INFORMATION
TECHNOLOGY JOURNAL

Feature Selection for Image Steganalysis using Hybrid Genetic Algorithm

[1]Zhihua Xia, [1]Xingming Sun, [1,2]Jiaohua Qin and [1]Changming Niu
[1]School of Computer and Communication, Hunan University, Changsha, 410082, China
[2]Department of Mathematics and Computer, Hunan City University, Yiyang, 413000, China

Abstract: Learning-based methodology has been demonstrated to be an effective approach to dispose the steganalysis difficulties due to the variety of image texture. A crucial process of the learning-based steganalysis is to construct a low-dimensional feature set. In this study, a feature selection method based on Hybrid Genetic Algorithm (HGA) is presented to select feature subsets which not only contain fewer features, but also provide better detection performance for steganalysis. First, the general framework about utilizing Genetic Algorithm (GA) to do feature selection for steganalysis is presented. Then, we analyze similarity among individuals (SI) in each generation and the Transformation of Generations (TG) to determine whether the GA has converged into a local area. Next, according to the SI and TG, the restarting operation is incorporated into the HGA to allow the algorithm to escape from the unsatisfactory local area. In the experiments, three feature subsets are formed from a universal feature set for three typical steganography methods, respectively. The experimental results show that the classifiers using the feature subsets gain better detection accuracy and higher speed than those using the universal set.

Key words: Information security, feature selection, local convergence, similarity among individuals, transformation of generations

INTRODUCTION

Steganography is the art of covert communication, which was first applied in ancient Greece and China (Luo *et al.*, 2008). At present, steganography utilizing digital media as cover has drawn wide attention. Steganalysis, as the opponent part to steganography, is to detect whether hidden message exists in a given media. This study focuses on image steganalysis, which faces a serious challenge due to various image textures.

Plenty of steganalysis methods have been proposed in the past years. Among these methods, the learning-based steganalysis is a popular domain where many excellent researches have been carried out. Learning-based method processes images in a multidimensional feature space and learns the differences between cover images and stego-images in this space to train a classifier. Farid (2002) first proposed a framework for learning-based steganalysis and demonstrated it as an effective approach to cope with the steganalysis difficulties caused by various image textures and unknown steganography algorithms. This study inspired the development of many works based on all kinds of features extracted from different domains (i.e., spatial, DCT, DFT and DWT domains etc.) using different models

such as histogram, co-occurrence matrix, IQM (Avcibas *et al.*, 2003), PDF moments (Farid and Lyu, 2003), CF moments (Xuan *et al.*, 2005; Wang and Moulin, 2007) and Markov model (Chen and Shi, 2008; Shi *et al.*, 2006) etc. For the sake of improving the detection accuracy, some other studies combined the features from different origins. For example, Pevný and Fridrich (2007) merged the features extracted by Shi *et al.* (2006) and Fridrich (2004) to obtain better performance.

Although, the aforementioned methods can successfully detect hidden messages, they achieve the purpose at the expense of increasing the feature number which should never be made light of. Furthermore, puzzled by the numerous features, it is usually difficult to tell which features are valuable, which features just increase the computational complexity and which features even degrade the accuracy of the classifier. For learning-based steganalysis, feature extraction is crucial. To get an informative feature set which howbeit contains less noise and redundancy, two fundamental problems need to be addressed:

- How to extract features which are informative in terms of discriminating cover images from stego ones?
- How to select and assemble these informative features?

Corresponding Author: Xingming Sun, YueluMountain, Changsha, Hunan, 410082, China
Tel: 86-731-8821341 Fax: 86-731-8821341

While, the earlier study mainly focused on the first problem listed above, some other works mentioned the reduction of feature number utilizing SFFS (Wang and Moulin, 2007), SFS (Miche *et al.*, 2006) and PCA (Holotyak *et al.*, 2007). However, sequential search algorithm becomes ineffective as the feature number increases. In addition, PCA needs to carry through a space transformation to reduce the dimension when detecting an image. It is a time-consuming process. In this paper, we will tackle the feature selection problem using the proposed hybrid genetic algorithm.

Feature selection aims at finding an optimal feature subset. However, with an exponential solution space, feature selection is a problem of nontrivial size, in which the optimal solution is computationally intractable to obtain. Consequently, many existing algorithms, such as sequential search algorithm, simulation annealing algorithm, Tabu search algorithm and genetic algorithm, are designed to find suboptimal satisfactory solutions.

The Genetic Algorithm (GA) is biologically inspired and has many mechanisms mimicking natural evolution (Holland, 1992). It has a great deal of potentialities in scientific and engineering optimization. Moreover, GA is naturally applicable to process feature selection, as it has inherent parallelism and global optimization capability in an exponential search space (Brill *et al.*, 1992; Raymer *et al.*, 2000). Kudo and Sklansky (2000) have shown the advantages of the GA in feature selection compared with the representative classical algorithms and GA is expected to hold a good performance when the cardinality of the universal feature set D is large, especially when the expected cardinality of the selected subset d is significantly smaller than D (Siedlecki and Sklansky, 1989).

GENERAL GENETIC ALGORITHM BASED FEATURE SELECTION

Feature selection involves the selection of a subset of d features from a universal set of D features, based on a given optimization criterion. Let us denote the D-cardinality universal feature set as $U = (f_1, f_2,...,f_d)$ the d-cardinality subset of the selected features as X ($|X| = d$) and the subset of remaining features as Y. Then, $U = X \cup Y$.

When using genetic algorithm to do feature selection, features are treated as genes which have two possible statuses: selected or unselected. An individual, which is composed of genes, rightly corresponds to a feature subset, i.e., a solution in the problem domain. The set of some individuals is called population, denoted as P; Pi

denotes i-th individual in P; the number of individuals in P is denoted as |P|. An objective, called fitness function, is used to evaluate the quality of individuals. GA starts with an initial population as the first generation. To evolve, the individuals (parents) in a generation (parent generation) are utilized to reproduce q child individuals using genetic operators, such as crossover and mutation. Then, |P| individuals are selected from the set of |P| parents and q children by selection operator. The |P| selected individuals make up of a new generation (child generation) on which the next iteration will apply. The algorithm repeats until a predefined criterion is met or the maximal number of iterations is reached. Present proposed GA is described in Algorithm 1.

Algorithm 1: GA
Input: Universal feature set U, GA parameters |P|, q, g, a, b, m, n
Output: Feature subset X
Body:
1. Define the individual encoding for U;
2. Initialize population P using Algorithm 2; k=1;
3. **while** k <= q
    3.1. Randomly select two different parents pi and pj from P;
    3.2. offspringk = pi, offspringk+1 = pj;
    3.3. Crossover(offspring$_k$, offspring$_k$+1, m);
    3.4. Mutation(offspring$_k$, n);
    3.5. Calculate the quality of offspringk using fitness function; k=k+1;
    3.6. **if**(k> q) discard offspring$_k$, **got**o 4;
    3.7. Mutation (offspringk, n);
    3.8. Calculate the quality of offspringk using fitness function; k=k+1;
4. Selection (P, offspring); k=1;
5. Record the quality and selected features of individuals in P;
6. **if** the quality of the best individual in P is lower than the predefined criterion, **goto** 3;
7. **Return** the feature subset X which contains the selected features of the best individual in P.

In the algorithm, the input parameter g is the number of selected features of individuals in initial population. a and b are the parameters of fitness function. m and n and the parameters of crossover and mutation, respectively. The values of all parameters will be suitably configured in the experiment. The detailed specifications of the GA are described in the following subsections.

Individual encoding: Individual is composed of genes (i.e., features). In our scheme, a gene is represented by a

binary digit with a value of 1 (0), indicating that the appointed gene is (or not) selected. Then, with D features, an individual could be denoted by a string of D binary digits with one bit corresponding to one feature, respectively. For example, supposing D = 9 and giving each feature a subscript, the universal feature set could be expressed as $U = (f_1,f_2,f_3,f_4,f_5,f_6,f_7,f_8,f_9)$. Then, establishing orderly the one-to-one map between the features and binary digits, a string of 9 binary digits 010100100 can be used to denote an individual, of which the 2nd, 4th and 7th features are selected. In this way, the corresponding selected feature subset $X = (f_2,f_4,f_7)$ and the subset of remaining feature $Y = (f_1,f_3,f_5,f_6,f_8,f_9)$.

Fitness function: The fitness function is used to evaluate the quality of individual, such as IX whose appointed feature subset is X. The selected feature subset is expected to contain more information and fewer features. Then, the classifier trained with these features would have high accuracy and speed. Consequently, the fitness function can be formulated as:

$$Fit(I_x) = a \cdot accuracy_x - b \cdot |X| \qquad (1)$$

where $accuracy_x$, is the correct detection rate of the classifier that utilizes the features in X; $|\cdot|$ denotes the cardinality of set $\cdot$, a and b are constants.

Initialization of population: Without researching on that prior knowledge here, the first generation is initialized randomly. |P| individuals, each of which consists of g randomly selected features, make up of the first generation. The process of initializing the population is described in Algorithm 2.

Algorithm 2: Initialization
Input: GA parameter |P|, g
Output: Initial population P
Body:
1. for i=1 to |P|
   Randomly select g features for individual p_i;
2. Calculate the qualities of individuals in P using fitness function;
3. Return initial population P;

In the algorithm, quality is equivalent to the detection accuracy of the classifier which adopts the selected feature of the individual. |P| and g are the same as algorithm 1.

Genetic operators: Genetic operators are utilized to generate a child generation from a parent generation.

Generally, genetic operators include crossover, mutation and selection.

Crossover: Crossover is the operator which reproduces new individuals by exchanging the genes between two individuals. Crossover inherits and reassembles selected genes from the parents. As, the better individuals usually contain better selected genes, the crossover operator assures that the better selected genes will be retained in the new generation. The types of crossover include one-point, two-point and multi-point crossover. In this study, the m-point crossover differs from general ones and is defined as follows to reassemble genes more neatly.

$$Crossover(p_i, p_j, m) = \{p_i \overset{m}{\Theta} p_j \mid p_i, p_j \in P, i \neq j, 0 < m < D\} \qquad (2)$$

where, $p_i \overset{m}{\Theta} p_j$ denotes the operation that reproduces two child individuals by swapping statuses of genes of two individuals pi and pj at m randomly chosen points. A 6-point crossover operation is illustrated in Fig. 1a. It shows that two parent individuals, whose appointed selected feature subsets are {f2,f4,f6,f7,f9,f12} and {f1,f4,f5,f7,f10,f13}, respectively, are inputted to reproduce child individuals by crossover operator. The corresponding subset of the two resulting children are {f4,f5,f6,f7,f10} and {f1,f2,f4,f7,f9,f12,f13}, respectively.

Mutation: Mutation is applied to the offspring propagated by the crossover operator. It randomly chooses some genes of individual and changes their statuses. The mutation operator is usually performed with a probability $\alpha, 0 < \alpha < 1$, which means that only portion of the genes in an individual will be allowed to mutate. Therefore, the
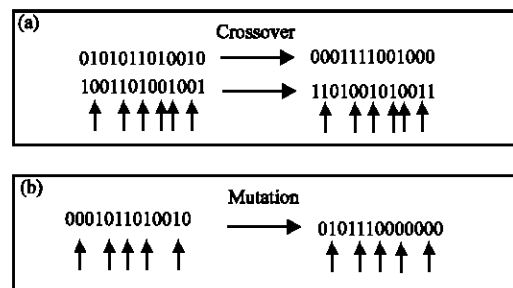

Fig. 1: (a) A 6-point crossover exchanges genes of two individuals at the 6 randomly chosen points pointed by arrows; (b) A 5-point mutation switches status (1/0) of genes at the 5 randomly chosen points pointed by arrows

number of mutated genes is $n = \alpha \times D$. The n-point mutation is defined as follow:

$$\text{Mutation}(p_k, n) = \left\{ \overset{n}{\odot} p_k \mid p_k \in P, 0 < n < D \right\} \qquad (3)$$

where, $\overset{n}{\odot} p_k$ denotes the operation that switches the statuses (1/0) of n randomly chosen genes of an individual pk. A 5-point mutation operator is shown in Fig. 1b. It expresses that an individual, whose appointed feature subset is $\{f_4, f_6, f_7, f_9, f_{12}\}$, is operated by the mutation operator. As the resulting individual, its corresponding feature subset is $\{f_2, f_4, f_5, f_6\}$.

Selection: After reproduced by crossover and mutation operators, the child individuals will be evaluated by fitness function and the children of better quality will replace bad-quality individuals in the parent generation. That is to say, there should be an operator that selects better individuals from the aggregation of child and parent individuals to make up of a new generation. This operator is called selection and is defined as follow:

$$\left\{ \xi_i \mid \psi = P \cup \text{offspring}, \xi_i \in \psi, i < |P|, \right\} \qquad (4)$$

where, ξ_i is the I-th fittest individual in Ψ. P is the parent population, offspring is the set of child individuals.

HYBRID GENETIC ALGORITHM

A common problem of genetic algorithm is that it may converge into local area, failing to return the global optimum solution. If the local optimal solution is not satisfactory, the genetic algorithm should escape from the local to find a better one. Usually, a General Genetic Algorithm (GGA) can be made capable of escaping from local by properly adjusting the parameters of GGA. However, it is often achieved at the high expense of time complexity. Moreover, it is likely to return us a subset with large cardinality. Considering these problems, in our algorithm, when the GA is trapped in a local area and the local optimum does not achieve the predefined target, the GA will restart from a new initial population. Hence, we should be able to judge whether the GA is trapped in a local. In following subsections, the similarity among individuals (SI) in each generation and the transformation of two consecutive generations (TG) are analyzed.

Similarity among individuals: To determine whether the GA has converged to a local area, we can observe the similarity among individuals (SI) in each generation. When there are great differences among the individuals,

individuals are distributed dispersedly in solution space. In this situation, it can be considered that GA has not converged into a local and vice versa.

The more identical features two individuals select, the more similar they will become. To calculate the SI of a generation, each pair of individuals should be compared. For simplification, the Feature Variety (FV) of the generation is utilized to evaluate the SI. If the selected features are more various, the individuals will contain less similarity. In this study, FV is equivalent to the proportion of number of distinguishing features and the number of all features in a generation. FV is formulated as follow:

$$FV = \frac{\left| \bigcup_{i=1}^{NI} SS_i \right|}{\sum_{i=1}^{NI} |SS_i|} \qquad (5)$$

where, NI is the individual number in the generation; SS_i is the corresponding feature subset of i-th individual in the generation:

$$\bigcup_{i=1}^{NI} SS_i = SS_1 \cup SS_2 \cup \cdots \cup SS_{NI}$$

Then SI can be defined as follow:

$$SI = 1 - FV \qquad (6)$$

Here, if all individuals in the generation do not select any identical feature with each other, the individuals in the generation do not have any similarity and the SI will be calculated to zero.

Transformation of generations: Whether the GA gets into a local domain can be judged by the differences of two consecutive generations. While, the GA is running, if the differences between the two consecutive generations become fewer, the GA as likely as not runs into a trap. In the paper, this difference is named as Transformation of Generations (TG).

To measure TG, all individuals in a generation should be compared with the individuals in its parent generation; TG is formulated as follow:

$$TG_j = \frac{\sum_{m=0}^{NI_j} \sum_{n=0}^{NI_{j-1}} \frac{\left| (SS_{m,j-1} - SS_{n,j}) \cup (SS_{n,j} - SS_{m,j-1}) \right|}{(|SS_{m,j}| + |SS_{n,j-1}|)/2}}{NI_j \times NI_{j-1}} \qquad (7)$$

where, $j = 2, 3, \ldots, NG$, NG is the number of generations; Ni_j is the number of individuals in j-th generation; $SS_{i,j}$ is denotes the corresponding subset of i-th individual

in j-th generation; the symbol (−) between two sets indicates the set theoretic difference operation of two sets.

Hybrid genetic algorithm: As general trend, while the GA is gradually converging to a local, the SI and TG will increase and descend respectively and their changing rates are descending (shown in subsection 4.4, Fig. 2a-c). In this study, when SI and TG reach some predefined thresholds and the quality of the best individual in a generation is lower than the target value, GA is considered trapped in unsatisfactory local. In this situation, algorithm should be restarted. The Hybrid Genetic Algorithm (HGA) is described in Algorithm 3.

Algorithm 3: HGA

Input: Universal feature set U£¬HGA parameters |P|, q, g, a, b, m, n, T_{SI}, T_{TG}, T_{DA}

Output: Feature subset X

Body:

1. Define the individual encoding for U;
2. Initialize population P using Algorithm 2; k=1; Calculate SI, TG, DA of P;
3. **while** k <= q
   - 3.1. Randomly select two different parents p_i and p_j from P;
   - 3.2. offspring$_k$ = p_i, offspring$_{k+1}$= p_j;
   - 3.3. Crossover (offspring$_k$, offspring$_{k+1}$, m);
   - 3.4. Mutation(offspring$_k$, n);
   - 3.5. Calculate the quality of offspring$_k$ using fitness function; k=k+1;
   - 3.6. **if**(k> q) discard offspring$_k$, **goto** 4;
   - 3.7. Mutation(offspring$_k$, n);
   - 3.8. Calculate the quality of offspring$_k$ using fitness function; k=k+1;
4. Selection (P, offspring); k=1;
5. Record the quality and selected features of individuals in P;
6. Calculate SI, TG, DA of P;
7. **if** SI>T_{SI}&&TG<T_{TG}&&DA<T_{DA} **goto** 2;
8. **if** the quality of the best individual in P is lower than the predefined criterion, **goto** 3;
9. **Return** the feature subset X which contains the selected features of the best individual in P.

In the algorithm, |P|, q, g, a, b, m and n carry the same meaning as they are in ALGORITHM 2. Let us denote the detection accuracy of the best individual in a generation as DA. TSI, TTG and TDA denote the thresholds of SI, TG and DA respectively. The values of these thresholds will be given out in the experiment.
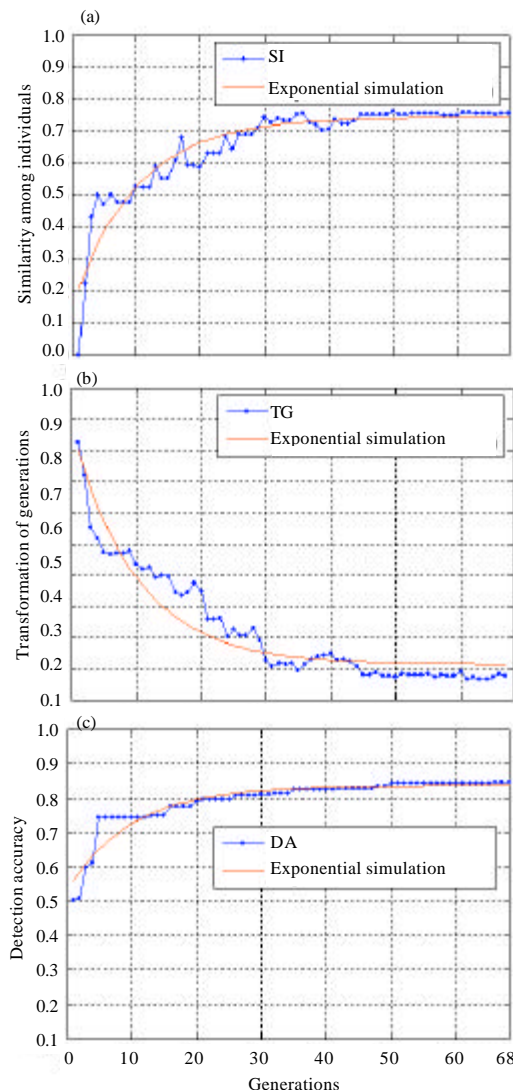

Fig. 2: (a) SI, (b) TG and (c) DA curves in a 68-generation GA running; this running is considered trapped in a local area; the red curves are the exponential simulation curves

RESULTS AND DISCUSSION

Image sets: The image property will influence the accuracy of steganalysis. For example, secrete messages embedded in images with complex texture will be harder to be discovered than that embedded in smooth images. For universality, the image set used in steganalysis always include various images. In our experiments, the images are downloaded from www.freefoto.com. These images span a range of indoor and outdoor scenes and are of different content and texture.

Three typical steganography methods, Outguess (Provos, 2001), F5 (Westfeld, 2001) and Model Based Steganography without deblocking (MB1) (Sallee, 2003), are tested in the experiments. In order to avoid the impact of quantization factor, we prepare cover images and stego-images for the three methods, respectively.

For each method, the corresponding 3700 images are embedded message with the length of 0, 10, 25, 50 and 100%, embedding capacity of each image, respectively. The images with the embedding rate of 0% are regarded as cover images and the remainders are utilized as stego images. The embedding capacity is defined as the maximum size of the message that can be embedded in an image with an embedding algorithm. Since, the number of the zero coefficients varies in different JPEG images, embedding capacities differ from image to image. Table 1 shows 4 statistics of capacity of cover images for three embedding algorithms: mean, standard deviation, minimum and maximum. The embedded secret messages are pseudo-random generated digitals; the secret messages for each stego-image are different.

Support vector machine: In the experiments, SVM-Chen 2.0 is adopted as the classifier. Support Vector Machine (SVM) is a supervised learning method separating clouds of data in feature space by an optimal hyper-plane (Burges, 1998). SVM can handle not only the linear case but also the nonlinear case. In SVM-Chen 2.0, there are four basic predefined kernels: linear, polynomial, radial basis function and sigmoid. The linear kernel is for linear SVM and the rest three kernels are for nonlinear SVM. In our experiments, we select the kernel of radial basis function with the other parameters of default setting.

Universal features set: Various features can be extracted from the image to represent the image itself. In this study, the universal feature set consists of 274 features which are originally described by Pevný and Fridrich (2007); it holds an excellent detection performance. The functional and features discussed in this study are listed in Table 2.

HGA parameters setting: Many factors need to be taken into consideration to configure GA parameters, among which the cardinality of the universal feature set D and the expected cardinality of selected subset d are very important. In our experiments, the cardinality of the universal set is 274. However, in a Fridrich's study (Fridrich, 2004), the number of features is only 23 and it is accurate in classification. Hence, using our HGA, we expect to find a satisfactory subset with low cardinality.

Table 1: The statistics of embedding capacity of covers for three embedding algorithm

Statistics	Statistical values (kb)		
	Outguess	F5	MB1
Mean	2.3190	4.543	4.806
SD	0.9623	1.883	2.050
Minimum	0.2722	0.197	0.232
Maximum	8.4810	12.530	12.500

Table 2: Universal feature set with 274 features

Functional/feature name	Feature No.
Global histogram H_r	11
5 AC histogram h_r^{ij}	55
9 AC Dual histogram g_{ij}^d	99
Co-occurrence matrix $C_{s,t}$	25
Variation (V)	1
Block discontinuities B_v	2
Markov feature $\overline{M}(i, j)$	81

In a word, the case in our experiments is that the D is large and the d is expected to be significantly smaller than D, i.e., d<<D.

General parameters for GA: Each parameter will impact greatly on the performance of GA. As the detection accuracy is very important to steganalysis, the fitness function coefficient a is set to 1 and the coefficient b is set to 0 in our algorithm. Generally, this will eliminate the impact of the number of selected features and increase the cardinality of the final selected subset. But in our case (d<<D), we can confine the cardinality of subset by configuring other parameters properly. For example, by initializing the individuals of the first generation with few selected features, namely, initializing the number of selected features of individuals in first generation g with a small value, the number of selected features of individuals will increase from a small value. When carrying through crossover operation, GA will choose genes from two individuals randomly. In the case of d<<D the genes are not likely selected by both individuals. Thus, it makes no sense to exchange their statuses. To assure the efficiency of crossover, a large crossover parameter m is configured to choose more genes to exchange their statuses. Similarly, when carrying through mutation operator, it as likely as not chooses unselected genes and changes to select them. If the mutation parameter n is large, it will rapidly increase the cardinality of the subset. Hence, a small n is configured. In the case of d<<D, according to the above analysis, the amount of selected features of individuals is increasing in the general trend while the GA running. The scale of population affects the searching density of the algorithm at each size of cardinality of subset. In order to obtain a low-cardinality subset, a comparatively big population scale |P| is configured. So the q, which is the number of child individuals reproduced in each generation.

Special parameters for HGA: To determinate whether the GA has converged into an unsatisfactory local area, the threshold values of the SI, TG and DA should be configured. The DA denotes the detection accuracy of the classifiers which utilize the selected features of the best individual in each generation.

In Fig. 2 the SI, TG and DA curves of a GA running are shown. It is a running of 68 generations with the stego-images embedded by MB1; the embedding rate is 25%.

Figure 2 shows that the DA and SI are increasing, while TG is descending in the general trend. All curves change acutely at the first 5 generations; it is the early period. (a) and (b) show that the slope of SI curve and that of TG curve both approach to 0 after 40th generation and 50th generation respectively. It implies that the GA gets into a local area gradually. Synchronously, from (c), we observe that the GA runs 38 generations to improve the DA by only about 5%, getting the DA of 85%, after 30th generation. It is computationally expensive. Furthermore, satisfactory result may never be provided in this situation. This is considered as a trapped GA running. Experimentially, we set threshold values $T_{SI} = 0.75$, $T_{TG} = 0.1$, $T_{DA} = 0.95$ to determine whether the GA is trapped in an unsatisfactory local area. In sum, the parameters of HGA are listed in Table 3.

Feature selection using HGA: For a special embedding method, the sensitive features may be some special ones. HGA is adopted to select three subsets for three embedding methods, namely Outguess, F5 and MB1, respectively. According to the definition of the fitness function, the quality of the individual is the detection accuracy of the classifier which utilizes the corresponding feature subset of the individual. Then, the training and testing image sets should be prepared to train and test the SVM classifier. For each embedding method, the training set consists of 2000 original images and 2000 corresponding stego-images with the embedding rate of 25%; testing set consists of the remaining 1700 original images and 1700 stego-images.

All individuals and their detection results in each generation are recorded. Fig. 3a shows detection accuracy of classifiers which use the best individual in each generation in the feature selection experiments for three embedding methods. SI curve and TG curve are also shown in Fig. 3b and c.

The HGA selects three feature subsets for three embedding methods, respectively. The selected feature subsets are listed in Table 4.
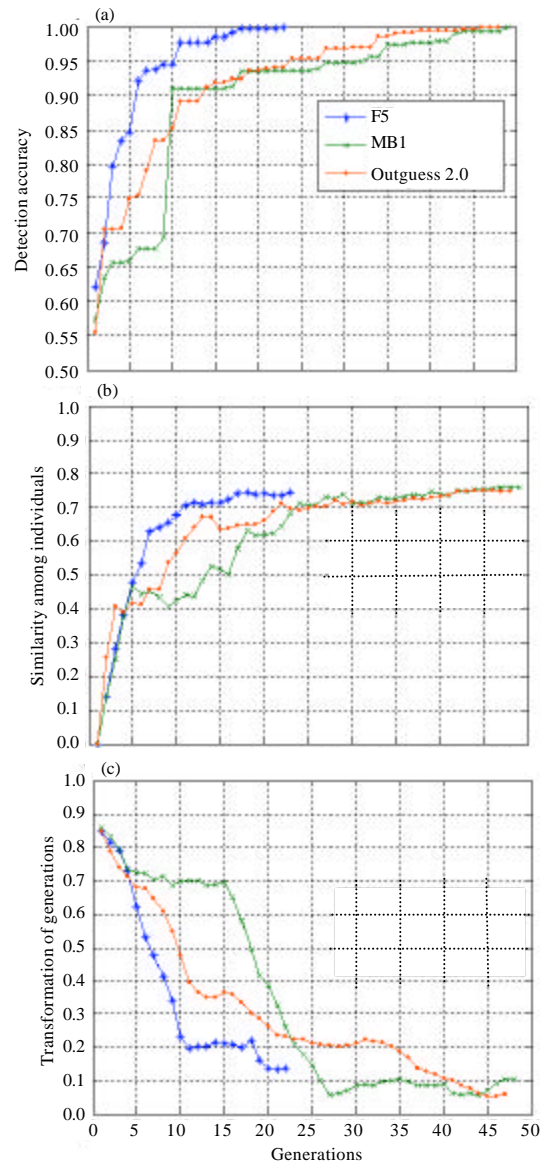

Fig. 3: The (a) DA, (b) SI and (c) TG curves in feature selection experiments; the number of generation for F5, MB1 and Outguess are 23, 49 and 48, respectively

Table 3: HGA parameters

Parameters	Symbol	Value
Population scale	|P|	20
Number of child individuals	q	20
Number of selected genes of individuals in initial generation	g	1
Contribution coefficient of fitness function	a	1
Penalty coefficient of fitness function	b	0
Parameter of crossover	m	100
Parameter of mutation	n	1
Stop condition	——	DA=100%
Threshold value of SI	T_{SI}	0.75
Threshold value of TG	T_{TG}	0.10
Threshold value of DA	T_{DA}	0.95

Table 4: The selected features subsets for the three steganography methods

Embedding method	Freature subset	Cardinality
F5	$C_{1,1}, h_{-5}{}^{20}, h_{-1}{}^{02}, h_2{}^{02}, h_3{}^{02}, g_{03}^2, g_{21}^0, g_{21}^2, \overline{M}(-4,3), \overline{M}(-2,-2)$ $\overline{M}(-1,0), \overline{M}(-1,1), \overline{M}(0,1), \overline{M}(1,-1), \overline{M}(3,-2), \overline{M}(3,1)$	16
Outguess 20	$C_{-2,-2}, C_{-2,1}, C_{0,-2}, C_{0,1}, C_{2,-2}, C_{2,0}, C_{2,2}, h_3{}^{20}, h_5{}^{11}, h_{-4}{}^{02}, h_3{}^{02}, \overline{M}(-3,1),$ $\overline{M}(-3,2), \overline{M}(-2,0), \overline{M}(-1,-4), \overline{M}(-1,-3), \overline{M}(-1,-1), \overline{M}(-1,0),$ $\overline{M}(1,-1), \overline{M}(2,0), \overline{M}(3,-4), \overline{M}(3,-2), \overline{M}(3,1)$	23
MB1	$C_{-2,-2}, C_{-2,-1}, C_{2,-2}, C_{2,2}, H_{-5}, H_4, \overline{M}(-4,-2), \overline{M}(-2,2), \overline{M}(-1,3),$ $\overline{M}(1,-1), \overline{M}(2,-3), \overline{M}(2,0), \overline{M}(2,3), \overline{M}(3,4), \overline{M}(4,-4)$	15

Detection performance: In this subsection, three feature subsets are utilized to train three classifiers and the universal set (P and F's feature set) is also utilized to train three classifiers. Subsequently, the three subsets are merged to form a union set which has the cardinality of 44. The union set is also utilized to train three classifiers to detect secret messages of three embedding methods. For further comparison, we implement another work (Chen and Shi, 2008) with 486 features (C and S's feature set) which exploited intra-block and inter-block characters among DCT coefficients. Thus, 12 classifiers will be trained and tested.

For each classifier, the training set contains 2000 cover images and the corresponding 2000 stego-images with the embedding rate of 10%, 2000 stego-images of 25%, 2000 stego-images of 50% and 2000 stego-images of 100%. When training classifier, the 2000 cover images are repeatedly used four times so as to ensure that the amount of covers equals that of stego-images. The testing set contains the 1700 remaining covers and corresponding 1700 stego-images with the embedding rate of 10%, 1700 of 25%, 1700 of 50% and 1700 of 100%. The detection accuracies of the 12 classifiers are shown in Table 5.

Table 5 shows the comparison about the detection accuracy of classifiers adopting the features in three selected subsets, merged set of these three subsets, universal set (P and F's) and C and S's feature set. The results show that the classifiers which adopt the selected feature subsets and which adopt the merged feature set, hold better detection accuracy than the classifiers adopting P and F's and C and S's feature set in our experiment. The average increases of detection accuracy of selected subsets to P and F's set are 0.04708, 0.0560 and 0.0648 for Outguess, F5 and MB1 respectively. The increases are especially obvious when detecting the images with the low embedding rate.

The classifiers using fewer features are expected to be faster in terms of speed. Table 6 lists the time consumed by 12 classifiers when detecting 15 testing image sets respectively. The experiments are carried out under the same condition. The CPU of the computer is Pentium (R) 4 3.00 GHz.

Table 5: Comparison of detection accuracy

Embedding method	Embedding rate	Detection accuracy			
		Selected	Merged	P and F's	C and S's
Outguess	100%	0.9982	0.9988	0.9900	0.9994
	50%	0.9853	0.9782	0.9741	0.9812
	25%	0.8553	0.8571	0.7929	0.8418
	10%	0.4271	0.2818	0.3094	0.4665
	Cover	0.9088	0.8982	0.8729	0.9047
F5	100%	0.9959	0.9988	0.9988	0.9994
	50%	0.9912	0.9888	0.9818	0.9741
	25%	0.8982	0.7582	0.7953	0.5024
	10%	0.3971	0.2841	0.2382	0.1435
	Cover	0.8759	0.9029	0.8641	0.8953
MB1	100%	1.0000	0.9912	0.9900	0.9988
	50%	0.9982	0.9988	0.9712	0.9994
	25%	0.9212	0.8753	0.7682	0.9553
	10%	0.4188	0.4088	0.2941	0.3400
	Cover	0.9112	0.9141	0.9018	0.9176

Table 6: Comparison of time expenditure

Embedding Method	Embedding rate	Consumed time (sec)			
		Selected	Merged	P and F's	C and S's
Outguess	100%	6.484	10.656	46.281	58.891
	50%	6.360	10.531	45.579	59.969
	25%	6.406	10.188	45.500	59.968
	10%	6.281	10.312	48.597	59.688
	Cover	6.281	10.266	45.473	60.312
F5	100%	7.156	11.437	50.188	81.844
	50%	6.938	11.344	43.891	82.281
	25%	6.875	11.359	43.968	82.883
	10%	6.859	11.250	43.766	81.906
	Cover	7.063	11.407	43.656	84.563
MB1	100%	4.016	8.094	60.918	63.500
	50%	3.953	7.984	57.383	62.297
	25%	4.109	8.031	57.379	61.875
	10%	3.954	7.985	57.539	61.734
	Cover	3.875	7.812	57.648	61.328

From Table 6, we can clearly observe that for an embedding method, the classifier using fewer features gains a higher speed. For Outguess, F5 and MB1, the average ratios of times consumed by classifiers using selected feature set to that consumed by classifiers using P and F's feature set are 0.1376, 0.1551 and 0.0685, respectively.

The feature numbers of selected three subsets (16, 23 and 15) are far less than the universal set (274). However, the subsets hold better detection accuracy and

higher speed than earlier study, the reason is that the earlier study mainly focus on the extracting features which may be sensitive to the data hiding. Then, lots of features are designed without necessary further analysis and comparison. Some of these features contribute a lot to the accuracy of classifier; however some may only increase the feature dimension. In addition, some features are highly correlation with each other, i.e. redundancy exists among features (Wang and Moulin, 2007). For example, among P and F's features, both dual histogram features and histogram features exploit global and individual histogram characters. These two kinds of features contain much similar information with each other. The combination of these features could not increase the classification performance as they carry the similar information. Genetic algorithm rightly eliminates these redundant and useless features and keeps enough useful features to train a compact classifier. Hence, the trained classifiers have better performance than previous ones in aspect of accuracy and speed. Although, the selected feature subsets provide good detection performance, they are not the unique results for three embedding methods.

The selected features also reveal significance of different kinds of features. Among the selected features, markov features and Co-occurrence Matrix features are in the majority. These two kinds of features exploit the intra-block and inter-block correlation among DCT coefficients respectively. They are expected as the most sensitive and stable features for JPEG image steganalysis. The histogram features are mainly selected for F5 because that Outguess recovers the histogram of the DCT coefficients after embedding data and MB1 holds these statistics by modulating the secret data suitably before embedding the data. For variation V and Block Discontinuities features, although both of them are not selected for any embedding methods, we can not define them as bad features because the amount of these features is comparatively little in the universal set.

Genetic algorithm always selects a feature subset with as few features as possible. It is good for reducing the amount of the features. However, the specially selected features may only hold excellent performance when detecting the target embedding methods. A small quantity of redundancy could increase the stability and universality of feature set to deal with the various steganographic methods and is not expected to influence the accuracy a lot. Therefore, the classifiers are trained using merged feature set (contain 44 features). Experiment results show that these classifiers hold the comparable accuracy to the classifiers that use selected feature sets when detecting the mentioned three steganography methods.

For feature selection based on genetic algorithm, a classifier needs to be trained every time to evaluate a feature subset and the score of the feature subset depends on the accuracy of the classifier when detecting a test set of image. This train-test process is very time-consuming. Therefore, compared with the some other feature reduction methods which do not need the train-test process, such as PCA, the feature selection based on genetic algorithm is much more time-consuming. In addition, HGA will restart when trapped in an unsatisfactory local area. Time may be saved a lot if the GA just rolls back to some stage where the GA does not trap in a local area.

CONCLUSION

In this study, a feature selection method based on hybrid genetic algorithm for image steganalysis has been brought forward. SI and TG are analyzed to determine whether the GA has converged into a local area and appropriate parameters are elaborately configured according to the particularity situation (d<<D). Three different feature subsets are formed for three typical embedding methods in the experiments. Experimental results show that the accuracies of classifiers adopting the selected features are improved and the time expenditures of these classifiers are greatly reduced. It can be concluded that there is no needed to extract more features to obtain better performance and feature selection is an important further work to feature extraction for steganalysis. The designed hybrid genetic algorithm is expected to be applicable to general case of d<<D. For learning-based steganalysis, extracting the informative features is very important. Accordingly, extracting some informative and low-dimensional feature sets for steganalysis is our future plan.

ACKNOWLEDGMENTS

REFERENCES

Avcibas, I., N. Memon and B. Sankur, 2003. Steganalysis using image quality metrics. IEEE Trans. Image Process, 12: 221-229.

Brill, F.Z., D.E. Brown and W.N. Martin, 1992. Fast Genetic Selection of Features for Neural Network Classifiers. IEEE Trans. Neural Networks, 3: 324-328.

Burges, C.J.C., 1998. A tutorial on support vector machines for pattern recognition. Data Min. Knowl. Discov., 2: 121-167.

Chen, C. and Y.Q. Shi, 2008. JPEG image steganalysis utilizing both intrablock and interblock correlations. Proceedings of the IEEE International Symposium on Circuits and Systems, May 18-21, IEEE Computer Society Press, Washington, USA., pp: 3029-3032.

Farid, H., 2002. Detecting hidden messages using higher-order statistical models. Proceedings of the IEEE International Conference on Image Processing, Sept. 22-25, IEEE Computer Society Press, New York, USA., pp: 905-908.

Farid, H. and S. Lyu, 2003. Detecting hidden messages using higher-order statistics and support vector machines. Proceedings of the 5th International Information Hiding Workshop LNCS, Berlin, Oct. 7-9, Springer-Verlag, pp: 340-354.

Fridrich, J., 2004. Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes. Proceeding of the 6th Information Hiding Workshop, LNCS 3200, May 23-25, Springer-Verlag, Toronto, Canada, pp: 67-81.

Holland, J.H., 1992. Adaptation in Nature and Artificial Systems. MIT Press, Michigan, USA., ISBN: 978-0-262-58111-0..

Holotyak, T., J. Fridrich and S. Voloshynovskiy, 2007. Blind statistical steganalysis of additive steganography using wavelet higher order statistics. Proceedings of the 4th International Symposium on Neural Networks, Jun. 3-7, Springer-Verlag, Nanjing, China, pp: 273-284.

Kudo, M. and J. Sklansky, 2000. Comparison of Algorithms that Select Features for Pattern Recognition. Pattern Recognit., 33: 25-41.

Luo, X.Y., D.S. Wang, P. Wang and F.L. Liu, 2008. A review on blind detection for image steganography. Signal Process., 88: 2138-2157.

Miche, Y., B. Roue, A. Lendasse and P. Bas, 2006. A feature selection methodology for steganalysis. Proceedings of the International Workshop on Multimedia Content Representation, Classification and Security, LNCS 4507, Sept. 11-13, Springer-Verlag, Istanbul, Turkey, pp: 49-56.

Pevný, T. and J. Fridrich, 2007. Merging markov and DCT features for multi-class JPEG steganalysis. Proceedings of the SPIE, Electronic Imaging, Security, Steganography and Watermarking of Multimedia Contents IX, Jan. 29, SPIE, San Jose, California, pp: 301-313.

Provos, N., 2001. Defending against statistical steganalysis. Proceedings of the 10th Usenix Security Symposium, Aug. 13-17, USENIX Assoc, Washington, DC, USA, pp: 323-335.

Raymer, M.L., W.F. Punch, E.D. Goodman, L.A. Kuhn and A.K. Jain, 2000. Dimensionality reduction using genetic algorithms. IEEE Trans. Evol. Comput., 4: 164-171.

Sallee, P., 2003. Model based steganography. Proceedings of the 2th International Workshop Digital Watermarking, Oct. 20-22, Springer-Verlag, Seoul, Korea, pp: 174-188.

Shi, Y.Q., C. Chen and W. Chen, 2006. A markov process based approach to effective attacking JPEG steganography. Proceedings of the 8th Information Hiding Workshop, Jul. 10-12, Springer-Verlag, Brittany, France, pp: 249-264.

Siedlecki, W. and J. Sklansky, 1989. A note on genetic algorithms for large-scale feature selection. Pattern Recognit. Lett., 10: 335-347.

Wang, Y. and P. Moulin, 2007. Optimized feature extraction for learning-based image steganalysis. IEEE Trans. Inform. Forensics Secur., 2: 31-45.

Westfeld, A., 2001. F5 a steganographic algorithm: High capacity despite better steganalysis. Proceedings of the 4th Information Hiding Workshop, Apr. 25-27, Springer-Verlag, Pittsburgh, PA, USA, pp: 289-302.

Xuan, G.R., Y.Q. Shi, J.J. Gao, D.K. Zou and C.Y. Yang *et al.*, 2005. Steganalysis based on multiple features formed by statistical moments of wavelet characteristic functions. Proceedings of the 7th International Information Hiding Workshop, LNCS 3727, Jun. 6-8, Springer-Verlag, Berlin, pp:262-277.