averaged performance scores are computed by first computing the scores for the per-category contingency tables and then averaging these per-category scores to compute the global means. Micro-averaged performance scores are computed by first creating a global contingency table whose cell values are the sums of the corresponding cells in the per-category contingency tables, and then use this global contingency table to compute the micro-averaged performance scores.

4. **Experimental Results.** In this study, we compared the performance of 8 methods, including Bagging-RS SVM, SVM, Bagging SVM, RS SVM, Boosting SVM, DT, NB and KNN ($k = 10$). For implementation of SVM, DT, NB and KNN, we chose SMO (WEKA's

own version of SVM) module, J48 (WEKA's own version of C4.5) module, NaiveByes module and IBk module in WEKA. And for implementation of ensemble learning, i.e., Bagging SVM, RS SVM and Boosting SVM, we chose Bagging module, RandomSubSpace module and ADBoostM1 module. For implementation of Bagging-RS SVM, we used WEKA Package, i.e., WEKA.JAR and implement in Eclipse. Except when stated otherwise, all the default parameters in WEKA were used. On each dataset, for each compared method, we used Information Gain (IG) as feature selection criteria. For each dataset, we follow Forman's experimental procedure [13] and chose $\{10, 20, 50, 100, 200, 500, 1000, 2000\}$ as feature size. Moreover, six subspace rates for Bagging-RS SVM are tested, where the value of k is set to 0.4, 0.5, 0.6, 0.7, 0.8 and 0.9, respectively. To minimize the influence of the variability of the training set and also follow Forman's experimental procedure [13], five times 4-fold cross validation is performed on the four datasets.

Tables 1 and 2 show the best performance comparison in Micro F_1 and Macro F_1. In addition, the experimental results on the other feature sizes and subspace rate will be presented later in this subsection. Note that for the four datasets, Bagging-RS SVM all gets best performances compared with other seven methods. For example, on the dataset fbis, the Macro F_1 of Bagging-RS SVM is 81.09%, which is approximately 24% higher than that of DT, 22% higher than that of NB, and 26% higher than that of KNN. On the dataset oshcal, the Micro F_1 of Bagging-RS SVM is 78.45%, which is approximately 9% higher than that of DT, 22% higher than that of NB, and 26% higher than that of KNN.

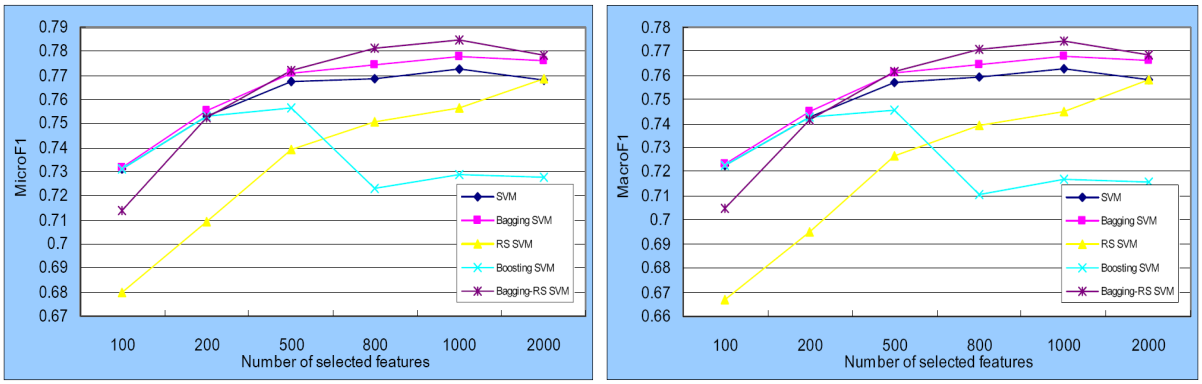
Table 1. The best Micro F_1 of different methods on four datasets

Dataset	Bagging-RS SVM	SVM	Bagging SVM	RS SVM	Boosting SVM	DT	NB	KNN
oshcal	0.7845	0.7729	0.7778	0.7689	0.7563	0.7197	0.6433	0.6220
fbis	0.8519	0.8455	0.8467	0.8408	0.8452	0.7411	0.7360	0.7487
la1	0.9033	0.8938	0.8985	0.8987	0.8938	0.7772	0.8509	0.7733
la2	0.9095	0.8969	0.9028	0.9062	0.8969	0.7878	0.8593	0.7934

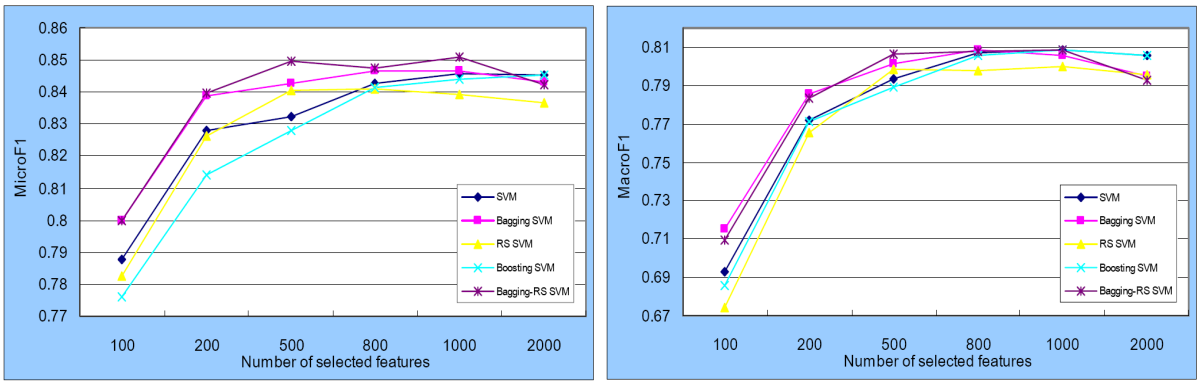
Table 2. The best Macro F_1 of different methods on four datasets

Dataset	Bagging-RS SVM	SVM	Bagging SVM	RS SVM	Boosting SVM	DT	NB	KNN
oshcal	0.7742	0.7627	0.7679	0.7582	0.7453	0.7066	0.6292	0.6103
fbis	0.8109	0.8089	0.8084	0.7996	0.8086	0.6544	0.6666	0.6427
la1	0.8845	0.8759	0.8810	0.8816	0.8759	0.7480	0.8193	0.7452
la2	0.8932	0.8811	0.8872	0.8907	0.8811	0.7584	0.8262	0.7666

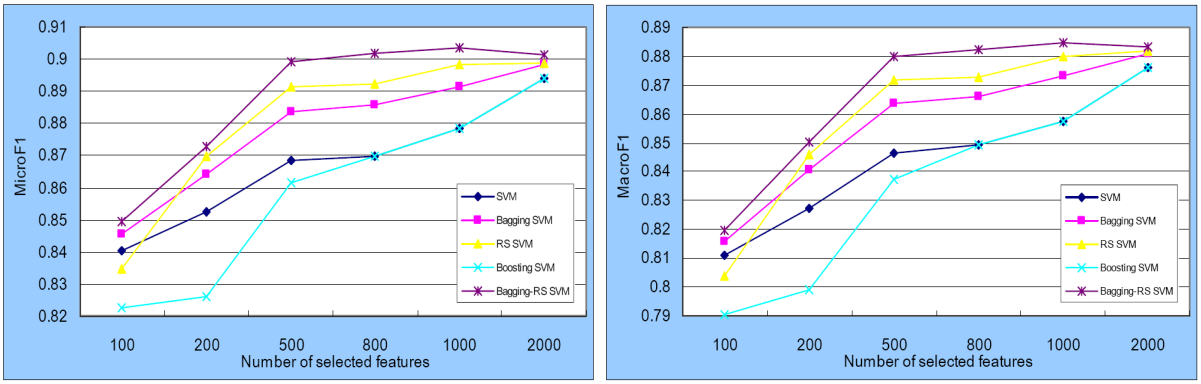
Subsequently, Figure 2 displays the Micro F_1 and Macro F_1 curve for SVM and other four ensemble SVM classifiers on four datasets. Note that for Bagging-RS SVM, the subspace rate $k = 0.7$. On all four datasets, the performance of Bagging-RS SVM exceeds all the other four methods. When the number of selected features is small, the performance of Bagging-RS SVM may be worse than Bagging SVM and RS SVM. However, with the increasing of selected features, the performance of Bagging-RS is becoming better than Bagging SVM and RS SVM. These results further prove that the combining two ensemble strategies, i.e., bagging and random subspace, can enhance the accuracy of automatic document classification. It is unexpected that Boosting SVM gets worse performance than other methods. Note that with the increasing of features, the performance of Boosting SVM comes near to that of the individual SVM except dataset oshcal.



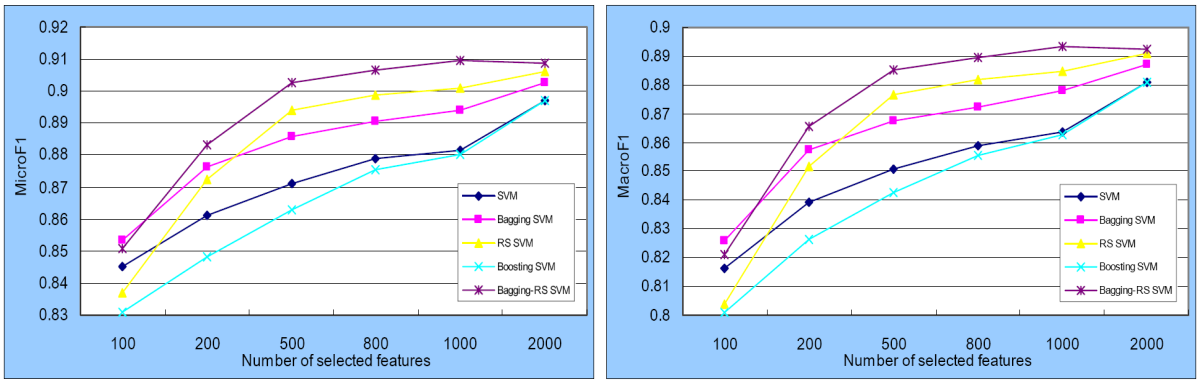
(a) oshal



(b) fbis



(c) la1



(d) la2

FIGURE 2. Performance curves of different methods at different selected features

5. **Conclusions and Future Directions.** With the exponential growth of textual information available from the Internet, there has been an emergent need to find and organize relevant information in text collections. For this purpose, automatic document classification becomes a significant tool to utilize text information efficiently and effectively. In this paper, a new SVM ensemble approach, called Bagging-RS SVM, is proposed for automatic document classification. This approach works through integration of two popular ensemble strategies, i.e., bagging and random subspace. Bagging-RS SVM outperforms bagging and random subspace in generating more diverse component SVM classifiers. The experiments on four benchmark evaluation datasets showed that Bagging-RS SVM gets the best performance among the eight methods, i.e., SVM, Bagging SVM, RS SVM, Boosting SVM, DT, NB and KNN. Based on these results, we can conclude that Bagging-RS SVM can be used as an alternative technique for automatic document classification.

Several future research directions also emerge according to this study. Firstly, large datasets for experiments and applications should be collected to further valid the conclusions of the study. Secondly, the reason why Boosting SVM gets worse performance in our experiments is not clear. In the future research, the mechanism should be studied and subsequently revised Boosting SVM can be investigated further. Thirdly, the experimental results have shown that combining different ensemble strategies can achieve better performance. Thus, more extensive combination of ensemble strategies can be investigated in the future research.

Acknowledgments. This work is partially supported by the National Natural Science Foundation of China (71471054, 91646111), Anhui Provincial Natural Science Foundation (1608085MG150), Social Science Knowledge Popularization Foundation of Anhui Province (Y2016016), Special Fund of Political Theory Research Center of Hefei University of Technology (JS2015HGXJ0051), Training Program of Application of Scientific and Technological Achievement of Hefei University of Technology (JZ2017YYPY0235).

REFERENCES

[1] N. V. Linh, N. K. Anh, K. Than and C. N. Dang, An effective and interpretable method for document classification, *Knowledge and Information Systems*, vol.50, pp.763-793, 2017.

[2] A. K. Uysal and S. Gunal, The impact of preprocessing on text classification, *Information Processing & Management*, vol.50, pp.104-112, 2014.

[3] C. Wang, Y. Song, H. Li, M. Zhang and J. Han, Text classification with heterogeneous information network kernels, *Proc. of the 30th AAAI Conference on Artificial Intelligence*, pp.2130-2136, 2016.

[4] J. Y.-H. Pong, R. C.-W. Kwok, R. Y.-K. Lau, J.-X. Hao and P. C.-C. Wong, A comparative study of two automatic document classification methods in a library setting, *Journal of Information Science*, vol.34, pp.213-230, 2008.

[5] E. Han, G. Karypis and V. Kumar, Text categorization using weight adjusted k-nearest neighbor classification, *Proc. of the 5th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp.53-65, 2001.

[6] D. Isa, L. H. Lee, V. P. Kallimani and R. Rajkumar, Text document preprocessing with the Bayes formula for classification using the support vector machine, *IEEE Trans. Knowledge and Data Engineering*, vol.20, pp.1264-1272, 2008.

[7] G. Wang, J. Hao, J. Mab and L. Huang, A new approach to intrusion detection using artificial neural networks and fuzzy clustering, *Expert Systems with Applications*, vol.37, pp.6225-6232, 2010.

[8] L. Breiman, Bagging predictors, *Machine Learning*, vol.24, pp.123-140, 1996.

[9] T. K. Ho, The random subspace method for constructing decision forests, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.20, pp.832-844, 1998.

[10] Z.-H. Zhou and M. Li, Tri-training: Exploiting unlabeled data using three classifiers, *IEEE Trans. Knowledge and Data Engineering*, vol.17, pp.1529-1541, 2005.

[11] L. Breiman, Random forests, *Machine Learning*, vol.45, pp.5-32, 2001.

[12] E. H. Han and G. Karypis, Centroid-based document classification: Analysis and experimental results, *Principles of Data Mining and Knowledge Discovery*, pp.116-123, 2000.

[13] G. Forman, An extensive empirical study of feature selection metrics for text classification, *The Journal of Machine Learning Research*, vol.3, pp.1289-1305, 2003.