EVOLUTIONARY RULE GENERATION FOR SIGNATURE-BASED COVER SELECTION STEGANOGRAPHY

Hedieh Sajedi, Mansour Jamzad

Abstract: A novel approach for selecting proper cover images in steganography is presented in this paper. The proposed approach consists of two stages. The first stage is an evolutionary algorithm that extracts the signature of cover images against stego images in the form of fuzzy if-then rules. This algorithm is based on an iterative rule learning approach to construct an accurate fuzzy rule base. The rule base is generated in an incremental way by optimizing one fuzzy rule at a time using an evolutionary algorithm. In the second stage of the proposed approach, the fuzzy rules generated in the first stage are used for selecting suitable cover images for steganography. We applied our approach to some state-of-the-art steganography techniques and validated it using an image database. The results indicate that a secret message can be securely embedded in selected cover images. Therefore, we can apply the proposed evolutionary fuzzy algorithm, as an intelligent rule generation approach, to select the appropriate cover images from an image database and use them to have more secure steganography.

Key words: Steganography, cover selection, evolutionary fuzzy systems, steganalysis

Received: March 20, 2009
Revised and accepted: February 10, 2010

1. Introduction

Steganography is the science of imperceptible communications. It is used sometimes together with cryptography to guard information from unwanted third parties. In contrast with cryptography, where the attacker is able to identify, catch, and change the transmitted information [1], steganography is used primarily when the fact of communicating needs to be kept covert. This is accomplished by embedding the secret messages within apparently innocuous covers. Nowadays, typical covers are computer files, mainly image, video and audio files [2].

Hedieh Sajedi, Mansour Jamzad
Department of Computer Engineering, Sharif University of Technology, E-mail: a_sajedi@ce.sharif.edu, jamzad@sharif.edu

©ICS AS CR 2010 297
The well-known and ancient steganographic methods include covert channel, invisible ink, microdot, and spread-spectrum communication [3-4]. A famous classic steganographic model presented by Simmons [5] is the prisoners’ problem that Alice and Bob in a jail, plan to escape together. All communications between them are monitored by a warden, Wendy. Therefore, they must hide the messages in other innocuous-looking media (cover object) in order to obtain the stego object. Then, the stego object is sent through the public channel. For a more complete description of uses and applications of steganography, see [5, 6].

For steganographic systems, the fundamental requirement is that the stego object should be perceptually indistinguishable to the degree that it does not raise suspicion. In other words, the hidden information should introduce only slight modification to the cover object [7].

Different image steganography methods have been proposed in the literature. Embedding techniques in Discrete Cosine Transform (DCT) domain are popular because of the vast usage of JPEG images. Embedding in a number of steganography methods like F5 [8], Model-based (MB) [9], Perturbed Quantization (PQ) [10], and YASS [11] is done by modifications of properly selected DCT coefficients. Some other methods have been proposed which embed secret messages in other transform domains, such as Contourlet-based steganography method [12] that embeds secret messages in contourlet coefficients of a cover image.

Different images have various properties due to their different contents. Therefore, corresponding stego images may have different grades of visual and statistical undetectability. Consequently, a steganography method can employ a cover selection technique to choose a cover image that after data hiding does not attract the attention of steganalysis methods, and make steganalyzers to misclassify the observed stego image as a clean image.

In this paper, we present a novel steganography approach that consists of two stages. In the first stage, we analyze an image database to discover the pattern or signature of cover (clean) images. By the signature, we mean the effective features of clean images and their relative values. This signature is constructed in the form of a set of fuzzy if-then rules that represent the similarity between clean images. In the second stage, the steganographer selects proper cover images according to the generated signature of clean images. A proper cover image is the one that after data hiding stimulates the generated fuzzy rules significantly and obeys the signature of clean images.

The process of generating the signature of clean images is done by an Evolutionary Algorithm (EA). Evolutionary algorithms have been used as rule generation and optimization tools in the design of fuzzy rule-based systems [13, 14].

To obtain accurate fuzzy rules, we present an evolutionary rule generation algorithm based on Iterative Rule Learning (IRL) approach. The rules are generated incrementally in that the evolutionary algorithm optimizes one fuzzy rule at a time. The generated fuzzy rules are then used as the signature of clean images in cover selection steganography. We applied our cover selection approach to MB, PQ, and YASS steganography techniques and validated it using an image database. The results illustrate that evaluating cover images based on the signature of clean images and selection of a proper cover image for embedding reduce the detection accuracy of state-of-the-art steganalysis methods considerably.
The rest of the paper is organized as follows: Related works are introduced in Section 2. Our signature-based cover selection approach is presented in Section 3. Experimental results are reported in Section 4, and finally Section 5 concludes this paper.

2. Related Works

2.1 Cover selection steganography

The cover object in steganography acts only as a carrier for secret messages. Therefore, the embedder is allowed to choose any cover images from the database using a cover selection steganography. Cover selection steganography is a technique that tries to find the best cover image from a database to embed a given secret message. In this respect, cover images are retrieved, based on a measure like undetectability of the corresponding stego images. Retrieval of proper cover images from the database based on an efficient measure can improve the security of stego images produced by a certain steganography method.

Cover selection steganography can offer some ranked images in order to help the embedder to decide whether to transmit the best stego image, or to select an alternative one. In this way, the embedder could choose a cover image so that the humans and steganalyzers would misclassify the resulted stego image. Therefore, the embedder can reduce the detectability of a given secret message by choosing an appropriate cover image to host it. In the sequel, we shortly review the existing cover selection steganography methods.

A cover selection technique for hiding a secret image in a cover image was first introduced in [15]. This method operates based on image texture similarity and replaces some blocks of a cover image with similar secret image blocks; then, indices of secret image blocks are stored in the cover image. In this cover selection method, the blocks of the secret image are compared with the blocks of a set of cover images and the image with most similar blocks to those of the secret image is selected as the best candidate to carry the secret image. An improvement on this method is proposed in [16] that uses statistical features of image blocks and their neighborhoods. Using block neighborhood information prevents appearance of virtual edges in the sides and corners of the replaced blocks. In [17], the cover selection problem was studied by investigating three scenarios in which the embedder has either no knowledge, partial knowledge, or complete knowledge of the steganalysis method. In addition, some measures for cover selection were introduced in [17] as follows: Cardinality of changeable DCT coefficients; JPEG quality factor of images; Number of modifications of a cover image; Mean square error (MSE) obtained from cover-stego image pairs; Local prediction error, which is the difference between the mean prediction error of the cover and stego images; Watson’s metric [18] that is used for quantifying the quality of JPEG images and Structural Similarity Measure (SSIM) [19] which is quantifying the similarity between the cover and stego images. The works in [15, 16] assumed that the secret object is an image. In this paper, we consider binary bit sequences with random distribution as secret data.

One of the most recent researches in steganography was introduced by us in [20], where we presented a technique to measure the embedding capacity of cover
images and then utilized this measurement as a criterion for cover selection. This technique uses an ensemble system, which applies different steganalyzer units to determine the embedding capacity of cover images. Each steganalyzer unit is formed by a combination of multiple steganalyzers from the same type, but each steganalyzer is trained to detect stego images with an equal certain payload. In [20], after determining the embedding capacity of a cover image, an embedding routine embeds in the image a secret data with size of less than or equal to the embedding capacity of the cover image. Experiments of this research investigated the security of produced stego images against steganalyzers. The results illustrated that considering the embedding capacity of cover images enhances the security of stego images comparing to the usual usage of steganography methods. Nevertheless, calculation of embedding capacity followed by embedding is more time-consuming than the usual steganography methods. In this paper, we tend to decrease the time of embedding process and enhance the security of stego images by evaluating the consistency of stego images properties with the signature of clean images during embedding. This signature is obtained via an Evolutionary algorithm.

2.2 Image feature selection

To detect the very existence of a hidden message in a stego image and discover the information-hiding behaviors in steganography, various steganalysis methods consider different features of images. In steganalysis, few publications deal with feature selection [17].

Generally, feature selection can be grouped into two categories: filtering and wrapper methods. Filtering methods select feature subsets independently from the learning classifiers and do not incorporate learning [18, 19]. The weakness of filtering methods is that they just consider the individual feature in isolation and ignore the possible interaction of features among them. Yet, the combination of these features may have a combined effect that does not necessarily follow from the individual performance of features in the group. However, if there is a limit on the number of features to be chosen, all the informative features may not be included. Wrapper methods wrap around a particular learning algorithm that can assess the selected feature subsets in terms of estimated classification errors and then build the final classifiers [21].

Using too many features is undesirable in terms of classification performance due to the curse of dimension [22]. Furthermore, performing feature selection in steganalysis offers several advantages [23] as follows:

1) Pruning the meaningless features for the classifier.
2) Improvement of classification performance.
3) Reducing the complexity for both feature generating and classifier training.
4) Help to point out the features that are sensitive to a given steganographic scheme and consequently highlight its weaknesses.

Hence, it is necessary to reduce the feature dimension by eliminating redundant features and selecting the most relevant ones.
In this paper, to find the effective features of images and their relative values we propose a learning approach in Section 3 to find accurate and interpretable fuzzy if-then rules.

2.3 Fuzzy rule generation

The studies on evolutionary algorithms in design of fuzzy rule-based systems are usually referred to as Evolutionary Fuzzy Systems (EFS), each of which can be classified into Michigan, Pittsburgh, or Iterative Rule Learning approaches [13]. Some studies are categorized as Michigan approach where a single fuzzy if-then rule is coded as an individual [24, 25]. Many EFS methods are categorized as Pittsburgh approach where a set of fuzzy if-then rules is coded as an individual [26, 27]. In the third approach, Iterative Rule Learning, each individual codes one rule and in every run of Genetic Algorithm (GA) a new rule is adapted and added to the rule set, in an iterative way [28, 29].

The most important characteristic of EFS that inspires us to use them as a rule generation tool is their impressive capability to produce accurate and interpretable knowledge [13]. Note that in other learning systems like Artificial Neural Networks, Naïve Bayes, k-Nearest Neighbors, and Support Vector Machines (SVM), the final Knowledge Base (KB) does not introduce effective features from input samples. However, in EFS the expert can understand and interpret the generated KB that is a fuzzy rule base in our work.

In this paper, an evolutionary rule generation algorithm, which is based on IRL approach, is presented. The generated fuzzy rule base is used to form the signature of clean images for cover selection in steganography. We will discuss the details of our approach in the following section.

3. Signature-based Cover Selection

In this paper, we utilize an iterative evolutionary fuzzy algorithm for cover selection steganography. In this utilization, the proposed algorithm extracts a fuzzy rule base for secure steganography problem. Introducing the signature of clean images based on fuzzy rules and defining ‘covering ability’ property for cover images are the main contributions of this paper. In other words, extracting fuzzy rules and using them in cover selection steganography has not been investigated in previous steganographic works. In this regard, we introduce covering ability of images as a new measure that can be used for cover selection. Consequently, to have a secure covert communication, we can select cover images from the database that do not deviate from the clean images signature, and their covering ability are high. It should be noted that the embedding procedure could be carried on by any steganography method.

Fig. 1 shows the block diagram of cover selection steganography based on signature of clean images. Following subsections describe the details of this approach.
3.1 Feature vector generation

This section deals with the generation of feature vector from the image database. We use a 622-dimensional feature vector, which is produced by appending the features of three efficient and well-known steganalyzers. Tab. 1 shows that the 622 features are computed according to the features of Pevny-Fridrich [30], Chen [31] and Lyu-Farid [32] steganalysis methods.

<table>
<thead>
<tr>
<th>Steganalysis Method</th>
<th>Number of Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pevny-Fridrich [30]</td>
<td>274</td>
</tr>
<tr>
<td>Chen [31]</td>
<td>324</td>
</tr>
<tr>
<td>Lyu-Farid [32]</td>
<td>24</td>
</tr>
</tbody>
</table>

Tab. 1 Three groups of 622 image features.

In the following, we briefly review the features, which are used by these steganalyzers.

1.) Pevny and Fridrich [30] extract 274 features by merging 193 extended DCT features with 81 averaged calibrated Markov features. However, many of the 274 features may be highly correlated to each other. In this method, Markov features model intra-block DCT dependencies and DCT features (first 193 features) model inter-block relations. In the rest of this paper, we refer to this steganalysis method as 274-dim steganalyzer.

Fig. 1 The block diagram of signature-based cover selection steganography.

2.) In [31], Chen proposed a steganalysis method that employs a 324-dimensional feature vector for analysis. It is based on statistical moments derived from both image 2-D array and JPEG 2-D array. This steganalyzer considers both the first order and the second order histograms. Consequently, the moments of 2-D characteristic functions are also used for steganalysis. In the following, this method is referred to as 324-dim steganalyzer.
3.) Wavelet-based steganalysis method [32], presented by Lyu and Farid, builds a model for clean images by using higher order statistics, and then shows the deviation of stego images from the constructed model. Quadratic Mirror Filters (QMF) are used to decompose the image into wavelet domain, after which higher order statistics such as mean, variance, skewness, and kurtosis are calculated for each subband. The higher order statistics are calculated from wavelet coefficients of each high-frequency subband to form one group of features. Another group of features is similarly formulated from the prediction errors of wavelet coefficients of each high-frequency subband. We called this method 24-dim steganalyzer.

The types of all 622 features are given in Tab. II. In 324-dim steganalyzer, the first three moments of Discrete Fourier Transform (DFT) of all feature types are considered. We have normalized the feature values into unit interval [0,1] in order to use the same membership function for them in the fuzzy rule generation stage. In the next subsection, this step is explained in detail.

<table>
<thead>
<tr>
<th>Feature Group</th>
<th>Number of Features</th>
<th>Feature Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>274</td>
<td>11</td>
<td>Global Histogram</td>
</tr>
<tr>
<td></td>
<td>66</td>
<td>5 AC Histograms</td>
</tr>
<tr>
<td></td>
<td>99</td>
<td>11 Dual Histograms</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Variation</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Blockiness</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>Co-occurrence Matrix</td>
</tr>
<tr>
<td></td>
<td>81</td>
<td>Markov Features</td>
</tr>
<tr>
<td>324</td>
<td>39</td>
<td>Histogram of Spatial Representation and Discrete Wavelet Transform (DWT) Representation</td>
</tr>
<tr>
<td></td>
<td>39</td>
<td>Histogram of Prediction Error and DWT of Error</td>
</tr>
<tr>
<td></td>
<td>39</td>
<td>Histogram of JPEG Representation and its DWT</td>
</tr>
<tr>
<td></td>
<td>78</td>
<td>Horizontal 2-D Histogram of JPEG Representation and its DWT 2-D Histogram</td>
</tr>
<tr>
<td></td>
<td>78</td>
<td>Vertical 2-D Histogram of JPEG Representation and its DWT 2-D Histogram</td>
</tr>
<tr>
<td></td>
<td>78</td>
<td>Diagonal 2-D Histogram of JPEG Representation and its DWT</td>
</tr>
<tr>
<td></td>
<td>39</td>
<td>Histogram obtained from Prediction Error of JPEG Representation and its DWT</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>Higher order statistics of each Wavelet subband</td>
</tr>
</tbody>
</table>

Tab. II Types of 622 image features.

3.2 Fuzzy rule generation

This subsection deals with the generation of fuzzy if-then rules from the feature vectors of images, which are prepared as mentioned in previous subsection. Each fuzzy if-then rule in this paper is coded as a string, and the following symbols are
used for denoting the six linguistic values (Fig. 2): 1: don’t care (DC), 2: small (S), 3: medium small (MS), 4: medium (M), 5: medium large (ML), 6: large (L).

The fuzzy rules generated in this paper are as follows:

Rule $R_j$: If $(x_1$ is $A_{j1}$ and \ldots and $x_n$ is $A_{jn}$) then Image is clean with $CF = CF_j$,

where $R_j$ is the label of the $j^{th}$ fuzzy if-then rule, $x_1$, \ldots, $x_n$ are the features which are extracted from the observed image. $A_{j1}$, \ldots, $A_{jn}$ are values in $[0, 1]$ that represent S, MS, M, ML, L, and DC as shown in Fig. 2. $CF_j$ is the certainty grade of the fuzzy if-then rule $R_j$. Each fuzzy rule has a certainty grade that demonstrates the confidence of the rule about its antecedent part.

The membership function of each linguistic value in Fig. 2 is specified by homogeneously partitioning the domain of each feature into symmetric triangular fuzzy sets. However, we can use other tailored membership functions in our fuzzy algorithm.

The total number of possible fuzzy if-then rules is $6^n$ (due to using six linguistic values) in case of $n$-dimensional feature vector. It is impossible to use all the $6^n$ fuzzy if-then rules in a single fuzzy rule base for large $n$ (e.g. steganography cover selection based on $n = 622$ features). Therefore, our proposed evolutionary method searches for a relatively small number of fuzzy if-then rules (e.g., 10 rules) with high performance. By performance we mean that the generated fuzzy if-then rules should be able to show the pattern or signature of clean images with high accuracy. This signature is extracted according to the training samples of clean and stego images.

![Fig. 2](image)

**Fig. 2** Fuzzy sets that are used in this paper. S: small, MS: medium small, M: medium, ML: medium large, L: large, and DC: don’t care.

We apply the following three steps to calculate the certainty grade of each fuzzy if-then rule:

*Step 1:* Calculate the compatibility of each training sample $x_p = (x_{p1}, x_{p2}, \ldots, x_{pn})$ with the fuzzy if-then rule $R_j$ by the following product operation:
\[ \mu_j(x_p) = \mu_{j1}(x_{p1}) \times \ldots \times \mu_{jn}(x_{pn}), \quad P=1,2,\ldots,M, \quad (1) \]

where \( \mu_{ji}(x_{pi}) \) is the membership function of \( i^{th} \) feature of \( p^{th} \) sample and \( M \) denotes the total number of samples.

**Step 2:** For clean and stego images, calculate the relative sum of the compatibility grades of training samples with fuzzy if-then rule \( R_j \):

\[
\beta_{\text{Clean}}(R_j) = \frac{\sum_{x_p \in \text{Clean}} \mu_j(x_p)}{N_{\text{Clean}}}, \quad (2)
\]

\[
\beta_{\text{Stego}}(R_j) = \frac{\sum_{x_p \in \text{Stego}} \mu_j(x_p)}{N_{\text{Stego}}}, \quad (3)
\]

where \( \beta_{\text{Clean}}(R_j) \) and \( \beta_{\text{Stego}}(R_j) \) are the relative sum of the compatibility grades of training samples that represent clean and stego images, respectively. Note that \( N_{\text{Clean}} \) and \( N_{\text{Stego}} \) represent the number of clean and stego images that are being used as the training samples.

**Step 3:** The grade of certainty \( CF_j \) is determined as follows:

\[
CF_j = \frac{(\beta_{\text{Clean}}(R_j) - \beta_{\text{Stego}}(R_j))}{(\beta_{\text{Clean}}(R_j) + \beta_{\text{Stego}}(R_j))}. \quad (4)
\]

By the proposed heuristic procedure, we can specify the certainty grade for any combination of antecedents in a fuzzy if-then rule. Such a combination is generated by the proposed evolutionary fuzzy algorithm.

In the next subsection, we will discuss about our evolutionary fuzzy algorithm in detail.

### 3.3 Proposed evolutionary fuzzy algorithm

Our proposed evolutionary fuzzy algorithm learns fuzzy if-then rules in an iterative way by optimizing one fuzzy rule in each iteration of the algorithm. Initially, all the training samples get the same weight and each individual in the algorithm is initialized by the feature vector of an image. In each iteration of the algorithm, the if-then rule with highest fitness is considered as the output of the iteration. Then, the learning mechanism reduces the weight of those training samples that are learned correctly. Samples with higher weight are more significant in the training process. Therefore, the next rule generation cycle searches for fuzzy rules that account for the currently training samples, which are uncovered by the rules obtained in previous iterations. In brief, the fuzzy rules that cover the distribution of training samples more than other rules are included in the final rule base.

The idea behind using the boosting mechanism (reducing the weight of training samples) is to aggregate different disciplines between features of training samples to form a perfect fuzzy rule base. In the above learning framework, we have used
a fitness function in evolutionary process, which is computed according to the equations (5) to (7).

\[ f_P = \frac{\sum_{x^k \in \text{Clean}} w^k \mu_{R_i}(x^k)}{\sum_{x^k \in \text{Clean}} w^k} \]  

(5)

\[ f_N = \frac{\sum_{x^k \in \text{Stego}} w^k \mu_{R_i}(x^k)}{\sum_{x^k \in \text{Stego}} w^k} \]  

(6)

\[ \text{fitness}(R_j) = w_P f_P - w_N f_N, \]  

(7)

where

- \( f_P \): is the rate of positive training samples covered by rule \( R_i \) (correctly covered).
- \( f_N \): is the rate of negative training samples covered by rule \( R_i \) (wrongly covered).
- \( w^k \): is a weight which reflects the frequency of the sample \( x^k \) in the training database.
- \( w_P \): is the weight of rule’s positive power.
- \( w_N \): is the weight of rule’s negative power.

The outline of the proposed iterative evolutionary fuzzy method is presented as follows:

Step 1: Generate an initial population of fuzzy if-then rules based on weight of training samples (Initialization).

Step 2: Generate new fuzzy if-then rules by genetic operations (Generation).

Step 3: Replace a part of the current population with the newly generated rules (Replacement).

Step 4: Terminate the inner cycle (Step 2 and Step 3) of the algorithm if a stopping condition is satisfied, otherwise go to Step 2 (Inner Cycle Termination Test).

Step 5: Terminate the outer cycle (Step 1 to Step 6) if a stopping condition is satisfied, otherwise go to Step 6 (Outer Cycle Termination Test).

Step 6: Reduce the weight of training samples that cover the new obtained fuzzy rule. Go to Step 1 (Weight Adjustment).

The output of each outer cycle of the above algorithm is one fuzzy if-then rule. The outline of the proposed evolutionary rule generation algorithm is presented in Fig. 3.
Step 1: Let us denote the number of fuzzy if-then rules in the population of genetic algorithm by $N_{pop}$. To produce an initial population, $N_{pop}$ fuzzy if-then rules are generated according to the features of random samples in the training database, which is introduced in Experiment Section. Note that the probability for each training sample to be chosen in this step is proportional to its current weight. This means that the algorithm considers a greater probability for those samples that have not been learned in previous iterations. Next, for these random samples, we determine the most compatible combination of antecedents in if-then rules using only six linguistic values, as shown in Fig. 2. The compatibility of antecedents with features of random a sample is measured by equation (1). The certainty grade of each fuzzy if-then rule is determined according to the heuristic method, explained in previous section. After generation of $N_{pop}$ fuzzy if-then rules, the fitness value of each rule is evaluated by classifying all the given training samples using the set of fuzzy if-then rules in the current population. Each fuzzy if-then rule is evaluated according to the fitness function, which is presented in equation (7).
Step 2: A pair of fuzzy if-then rules is selected from the current population to generate new fuzzy if-then rules for the next population. Each fuzzy rule in the current population is selected using the tournament selection strategy. This procedure is iterated until a prespecified number of pairs of rules are selected. A crossover operation is then applied to a selected random pair of fuzzy rules with a prespecified crossover probability. In computer simulations of this algorithm, we have used the uniform crossover. With a prespecified mutation probability, each antecedent of fuzzy if-then rules is randomly replaced with a different antecedent fuzzy set after the crossover operation. The probability of changing to don’t care value is more than the other five linguistic values. We call this probability $P_{DC}$. After performing selection, crossover, and mutation steps, the fitness value of each of the generated rules is evaluated according to equation (7).

Step 3: A prespecified number of rules in the current population are replaced with the newly generated rules. In our fuzzy classifier, $P_R$ percent of the worst rules with the smallest fitness values are removed from the current population and $(100 - P_R)$ percent of the newly generated fuzzy if-then rules are added ($P_R$ is the replacement percentage). After performing the mentioned replacement procedure, the fitness value of each of the individuals is evaluated according to equation (7).

Step 4: We can use any stopping condition for terminating the inner cycle of the rule-learning algorithm. In our computer simulations, we used the total number of generations as a stopping condition.

Step 5: After termination of the inner cycle, the algorithm adds the best fuzzy rule of the evolved population to the final classification rules list and checks if this added fuzzy rule is capable of improving the classification rate of final classification system. If the classification rate is not improved, the algorithm removes the added fuzzy rule from the final rule base and terminates. Otherwise, it goes to next step.

Step 6: In each iteration of the main evolutionary process, rule $R_t$ with the best fitness value is inserted into the primary fuzzy rule base. After each rule extraction process, samples that are misclassified will end up having the same weight. The weight of those instances that are classified correctly will become zero. Note that initially $w^k = 1$. After this step, the algorithm jumps to step 1.

3.4 Proposed cover selection steganography

The steganographer can search the entire database to find the best cover image or sequentially searches until it finds an acceptable cover image that results in an undetectable stego image according to the clean image signature. Acceptable cover images were found in our experiments.

As the Fig. 1 shows, each stego image database and clean image database are fed to the evolutionary fuzzy rule generation stage (we set the parameters of MB, PQ, and YASS to construct stego image databases with variety of payloads). After
this stage, a clean image rule set is resulted considering the effects of steganography method on images. We have three types of stego image databases (MB, PQ, and YASS), therefore three types of rule sets are generated. Putting all the rules of clean images in a clean image rule base, we define ‘covering ability’ function that is used for evaluation of cover images as a selection measure.

For a given rule base \( S \), in order to determine whether or not a clean image with feature vector \( x_{pc} = (x_{p1}, x_{p2}, \ldots, x_{pn}) \) is reliable to host a secret message, two parameters \( \tau_{Clean} \) and \( \tau_{Stego} \) are computed using equations (8) and (9). After hiding a secret message in the cover image, \( \tau_{Stego} \) is computed based on the features \( x_{ps} = (x_{ps1}, x_{ps2}, \ldots, x_{psn}) \) of the produced stego image.

\[
\tau_{Clean} = \sum_{R_j \in S} \mu_j(x_p) \cdot CF_j \quad (8)
\]

\[
\tau_{Stego} = \sum_{R_j \in S} \mu_j(x_{ps}) \cdot CF_j \quad (9)
\]

\[
\tau_{Clean} > \tau_{Stego} \quad (10)
\]

Generally, equation (10) is valid for a clean image and its stego version because the rule base \( S \) contains rules that are achieved from features of clean images. However, these rules are mostly true for clean images compared to their stego versions.

We define covering ability function that is computed for each image as equation (11):

\[
\text{covering ability}(\tau) = \frac{1}{\tau_{Clean} - \tau_{Stego}}. \quad (11)
\]

If \( \tau \geq \text{threshold} \ T \), it means that, the clean image is appropriate to be selected as the cover image since it has an acceptable covering ability. \( T \) denotes the sensibility rate of the steganography method.

4. Experimental Results

To evaluate performance of the proposed approach, different experiments were done. We obtained 1000 JPEG images from Washington University image database [33]. All images were converted to grayscale images of size 512×512. To make stego image databases, MB, PQ, and YASS steganography methods are employed. We set the parameters of these methods to different values to obtain three stego image databases with a variety of payloads. Each stego database has 1000 stego images.

We quantified the steganalysis performance according to the detection accuracy [11]. The SVM classifier is used to distinguish between two classes: cover (class ‘0’) and stego (class ‘1’) images. Let \( X_0 \) and \( X_1 \) denote the events that the image being observed belongs to classes ‘0’ and ‘1’, respectively. On the detection side, let \( Y_0 \) and \( Y_1 \) denote the events that the observed image is classified as belonging to classes ‘0’ and ‘1’, respectively. We use detection accuracy \( (D_a) \) which is the percent of detection probability \( (P_d) \) as our evaluation criteria according to the following equations:
where $P_{FA} = P(Y_1|X_0)$ and $P_{miss} = P(Y_0|X_1)$ denote the probability of false alarm and missed detection, respectively. The above equation assumes an equal number of cover and stego images in the dataset.

An uninformed detector can classify all the test images as stego (or cover) and get an accuracy of 50. Thus, $D_a$ being close to 50 implies nearly undetectable hiding, and as the detectability improves, $D_a$ increases towards 100.

We normalized the train and test features extracted from the databases, where each numerical value in the data set is normalized between 0 and 1 according to the following equation:

$$x_{normalized} = \frac{x - x_{min}}{x_{max} - x_{min}}.$$  

(13)

Hence, 622 numeric features are constructed and normalized to the interval [0, 1].

Tab. III shows parameter specification that we have used in our computer simulations for the presented evolutionary fuzzy rule generation algorithm.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>population size ($N_{pop}$)</td>
<td>500</td>
</tr>
<tr>
<td>don’t care replacement rate ($P_{DC}$)</td>
<td>0.5</td>
</tr>
<tr>
<td>crossover probability ($P_X$)</td>
<td>0.9</td>
</tr>
<tr>
<td>mutation probability ($P_M$)</td>
<td>0.5</td>
</tr>
<tr>
<td>fitness positive weight ($W_P$)</td>
<td>0.1</td>
</tr>
<tr>
<td>fitness negative weight ($W_N$)</td>
<td>0.9</td>
</tr>
<tr>
<td>replacement percentage ($P_R$)</td>
<td>10</td>
</tr>
<tr>
<td>maximum generations of the iterative algorithm</td>
<td>150</td>
</tr>
</tbody>
</table>

Tab. III Parameters specification in computer simulations.

4.1 Time complexity evaluation of the proposed method

The experimental results are carried out on a 2046 MB PIV processor using MATLAB 7.6.0. The proposed approach consists of two stages. The first stage is extracting the signature of clean images in the form of fuzzy if-then rules. This signature is obtained using a clean image database and three stego image databases that were produced by MB, PQ, and YASS steganography methods. In other word, we performed rule extraction using four image databases (three stego image databases and one clean image database). Extracting the rules from the pair of YASS stego
image database and clean image database is effortless. These rules are achieved via a small number of evolutionary algorithm iterations (e.g., 30 iterations). Conversely, extracting the rules from MB and PQ stego images is more complicated, and the evolutionary algorithm requires a larger number of iterations (e.g., 140 iterations). In our implementation, the whole time of clean images signature extraction takes around 8 hours. In the second stage of the proposed approach, the signature generated in the first stage is used for selecting suitable cover images for steganography. Using our proposed method, we hid some randomly generated secret messages and recorded the embedding times. For embedding each secret message, the covering ability of the cover image was calculated. Afterward, the secret data that its size is related to the computed covering ability was hidden in the image. The most time-consuming part of this process is the covering ability computation of the cover image.

Due to the differences in contents of various images, the embedding time may be different. Tab. IV demonstrates the average embedding time of 200 secret messages in our implementation environment. The size of secret messages varies from 1000 to 10000 bits. Using our proposed method, the average time for embedding in one image is less than 1 minute, while the embedding time for the steganography method proposed in [20] is about 2 minutes. Our proposed method only checks few rules (e.g., 10 rules) for computing the covering ability of each cover image, whereas the method in [20] investigates 622 features of three steganalyzers to compute the embedding capacity of the image. Although the time required for our proposed technique is more than the usual embedding techniques, but assuming the steganography as an offline process and the fact that our method provides more secure stego images, the higher time complexity can be acceptable. Since the main goal of steganography is to hide the secret data securely and if any of the steganalyzers gets suspicious, the purpose of steganography is broken, therefore it is worth to spend more time to construct more secure stego images.

<table>
<thead>
<tr>
<th>Steganography Method</th>
<th>MB</th>
<th>PQ</th>
<th>YASS</th>
<th>Cover Selection introduced in [20]</th>
<th>Signature-based Cover Selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time Average Time</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>113 98 71</td>
<td>53 45 26</td>
</tr>
</tbody>
</table>

Tab. IV Average embedding time of different steganography methods (in seconds).

4.2 Security evaluation of the proposed method

Employing signature-based cover selection steganography, the steganographer can search the entire cover image database and selects an image that its corresponding stego version is more undetectable than other images. Since this strategy takes a lot of time, the steganographer can find an acceptable cover image instead. To find an acceptable cover, the secret message is embedded in an image and the resulted stego image is evaluated according to the signature of clean images. If the stego image deviates the signature significantly, the cover image is not suitable and the
steganographer can switch to the next cover image. In this way, a proper cover image can be found very quickly.

To investigate the relation between covering ability of cover images (τ) and the detection rates of steganalyzers, we adjust the threshold of covering ability to different values. Then, we measure the average detection rates of three used steganalyzers on YASS, PQ, and MB steganography methods.

The results in Fig. 4 show that detection accuracy has an inverse relation with covering ability of cover images (τ) in all the cases. Therefore, an image with higher covering ability value results in a more undetectable stego image.

We expect that the stego images that have behavior similar to the clean images, get less change compared to other stego images. To investigate this issue, we did an experiment to measure the quality of stego images in Peak-Signal-to-Noise-Ratio (PSNR) by changing the threshold of covering ability of images. Fig. 5 shows that as the covering ability increases, the average quality of stego images improves too.

Tab. V shows the detection accuracy of three steganalysis methods on MB, PQ, and YASS steganography methods when used without cover selection (the first 3 columns of the table), the cover selection steganography method proposed in [20], and the cover selection steganography method based on signature of clean images (the last three columns).

As the results demonstrate, signature-based cover selection approach improves the security of stego images. Consequently, we can comprehend that our proposed signature-based cover selection approach enhances the performance of steganography methods considerably compared to the usual usage of these methods. Comparing the detection accuracy of steganalysis methods in detection of stego images produced by our method and the method presented in [20] shows that the former method provides more or equal security for stego images than the latter.

<table>
<thead>
<tr>
<th>Steganography Method</th>
<th>MB</th>
<th>PQ</th>
<th>YASS</th>
<th>MB</th>
<th>PQ</th>
<th>YASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>274-dim</td>
<td>70</td>
<td>74</td>
<td>67</td>
<td>65</td>
<td>69</td>
<td>61</td>
</tr>
<tr>
<td>324-dim</td>
<td>92</td>
<td>54</td>
<td>55</td>
<td>64</td>
<td>59</td>
<td>66</td>
</tr>
<tr>
<td>24-dim</td>
<td>75</td>
<td>73</td>
<td>60</td>
<td>56</td>
<td>57</td>
<td>59</td>
</tr>
<tr>
<td>Average</td>
<td>79</td>
<td>74</td>
<td>61</td>
<td>61</td>
<td>58</td>
<td>60</td>
</tr>
</tbody>
</table>

Tab. V *The relation between PSNR and covering ability of cover images (τ) in cover selection.*

5. Conclusions

Images have various properties due to their different contents. Therefore, for a certain secret message, cover images could result in stego images with unequal degree of undetectability. In this paper, to increase the security of stego images, we proposed a novel cover selection measure based on signature of clean images, which is achieved by analyzing the similarity between features of clean images. In
Fig. 4 The relation between detection rate and covering ability of cover images ($\tau$) in cover selection using (a) YASS, (b) PQ, and (c) MB steganography methods.

This regard, a new evolutionary fuzzy algorithm is proposed to generate fuzzy rules from features of clean images. These rules are used to form the signature of clean images.
After discovering the signature of clean images, in the next step, the steganographer can select the proper cover images from the database. A proper cover image is the one that after embedding, its effective features do not deviate from the signature of clean images.

According to the obtained results, our approach reduces the detection rate of steganalyzers compared to the classical use of steganography methods. The advantage of our proposed approach is that in case of appearance of new steganalyzer methods, the fuzzy rule base can upgraded and the signature of clean images can become more trustable.

References

Sajedi H., Jamzad M.: Evolutionary rule generation for signature-based...
