Journal of Information Hiding and Multimedia Signal Processing ©2015 ISSN 2073-4212
Ubiquitous International Volume **6**, Number **2**, March **2015**

A Classifier Ensemble Algorithm Based on Improved RSM for High Dimensional Steganalysis

[1]Feng-Ying He, [2]Tian-Shun Chen, and [3]Shang-Ping Zhong

College of Mathematics and Computer Science
Fuzhou University
Min hou county, Fuzhou, China
[1]hfy@fzu.edu.cn
[2]cts@163.com
[3]spzhong@fzu.edu.cn

Received December, 2013; revised October, 2014

Abstract. *Today, ensemble learning algorithms are proposed to address the challenges of high dimensional classification for steganalysis caused by the curse of dimensionality and obtain superior performance. In this paper, we propose a classifier ensemble algorithm based on improved Random Subspace Method (RSM) for high-dimensional blind steganalysis. Firstly, sequential forward selection (SFS) algorithm is adopted to select part of features with high classification ability as fixed subset, so that the original feature space is partitioned into two parts: fixed subset and remaining feature subset, then the final feature subset is formed by selecting features randomly in each part according to the given sampling rate. Secondly, each base classifier is trained on the feature subset and the weight of each classifier is computed according to the classification accuracy and mutual information. Finally, the final decision is yielded using the weighted voting. Experiments with the steganographic algorithms HUGO demonstrate that the proposed algorithm can effectively increase the classification accuracy, in most cases, the detection accuracy is better than RSM and other classical classifier ensemble methods.*
Keywords: High-dimensional feature, Random subspace method, Ensemble classifier, Sequential forward selection, Feature selection, Mutual information

1. **Introduction.** With the increased sophistication of steganographic algorithms, steganalysis has already begun using feature spaces of increased dimensionality [1-4]. For example, the most accurate spatial domain steganalysis of embedding (LSB matching) uses the 686-dimensional subtractive pixel adjacency matrix (SPAM) features [5]. In [6] a 1,234-dimensional Cross-Domain Feature (CDF) set is employed to attack YASS. Moreover, the recent steganalysis competition BOSS [7] has constructed a 24993-dimensional HOLMES proved especially effective against HUGO.

However, classifying high dimensional features leads to the problem called as the curse of dimensionality [8]. Even though the support vector machine (SVM) seems to be the most popular machine learning tool used in steganalysis, SVM is quite restrictive due to the complexity of SVM will be increased rapidly with the dimensionality of feature space growing. Traditional approach that deals with the adverse effects of high dimensionality employs dimensionality reduction or projection technique like principal component analysis (PCA) [9] or independent component analysis (ICA) [10]. However, it is indispensable to calculate the covariance matrix of the original feature vectors in such techniques. The computational complexity is too large for high-dimensional feature vectors, so these traditional techniques are not applicable to feature vectors with thousands of dimensions.

To address the challenges associated with the curse of dimensionality arising in staganalysis, Random Subspace Method (RSM) [11] is used in [12, 13] and achieves perfect classification performance using 24993-dimensional HOLMES features. RSM is a classical ensemble learning algorithm. The method extracts low-dimensional feature subsets from the original high-dimensional feature space and trains each based classifier on the low-dimensional feature subsets. Then the final ensemble decision is made by combining individual predictions according to a certain fusion strategy. RSM significantly lowers training complexity to train base learners on subsets with low dimensions. In addition, RSM just selects a subset of the important features from the original feature space to form a new low-dimensional space, so the operation on the global features can be avoided. Recently, RSM has been successfully used in the field with high dimensionality, such as biological data [14] and face recognition [15].

However, the features in subspaces used to train the base learners in RSM are selected randomly from the original feature space, which is not suitable for high dimensional feature space consisting of thousands of features, because a randomly sampled subspace contains many features, which are uninformative to classification, thus affecting the performance of the base learners negatively. Aimed at the deficiency of RSM, many scholars have conducted related research, the major improvements are: 1) combining RSM and other ensemble learning algorithms. An improved method combining RSM and Boosting is proposed [11] and has obtained good results on UCI datasets. 2Reducing the randomness of feature subset extraction to improve the classification ability. A PCA-based RSM algorithm is proposed [16]. The method first uses PCA to remove redundant information and then uses RSM to construct optimal subspace. Experimental results show that the method has better performance. A local random subspace algorithm is proposed [17]. The method uses Simba [18] algorithm to select features and experiments demonstrate the improved algorithm outperforms the classical RSM algorithm.

To increase the performance and maintain the diversity of the base learners for RSM in high dimensional data, we propose a new method, called SFSRSM, which modifies the generation method of feature subspaces and uses the weighted voting to yield the final class predictor. Firstly, sequential forward selection (SFS) [19] algorithm is used to select part of features with high classification ability as fixed subsets. Then the original feature space is partitioned into two parts: fixed subset and remaining feature subset. Secondly, the feature subset is formed by selecting features randomly in each part according to the given sampling rate. Such a method increases the probability for informative features to be included in each subspace. As a result, the accuracy of the base learners is increased and the diversity between them is not reduced due to the random selection of features in each part. Finally, each base classifier is trained on these feature subsets and the weights of base classifiers are computed according to classification accuracy and mutual information. To improve the stability of ensemble classifier, the final decision is formed as a weighted combination of individual predictions. Experiments with steganographic algorithm HUGO [20] demonstrate the usefulness of the proposed algorithm over current popular approaches. In most cases, the detection accuracy is better than RSM and other classical classifier ensemble methods such as Bagging and AdaBoost.

The remaining part of this paper is organized as follows: The SFSRSM algorithm is described in detail in Section 2. The results of experiments are presented in Section 3 and a conclusion is given in the last Section.

2. **The Classifier Ensemble Algorithm Based On Improved RSM.** Theoretical and empirical results [21] suggest that ensemble classifier gives optimal improvements in accuracy if the base classifiers are diverse. The effective method to enhance diversity is to train base classifiers on different feature subsets [22]. Therefore, the form of feature subset is the most crucial step in this study. The main idea of the proposed SFSRSM is : 1) firstly, SFS algorithm is adopted to select part of features with higher classification ability as fixed subset; 2) the original feature space is partitioned into two parts: fixed subset (part 1) and remaining feature subset (part 2).Then each part is assigned a sampling rate, the feature subset is formed by selecting features randomly in each part according to the given sampling rate, by this way, we can guarantee good performance of base learners, while ensuring diversity;3) train base classifiers on feature subsets and calculate the weights of base classifiers by mutual information and classification performance, then the weighted voting method is used to build an effective ensemble classifier. The frame of SFSRSM algorithm is shown in Fig.1. The proposed algorithm has the following characteristics: (1) The diversity between base learners is obtained by the randomization in the feature subspace generation. (2) The classification accuracy of base classifiers is improved, because the probability for informative features to be included in each subspace is increases. It is suggested that the feature subset can contain more features with stronger classification ability to improve performance. (3) The flexibility of forming the feature subspace can be effectively enhanced by setting different sampling rate in corresponding parts to adjust dynamically the distribution of features of different parts in the random subspace for each base learner. (4) The performance of ensemble classifier can be improved by combining individual predictions with weights corresponding to the accuracy of each base learner.

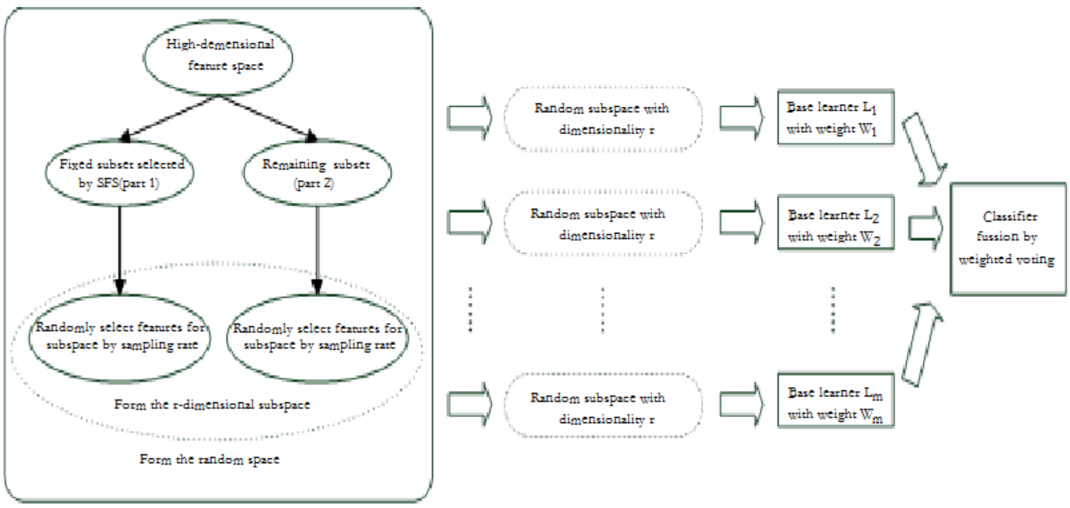

FIGURE 1. The frame of SFSRSM algorithm

2.1. **Sequential Forward Selection(SFS) Algorithm.** SFS algorithm is a kind of feature selection algorithm. Its essence is a kind of greedy selection algorithm. Suppose feature subset $X = \emptyset$, in each loop we select a feature d_i to join X , which makes the classification accuracy for the validation set V is highest. The classification accuracy is obtained by classifier trained by classification algorithm C on feature subset $X'(X' = X \cup d_i)$. SFS algorithm is described as follows:

1. Input: training set D, number of selected features S, classification method C, validation set V.
2. Initialize the feature subset X as an empty set, $X = \emptyset$.

3. **For** each feature d_i in training set D **do**
(1) **If** $d_i \notin X$ **then**
(2) $X' = X \cup d_i$
(3) Train classifier using classification algorithm C on feature subset X' and calculate classification accuracy T_i for validation set V.
(4) **End if**
4. Find out the largest classification accuracy in step (3) and add the corresponding feature d_i to feature subset X.
5. Count the number of features in X, which is represented with S'. If $S' < S$, go to step (3); Otherwise, go to step (6)
6. Output: the selected feature subset X.

2.2. **Calculation of the weights of base classifiers.** Due to the limitation of computer resources, it is difficult to select important features using all samples in a large data set. Most feature selection algorithms extract a training subset to select features and results tie up with the selected training subset, so that the performance and structure of classifier relate with training data subset.

To yield an ensemble classifier with higher stability and reliability, we combine results of base classifiers using weighted voting strategy. Refer to the idea of [23], the weights of base classifiers are calculate using mutual information and classification performance.

Zhang [23] has shown that, mutual information between two different feature subsets can depict the difference between base classifiers. So using mutual information to compute the weight of each base classifier will be useful to the ensemble result.

Denote m as the number of base learners, N as the dimensionality of the high dimensional feature space and H_j(j=1,2,...,m) as the *jth* feature subset of dimension r. Each base learner, denoted L_j(j=1,2,..., m) is trained on H_j. Suppose that H_j and H_i have n^{ij} same features. The mutual information between H_j and H_i can be defined as formula (1):

$$I(H_i, H_j) = n^{ij} r \log \frac{n^{ij} N^2}{r^3} \qquad (1)$$

The average mutual information of H_i (i=1,2,..., m)can be computed as formula (2):

$$\beta_i = \frac{1}{m-1} \sum_{j=1, j\neq i}^{m} I(H_i, H_j) \qquad (2)$$

On the other side, classification accuracy also depicts the difference between classifiers. Suppose that α_i is the accuracy of the classifier trained on H_i, so the weight of classifier L_i (i=1, 2, ... ,m) is computed as formula (3):

$$w_i = \alpha_i . \frac{1}{\beta_i} . \mathrm{Z} \qquad (3)$$

Where Z is used to normalize w_i, making $w_i > 0, \sum_{i=1}^{m} w_i = 1$.

2.3. **SFSRSM algorithm.** How to improve the performance of base classifiers, while enhancing the diversity between them is the main goal of this study. The improved algorithm is called as SFSRSM algorithm. Firstly, the SFSRSM algorithm uses SFS algorithm to select r features from the whole feature set D as fixed feature subset X (r is the dimensionality of the subspace) , then the whole set is divided into two parts: fixed feature subset X and the remaining feature subset R, $R = D - X$.Secondly, the SFSRSM algorithm randomly select the features from two parts with the given sampling rate to form the subspace for each base classifier. It randomly selects N_1 features from X and N_2 features from R to form the subspace with dimensionality r.Denote p_1 as the sampling rate in the first part X,$N_1 = \lceil p_1 \times r \rceil$, $N_2 = r - N_1$. Finally, the SFSRSM algorithm trains base classifiers on subspaces and computes the weights of base classifiers according to classification accuracy and mutual information, then gets the final decision by weighted voting.

The algorithm is described as follows:

1. Input: training set D, test sample x, classification method C, dimensionality of feature subset r, number of base classifiers m, sampling rate p, validation set V.

2. Select feature subset X with better classification ability using SFS algorithm, $X = SFS(D, r, C, V)$.

3. Divide training set D into two parts: fixed feature subset X and the remaining feature subset R, $R = D - X$.

4. For j=1 to m do

(1) Form r-dimensional feature subset H_j by selecting features randomly in X and R according to the given sampling rate p.

(2) Project D onto the feature subset H_j and obtain the training sample subset $Subtrain_j$.

(3) Obtain base classifiers L_j trained on $Subtrain_j$ with classification method C and calculate the classification accuracy α_j of L_j.

5. For j=1 to m do

Calculate the weight w_j of L_j according to the formula (3)

6. For j=1 to m do

(1) Project x onto the feature subset H_j and obtain the testing sample subset x_j

(2) Use the classifier L_j to predict x_j and obtain the prediction $L_j(x_j)$

7. Obtain the final prediction of x by weighted voting:

$$L(x) = \begin{cases} 1 & when \sum_{j=1}^{m} w_j L_j(x_j) > m/2 \\ 0 & when \sum_{j=1}^{m} w_j L_j(x_j) \le m/2 \\ random & otherwise \end{cases}$$

3. **Experiment Results and Analysis.**

3.1. **Experimental setting.**

We demonstrate the power of the proposed algorithm by applying it to the recently proposed adaptive spatial-domain steganographic algorithm called HUGO. All experiments were carried out on the database BossBase v1.00 (http://www.agents.cz/boss/BOSSFinal/)[24].This training database is made of 10000 512×512 greyscale cover images in the pgm format, and the same 10000 images embedding a message at 0.4 bpp with HUGO algorithm with default parameters. In each test, the randomly selected 1000 pairs of stego and cover images are used as validation set, the randomly selected 8000 pairs of stego and cover images are used for training and the remaining 1000 pairs of images for testing. On each image, we extract 12753-dimensional SRM feature for the experiments.

The dimensionality of the subspace r is 100,150,200,250,300,350,400,450,500,600,700,800, 900,1000,respectively. Due to the computational resource limitation, for a given subspace size we set the number of base learners m = 25. For our proposed algorithm, we set the sampling rate p={0.5,0.5},{0.6,0.4},{0.7,0.3},{0.8,0.2}, respectively. The program code is implemented using C/C + + language, the test platform is WIN7 operating system, Intel Xeon E5300 2.60GHz, 8GB Memory.

To evaluate the performance of the proposed algorithm in a fair and reasonable way, the proposed algorithm is compared with three different types of representative ensemble algorithms. They are Bagging, AdaBoost and RSM. For each algorithm, we use C4.5,1NN,L-SVM and FLD as base classifier respectively. C4.5 and 1NN are set the default configuration in Weka3.4, the training parameters of L-SVM are obtained using five-fold cross-validation to search over the grid of the cost parameter C$\in \{10^{a}\} \quad \alpha \in \{-4, ..., 3\}$.

3.2. **Comparison of the accuracy of the algorithms.** In order to show the advantages of the new generation method of feature subset, we compare our proposed method and the RSM algorithm. For each algorithm, there are totally 14 sets of experiments carried out in this paper. To make the results statistically reliable, each set of experiment has been repeated for 10 times independently to take the average detection accuracy and its variance as the final results, as shown in Table1∼Table4. The best results are marked in bold. The average result is shown in the last line.

(1) From the results shown in Table1∼Table4, it can be seen that, in most cases, the detection accuracy of the proposed SFSRSM algorithm is better than that of RSM. When base learner is C4.5, L-SVM, FLD, respectively, the proposed SFSRSM gets optimal results in 10 out of 14 subspaces, in 13 out of 14 subspaces, in 11 out of 14 subspaces, respectively while RSM algorithm gets optimal results in the other 6 subspaces, in the other 1 subspace, in the other 3 subspaces, respectively. When base learner is 1NN, the performance of the proposed SFSRSM is similar to RSM. Fig.2 demonstrates that clearly.

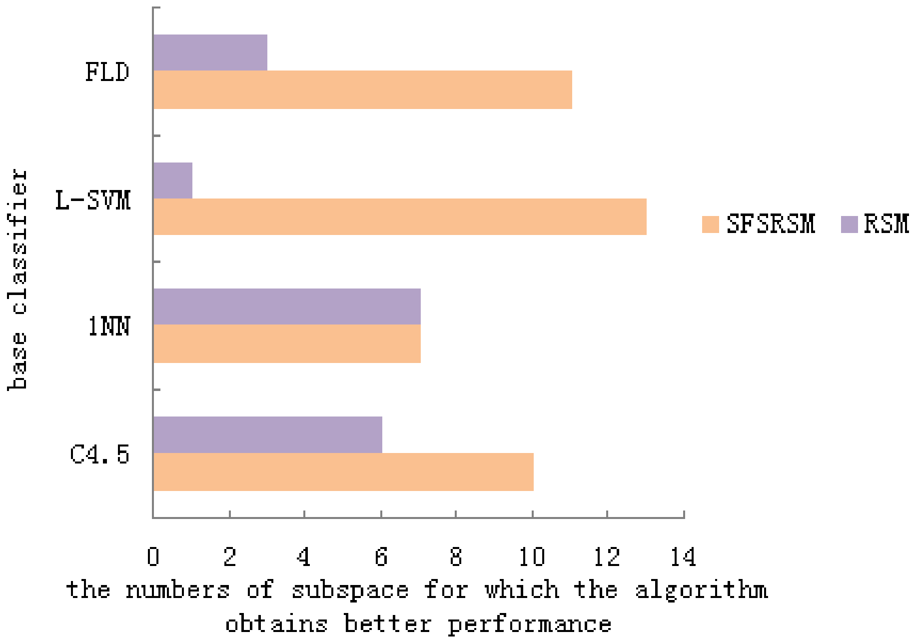

FIGURE 2. Comparison of the numbers of subspace for which the algorithm obtains better performance

From Table1∼Table4, it also can be seen that when base learner is C4.5,L-SVM,FLD, respectively, at p=\{0.7,0.3\} the proposed SFSRSM gets the average detection accuracy of all subspaces of 68.51%, 72.7%,74.95%,respectively,which increases the average detection accuracy of the RSM algorithm of all subspaces by 0.6%, 3.6%, 0.5%,respectively. Especially, the SFSRSM algorithm gets the highest detection accuracy of 80.28% which increases the optimal value of the RSM algorithm by 1%. The data represented in Table1∼Table4 demonstrates the effectiveness of the proposed algorithm.

(2)Experimental results in Table1∼Table4 show that, for high dimensional features a large proportion of the features may not be informative of the class of an object, there is a high chance to miss the informative features in RSM by using extremely random sampling method. However, our algorithm can ensure that each subspace contains enough informative features for classification in high dimensional features. The key idea is to divide the original feature space into two parts: fixed subset and remaining feature subset, fixed part will contain strong informative features and the other weak informative features. We randomly select features from each group proportionally to form subspace. Therefore, the classification ability of base learners is improved and the diversity between them is maintained due to the randomization in the feature subspace generation, thus, the performance of our algorithm is better than that of RSM.

(3)It can be seen from Table1∼Table4, partial detection accuracy of the proposed algorithm is lower than that of RSM when the dimensionality of the subspace r is low or high. The reason for that is in our algorithm, the size of the fixed subset is the same with$r$. When r is low, we select features from fixed subset with small size randomly, so that the possible number of the same features contained in subspaces may be more. Thus the diversity within the ensemble may be reduced. While when the value of r increases to a certain extent, the classification accuracy of base learners may be weakened due to the fact that the probability of selecting less informative features from fixed subset with big size is relatively large. Thus, the size of the fixed subset affects the performance of the proposed algorithm and should be set to make a good balance between the accuracy and diversity of base learners. The search for the optimal value of the fixed subset size is the further work.

(4)For the proposed algorithm, increasing the proportion of the features in the first part appropriately can extend the subspace's classification ability to improve the fusion decision. However, if we select too much features with higher classification ability, the accuracy of ensemble classifier will decline. This further indicates that we should make a trade-off between diversity and accuracy of the base classifier. From Table1∼Table4, it can be seen when p is $\{0.7,0.3\}$, the detection accuracy is the best on the whole, which provides the basis for feature selection in the practical application.

(5) As is already apparent from Table1∼Table4, the detection accuracy of the SFSRSM and the RSM algorithm first increases and then decreases with the increase of dimensionality of feature subset, which is mainly because of the following two reasons. On the one hand, the redundancy between features will increase with the growing dimension sizes of subsets, which may reduce the classification accuracy and the dependency between the base classifiers increases, thus the ability of the ensemble classifier to form non-linear boundaries decreases, the other hand, the base classifiers start to suffer from overtraining as the subspace dimensionality increases while the training size remains the same.

In order to visualize the performance of the algorithms, the Receiver Operating Characteristic (ROC) curve is used when comparing both methods. Fig. 3 plotted the curves at a fixed subspaces dimensionality(r=300), for different base learners (C4.5, 1NN, L-SVM, FLD). The great advantages of our proposed algorithm in detection accuracy can be obviously seen from Fig.3.

In order to further verify the effect of the proposed algorithm, we also compare the training time of the SFSRSM algorithm with FLD as the base learner and RSM for different r. The experiment is repeated 20 times, and we calculate the average value of the training time as the final result. The training time of the SFSRSM algorithm is the average value of the SFSRSM algorithm with different sampling rate. The comparison is reported in Fig.4. This experiment reveals that although our algorithm needs to select features using SFS algorithm and compute the weights of base classifiers, there is little

Table 1. Comparison of the detection accuracy of different algorithm using C4.5 as base learner

r	RSM	The proposed algorithm			
		p ={0.5,0.5}	p ={0.6,0.4}	p ={0.7,0.3}	p ={0.8,0.2}
100	**71.80±1.67**	67.46±1.45	67.36±1.73	68.25±1.96	67.7±1.73
150	65.17±1.63	66.78±1.05	66.67±0.59	**66.85±0.62**	66.6±0.87
200	**70.95±1.85**	68.40±1.64	68.57±1.32	68.63±1.62	68.2±1.67
250	67.18±2.22	67.41±1.45	67.38±2.68	**67.99±2.41**	67.2±2.12
300	70.50±3.02	71.74±2.79	71.61±2.48	**71.85±2.99**	71.7±2.68
350	**66.83±1.62**	65.40±0.97	65.33±0.80	**66.83±0.91**	65.40±0.86
400	68.00±0.80	68.55±0.63	**69.52±0.47**	68.63±0.66	68.4±0.55
450	**67.84±1.43**	66.65±1.39	**67.84±1.11**	67.83±1.88	67.37±1.32
500	69.17±1.54	69.17±1.16	69.20±1.03	**69.45±1.09**	69.37±1.21
600	**68.05±2.09**	66.95±1.20	66.87±1.28	67.25±1.40	66.75±1.93
700	67.95±2.25	69.81±1.93	69.95±2.24	**70.00±2.21**	69.64±2.59
800	66.73±1.72	**70.47±1.11**	70.33±1.76	70.33±1.88	70.11±1.75
900	**67.85±1.33**	66.94±1.10	66.00±1.32	66.95±1.45	67.82±1.45
1000	65.29±0.80	67.69±0.52	67.68±0.86	**67.84±0.92**	67.50±1.01
average	68.09	68.10	68.17	**68.51**	68.10

Table 2. Comparison of the detection accuracy of different algorithm using 1NN as base learner

r	RSM	The proposed algorithm			
		p ={0.5,0.5}	p ={0.6,0.4}	p ={0.7,0.3}	p ={0.8,0.2}
100	**77.26±1.12**	74.00±1.07	75.63±1.56	74.40±1.17	73.48±1.31
150	**74.58±2.38**	71.50±1.67	72.85±2.10	71.90±2.25	70.30±1.83
200	77.89±0.91	77.91±1.03	**79.10±0.86**	78.31±0.92	76.03±0.71
250	**75.15±1.49**	73.65±1.93	74.73±1.59	74.05±1.67	72.20±1.72
300	70.12±0.60	70.24±0.52	**71.55±0.75**	70.64±0.44	69.54±0.45
350	**72.21±0.93**	69.45±0.97	69.95±0.91	69.85±0.99	68.40±0.97
400	71.05±2.22	70.25±2.91	**71.85±1.33**	70.65±1.33	69.75±2.23
450	69.69±1.43	69.19±1.39	**70.35±1.12**	69.59±1.85	68.50±1.70
500	**76.33±1.49**	73.43±2.03	74.85±2.24	73.83±2.22	72.93±2.24
600	68.31±1.54	68.05±0.95	**69.40±1.07**	68.55±1.17	66.35±1.54
700	**69.56±2.03**	66.35±1.67	67.85±1.72	66.65±1.59	65.95±1.93
800	65.64±0.91	66.55±1.09	**67.81±0.86**	67.05±0.96	65.83±0.80
900	**67.40±0.87**	65.20±0.91	66.51±1.20	65.50±0.99	64.55±0.97
1000	66.65±2.37	67.14±2.41	**68.45±2.08**	67.54±2.59	66.58±2.41
average	**71.56**	70.21	**71.56**	70.61	69.31

difference between the training times of both algorithms. The maximum time difference is about 1minutes. Hence, the SFSRSM algorithm can offer training complexity comparable to RSM.

We also compare our algorithm with Bagging and AdaBoost. The results are reported in Table5. To facilitate the description, our method is described as SFSRSM_x.y, where x represents the sampling rate in the first part and y represents the sampling rate in the second part. For example, the SFSRSM_6.4 represents the sampling rate in the first

Table 3. Comparison of the detection accuracy of different algorithm using L-SVM as base learner

r	RSM	The proposed algorithm			
		p ={0.5,0.5}	p ={0.6,0.4}	p ={0.7,0.3}	p ={0.8,0.2}
100	72.20±2.03	73.11±2.31	73.71±2.37	**74.16±2.48**	73.50±2.19
150	71.63±3.71	73.63±3.29	73.85±3.55	**74.45±3.63**	73.58±3.30
200	72.85±2.49	**77.61±1.54**	76.35±1.20	76.57±1.14	77.55±1.52
250	72.05±3.43	76.65±2.93	76.48±2.45	**77.55±2.45**	76.25±2.61
300	69.12±2.39	74.45±2.68	74.60±2.38	**75.61±2.41**	74.61±2.35
350	71.26±1.62	72.80±1.97	72.86±2.36	**74.00±2.02**	73.00±2.51
400	70.25±1.49	72.55±1.59	**73.93±1.67**	73.93±1.72	72.25±1.91
450	71.72±1.13	70.26±0.92	70.46±1.07	**71.91±0.95**	70.43±1.02
500	71.64±2.44	73.11±2.67	72.97±2.71	**74.18±2.47**	72.85±2.71
600	66.49±2.01	67.12±1.20	**68.43±1.59**	**68.43±1.59**	67.18±1.28
700	68.15±2.65	69.46±2.83	69.49±2.35	69.21±2.61	**70.73±2.60**
800	69.65±1.78	69.15±1.61	69.07±1.26	**71.13±1.14**	69.20±1.31
900	65.16±2.23	**67.70±2.45**	66.55±2.32	66.65±2.60	67.65±2.41
1000	**70.05±2.63**	69.90±2.52	**70.05±2.55**	69.95±2.62	**70.05±2.52**
average	70.16	71.96	72.06	**72.70**	72.06

Table 4. Comparison of the detection accuracy of different algorithm using FLD as base learner

r	RSM	The proposed algorithm			
		p ={0.5,0.5}	p ={0.6,0.4}	p ={0.7,0.3}	p ={0.8,0.2}
100	79.18±0.60	80.05±0.87	80.05±0.88	**80.28±0.91**	80.15±0.93
150	77.49±1.73	78.15±1.91	77.35±1.59	**78.30±1.67**	77.90±1.72
200	78.20±1.89	**79.30±1.41**	78.00±1.64	78.25±1.23	78.33±1.65
250	77.79±0.86	78.00±1.03	77.29±1.33	**78.33±1.20**	77.79±1.09
300	73.80±1.32	74.61±1.54	**74.80±1.64**	74.61±1.71	74.55±1.72
350	75.55±0.78	75.80±0.37	**76.15±0.39**	**76.15±0.39**	75.61±0.32
400	74.63±1.10	75.05±1.29	**75.11±1.39**	74.90±1.32	74.85±1.35
450	71.96±0.87	72.33±0.96	72.54±0.63	**72.60±0.80**	71.96±0.59
500	**79.51±1.66**	**79.51±1.56**	**79.51±1.55**	**79.51±1.56**	79.33±1.50
600	**71.00±1.92**	70.42±1.33	**71.00±1.15**	**71.00±1.15**	70.40±1.30
700	73.14±2.89	**73.40±2.41**	**73.40±2.58**	73.35±2.58	73.14±2.30
800	**72.87±1.32**	72.67±1.73	72.67±1.72	72.67±1.96	72.67±1.67
900	70.95±2.03	70.55±1.96	70.75±1.80	**71.00±1.80**	70.75±1.83
1000	68.36±1.07	**68.93±0.52**	68.21±0.47	68.36±0.40	68.21±0.66
average	74.60	74.91	74.77	**74.95**	74.69

part is 60% and the sampling rate in the second part is 40%.The detection accuracy of our algorithm at different sampling rate is the average of the accuracy in 14 subspaces. From the results shown in Table5,it can be seen that the proposed algorithm obtains the highest detection accuracy compared with Bagging, AdaBoost and RSM with C4.5, 1NN and FLD as base classifier respectively. Especially with FLD as base classifier, the accuracy of the proposed algorithm is the highest. Compared with Bagging, AdaBoost and RSM, the average detection accuracy of the SFSRSM algorithm is increased by about

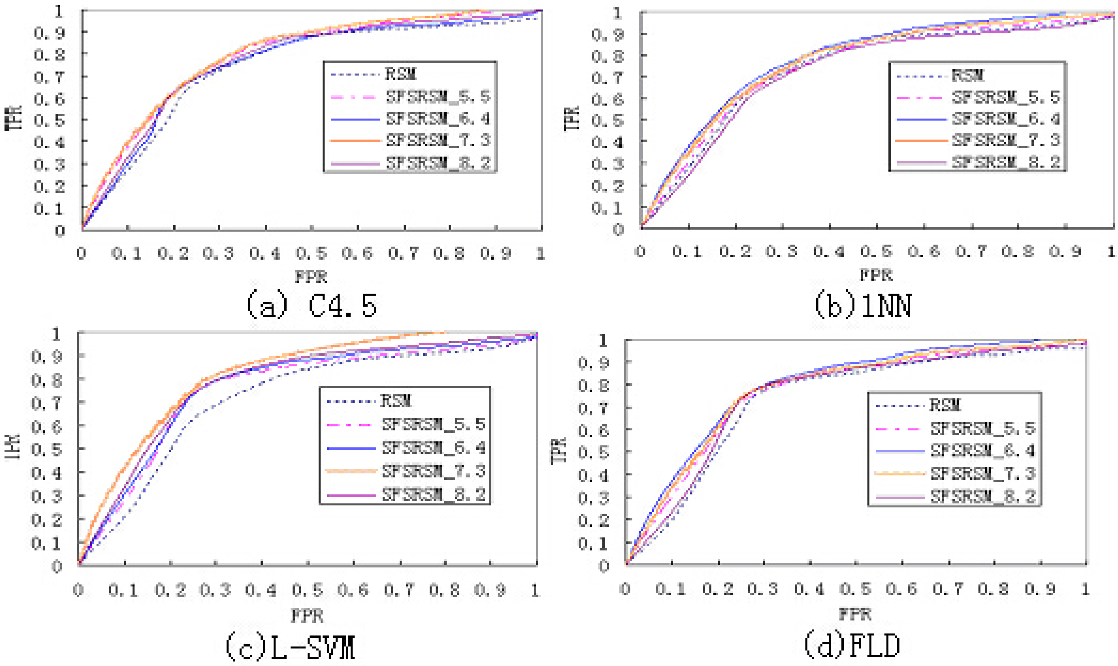

FIGURE 3. ROC curve of different algorithms using different base learners with r=300

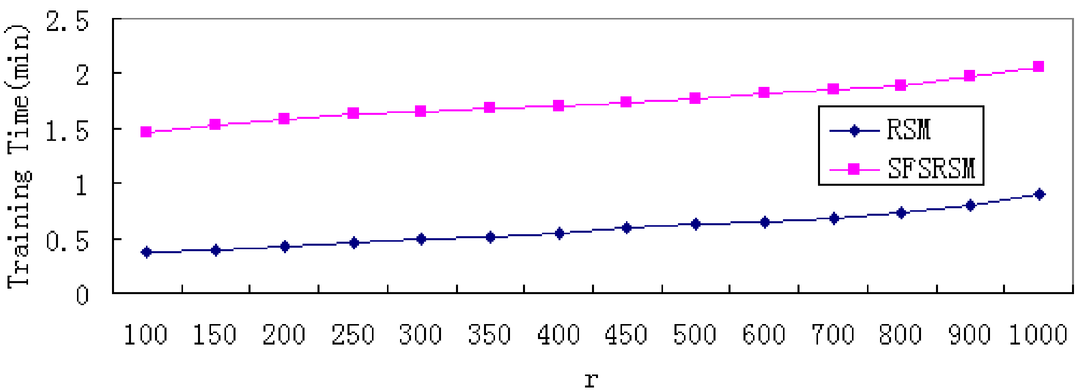

FIGURE 4. Training complexity of SFSRSM and RSM

3%, 3.6% and 0.3% respectively, which illustrates that the proposed algorithm performs better than these classical classifier ensemble methods.

3.3. **Effects of the fusion strategy on detection accuracy.** In order to compare the effect on the performance of the algorithm caused by different fusing strategies, paired t test is run at significance level 0.05 to compare the detection accuracy of the algorithms using the weighted voting strategy and the majority voting strategy in all the 14 subspaces. The results are shown in Table6,where w, l and d, represents the numbers of feature subspace for which the algorithm using weighted voting obtains better, worse, and equal performance than that using majority voting, respectively. It can be seen from Table6 that, when using C4.5 and 1NN as base classifier, the performance of ensemble classifier using weighted voting strategy is almost the same than that using majority voting strategy ,while using the other two base classifiers, the performance of the proposed algorithm is

TABLE 5. Comparison of the detection accuracy of different algorithms

Ensemble algorithm	Base learner			
	C4.5	1NN	L-SVM	FLD
Bagging	63.80%	68.00%	**73.94**%	72.39%
AdaBoost	68.20%	70.29%	69.94%	72.08%
RSM	68.09%	**71.56**%	70.16%	74.60%
SFSRSM_5.5	68.10%	70.21%	71.96%	74.91%
SFSRSM_6.4	68.17%	**71.56**%	72.06%	74.77%
SFSRSM_7.3	**68.51**%	70.61%	72.70%	**74.95**%
SFSRSM_8.2	68.10%	69.31%	72.06%	74.69%

much better, which illustrates that the method for weight calculation is reasonable and the weighted voting strategy can be helpful to improve the performance of ensemble classifier.

TABLE 6. Comparison of detection accuracy of the algorithms using different fusion strategies

fusion strategy	base classifier	p =$\{0.5,0.5\}$	p =$\{0.6,0.4\}$	p =$\{0.7,0.3\}$	p =$\{0.8,0.2\}$
		w/l/d	w/l/d	w/l/d	w/l/d
weight vs. majority	C4.5	1/2/11	5/4/5	3/3/8	3/3/8
weight vs. majority	1NN	3/3/8	2/1/11	1/0/13	3/2/9
weight vs. majority	L-SVM	5/3/6	6/4/4	6/4/4	7/4/3
weight vs. majority	FLD	6/3/5	8/3/3	9/3/2	5/2/7

3.4. **Effects of the number of base classifiers on the proposed algorithm.** To demonstrate the effect of the number of base classifiers on the performance of the proposed algorithm, Fig.5 shows the detection accuracy of the proposed algorithm for different number of base classifiers, three feature subsets of different dimensionality and sampling rate $p=\{0.7,0.3\}$. As can be seen from Fig.5, the classification accuracy of the proposed algorithm using different base classifiers quickly saturates with the number of base classifiers. For the fastest performance, one should choose the smallest number of base classifiers that gives satisfactory performance. We suggest the number of base classifiers m=25.

4. **Conclusions.** This algorithm is designed to encourage simultaneously individual accuracy and diversity within the ensemble. Specifically, the algorithm first selects features with higher classification ability as fixed subset using SFS, so that the original feature space is divided into two parts, then the final feature subset is formed by selecting features randomly in each part according to the given sampling rate and the finial decision is yielded by the weighted voting. Experimental results suggest that the proposed algorithm can get higher classification accuracy. The future study will focus on how to implement the new algorithm in parallel in a distributed environment, significantly reducing the time for creating an ensemble model from large data.

Acknowledgment. The authors would like to thank the reviewers for their careful reading and insightful comments. This work was supported by the Technology Innovation Platform Project of Fujian Province under Grant No.2009J1007, the Natural Science Foundation of Fujian Province under Grant No. 2010J01331.

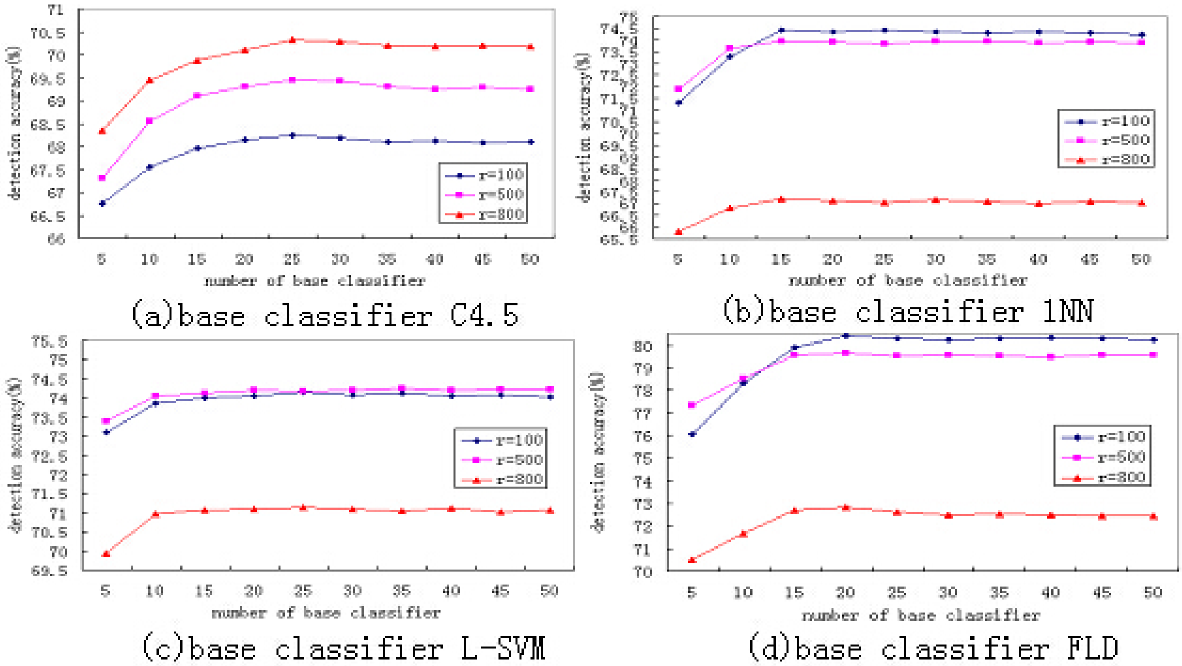

Figure 5. The effect of the number of base classifiers on the performance of the proposed algorithm

REFERENCES

[1] T. Filler, and J. Fridrich, Gibbs construction in steganography, *IEEE Trans. Information Forensics and Security*, vol. 5, no. 4, pp. 705-720, 2010.

[2] G. Gul, and F. Kurugollu, A new methodology in steganalysis: breaking highly undetectable steganograpy (HUGO), Information Hiding, pp.71-84, 2011.

[3] B. Li, J. He, and J. Huang, A survey on image steganography and steganalysis, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 2, no. 2, pp. 142-172, 2011.

[4] T. Mihara, A New Framework of Steganography Using the Content of Cover Data, *Journal of Information Hiding and Multimedia Signal Processing*, vol.5, no. 2, pp. 117-123, 2014.

[5] T. Pevny, P. Bas, and J. Fridrich, Steganalysis by subtractive pixel adjacency matrix, *IEEE Trans. information Forensics and Security*, vol. 5, no. 2, pp. 215-224, 2010.

[6] J.Kodovský, T. Pevný, and J. Fridrich, Modern steganalysis can detect YASS, *Proc. of the 7th SPIE Electronic Imaging, Media Forensics and Security*, pp. 1-11, 2010.

[7] J. Fridrich, J. Kodovský, and V. Holub, Steganalysis of content-adaptive steganography in spatial domain, *Proc. of the 13th Information Hiding*, pp. 102-117, 2011.

[8] R. L. Somorjai, B. Dolenko, and R.Baumgartner, Class prediction and discovery using gene microarray and proteomics mass spectroscopy data: curses, caveats, cautions, *Bioinformatics*, vol. 19, no. 12, pp. 1484-1491, 2003.

[9] M. Turk, and A. Pentland, Eigenfaces for recognition, *Journal of cognitive neuroscience*, vol. 3, no. 1, pp. 71-86, 1991.

[10] P. Comon, Independent component analysis, *Higher-Order Statistics*, pp. 29-38, 1992.

[11] N. Garcíapedrajas, and D. Ortizboyer, Boosting random subspace method, *Neural Networks*, vol. 21, no. 9, pp. 1344-1362, 2008.

[12] J. Kodovský, and J. Fridrich, Steganalysis in high dimensions: Fusing classifiers built on random subspaces, *Proc. of the 8th SPIE Electronic Imaging, Media Watermarking, Security and Forensics*, pp. 1-13, 2011.

[13] J. Kodovsky , J. Fridrich , and V. Holub, Ensemble classifiers for steganalysis of digital media, *IEEE Trans. Information Forensics and Security*, vol. 7, no. 2, pp. 432-444, 2012.

[14] C. Lai, M. J. T. Reinders, and L. Wessels, Random subspace method for multivariate feature selection, *Pattern Recognition Letters*, vol. 27, no.10, pp. 1067-1076, 2006.
[15] Y. Zhu, J. Liu, and S. Chen, Semi-random subspace method for face recognition, *Image and Vision Computing*, vol. 27, no.9, pp. 1358-1370, 2009.
[16] X. Wang, and X. Tang, Random sampling for subspace face recognition, *International Journal of Computer Vision*, vol. 70, no. 1, pp. 91-104, 2006.
[17] M .Yang, and F .Wang, A Classifier Ensemble Algorithm Based on Local Random Subspace, *Pattern recognition and artificial intelligence*, vol. 25, no. 4, pp. 596-602, 2012.
[18] R. Gilad-Bachrach , A. Navot , and N. Tishby , Margin based feature selection-theory and algorithms, *Proc. of the 21th international conference on Machine learning*, pp. 43, 2004.
[19] Y. Mao, X. B. Zhou, and Z .Xia, A Survey for Study of Feature Selection Algorithms, *Pattern recognition and artificial intelligence*, vol. 20, no.2, pp. 211-218, 2007.
[20] J. Fridrich, and J. Kodovsky, Rich models for steganalysis of digital images, *IEEE trans. Information forensics and security*, vol. 7, no. 3, pp. 868-882, 2012.
[21] R.Polikar, *Ensemble learning, Ensemble Machine Learning*, Springer, US, 2012.
[22] T. G. Dietterich, An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization, *Machine learning*, vol. 40, no. 2, pp. 139-157, 2000.
[23] L. Zhang, A Method for Improving the Stability of Feature Selection Algorithm, *Proc. of the 3th International Conference on Natural Computation*, pp. 715-717, 2007.
[24] P. Bas, T. Filler, and T. Pevný, “Break Our Steganographic System”: The Ins and Outs of Organizing BOSS, *Information Hiding*, pp. 59-70, 2011.