JOURNAL OF INFORMATION SECURITY AND APPLICATIONS 30 (2016) 3–14

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/jisa

Steganalysis based on steganography pattern discovery

Hedieh Sajedi *

Department of Computer Science, School of Mathematics, Statistics and Computer Science, College of Science, University of Tehran, Tehran, Iran

ARTICLE INFO

Article history:
Available online 19 April 2016

Keywords:
Steganography
Evolutionary fuzzy rules
Steganalysis
Blind steganalysis

ABSTRACT

The goal of steganalysis algorithms is detection of stego images from clean images. Each steganography method based on its embedding mechanism puts a special pattern on the stego images. Finding this pattern in the images leads us to employ a classifier to be constructed specially for detecting stego images which are the results of a special steganography algorithm. In this paper, to have high detection accuracy, we propose an approach for Steganography Pattern Discovery (SPD). Our proposed approach employs an evolutionary method to extract the signature of stego images against clean images via fuzzy if–then rules. Based on the discovered knowledge, suitable trained models for steganalysis can be employed and stego images will be detected with high accuracy. Using SPD, we can predict the type of steganography method from a stego image. Employing SPD can enhance the approaches, which assume that a special steganography method is used. The effect of SPD before applying steganalysis methods has been investigated by some steganography and steganalysis techniques and it has been validated using some image databases. The results indicate that the pattern of a steganography method is extracted well and the type of steganography method used to make a stego image can be predicted with high accuracy.

1. Introduction

Steganography is the science of imperceptible communications. In cryptography, the attacker is able to identify, catch, and change the transmitted information (Kahn, 1996), nevertheless, steganography is used when we need to hide the existence of communicating. Steganography methods embed secret messages within visually innocent covers. Typical medias that can cover secret messages are image, video, and audio files (Munuera, 2007).

Invisible ink, covert channel, microdot, and spread-spectrum communication are some famous and ancient steganographic methods (Kahn, 1996; Norman, 1973). A famous classic steganographic model is the prisoners' problem. In this problem, Alice and Bob are in a jail and they plan to escape together (Simmons, 1984). The communications between them are monitored by Wendy, who is a warden. In this regard, they must hide the secret messages in another innocuous-looking means (cover object) to achieve the stego object. Afterward, the stego object is sent through the public channel. For more explanation about applications of steganography method, refer to Simmons (1984) and Westfeld and Pfitzmann (1999).

The fundamental requirement of steganographic systems 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 (Wu and Shih, 2006).

Various image steganography methods have been proposed in the literature. Due to the great use of JPEG images,

* Department of Computer Science, School of Mathematics, Statistics and Computer Science, College of Science, University of Tehran, Tehran, Iran. Tel.: 009821 61112915.
E-mail address: hhsajedi@ut.ac.ir.
http://dx.doi.org/10.1016/j.jisa.2016.04.001
2214-2126/

embedding in Discrete Cosine Transform (DCT) domain is well known. Steganography methods like F5 (Westfeld, 2001), Model-based (MB) (Sallee, 2003), Perturbed Quantization (PQ) (Fridrich et al., 2004), and YASS (Solanki et al., 2007) embed secret messages in images by modifications of carefully chosen DCT coefficients. In addition, some methods have been proposed which embed messages in other transform domains, such as Contourlet transform (Sajedi and Jamzad, 2008). The method presented in Sajedi and Jamzad (2008) embeds secret messages in Contourlet coefficients of a cover image.

Adaptive steganography schemes like WOW (Holub and Fridrich, 2012) started with the advancement of coding schemes (Fridrich et al., 2014) capable of embedding messages while nearly optimally minimizing arbitrarily defined additive distortion functions. Since the capacity of steganography methods is limited based on the properties of images, and the goal of the steganography methods is to be undetectable, the research in Sajedi and Jamzad (2009a, 2010a, 2010c, 2010d) is about increasing the embedding capacity of steganography methods to provide the capability of embedding larger secret messages.

In Qin et al. (2013), a prediction-based reversible steganographic scheme based on image inpainting is proposed. Another reversible data-hiding scheme in encrypted image is proposed in Qin and Zhang (2015). This scheme has better decrypted image quality and higher image recovery accuracy. In Zhang and Wang (2006), a method of steganographic embedding in digital images is proposed, in which each secret digit in a $(2n+1)$-ary notational system is carried by n cover pixels and, at most, only one pixel is increased or decreased by 1. In other words, the $(2n+1)$ different ways of modification to the cover pixels correspond to $(2n+1)$ possible values of a secret digit. Because the directions of' modification are fully exploited, this method provides high embedding efficiency.

In Qin et al. (2014), a joint data-hiding and compression scheme is presented for digital images using side match vector quantization (SMVQ) and image inpainting. The two functions of data hiding and image compression can be integrated into one single module seamlessly. On the sender side, except for the blocks in the leftmost and topmost of the image, each of the other residual blocks in raster-scanning order can be embedded with secret data and compressed simultaneously by SMVQ or image inpainting adaptively according to the current embedding bit.

In Qin et al. (2015), a data-hiding scheme with reversibility based on exploiting modification direction (EMD) is proposed. One cover image is first chosen and prepared to generate two visually similar steganographic images. During the secret embedding, the pixels in the first steganographic image are modified by no more than one gray level to embed secret data using the traditional EMD method, while the pixels in the second steganographic image are adaptively modified through referring to the first steganographic image without any confusions in image recovery process. On the receiver side, secret data can be extracted easily and the original cover image can also be recovered from the two steganographic images correctly.

In Sajedi and Jamzad (2009b) an approach for selecting proper cover images in steganography is presented. This 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. In the second stage, the fuzzy rules are used for selecting suitable cover images for steganography. This approach selects the appropriate cover images from an image database and using them produces more secure steganography.

Due to the various contents of images, the stego images produced by a steganography method may have different levels of undetectability against steganalyzers. In other words, a steganography method may cause less detectable statistical artifacts on some images compared to other images. In Sajedi and Jamzad (2010e), different features of images are analyzed to find the similarity between proper cover images for each steganography method. Similarity between images is modeled in the form of fuzzy rules. Subsequently for hiding secret data in a cover image, a reliable steganography method is suggested in Sajedi and Jamzad (2010e) that results to an undetectable stego image against steganalysis methods.

In the current paper, the idea of extracting fuzzy rules in a similar way of Sajedi and Jamzad (2009b, 2010e) is used to reveal the signature of steganography methods and enhance the performance of steganalysis algorithms.

Steganalysis algorithms try to distinguish stego images from clean images. Generally, a classifier is built based on stego and clean image. In condition of observing a new image, we do not have any information about the used steganography method or the payload. Therefore, a general steganalyzer is built using a set of clean images and a set of stego images generated by various steganography algorithms and different payloads.

On the other words in popular steganalysis methods, the important issue is to detect the existence of the hidden information. They do not consider different patterns of steganography algorithms. Therefore, the classifier should learn a complex function that can distinguish clean images from stego images with various steganography patterns.

Each steganography method employs a special mechanism to embed secret data in the images. Therefore, it puts a distinct pattern on the stego images. Discovering this pattern in the images leads us to hire a proper classifier to be constructed particularly to detect stego images which are the results of a special steganography algorithm.

In this paper, we present an approach that consists of two stages. In the first stage, we analyze an image database to discover the pattern or signature of stego images. By the pattern, we mean the effective features of stego images and their relative values. This pattern is constructed in the form of a set of fuzzy if–then rules that represent the similarity between stego images. In the second stage, the steganalyzer is trained to detect only one steganography method at once. After discovering patterns of the used steganography method from the stego image, the proper model is used to analyze it. This approach simplifies the problem of blind steganalysis to partially blind steganalysis.

The process of generating the signature of stego images is done by an Evolutionary Algorithm (EA). EAs have been used as rule generation and optimization tools in the design of fuzzy rule-based systems (Cordon et al., 2004; Hu et al., 2003).

To obtain accurate fuzzy rules, we employ an evolutionary rule generation algorithm based on Iterative Rule Learning (IRL) approach (Sajedi and Jamzad, 2009b, 2010e). The rules are generated incrementally so that the evolutionary algorithm optimizes one fuzzy rule at a time. The generated fuzzy rules are

then used as the signature of stego images or, in other words, pattern of the employed steganography method. We applied our approach to MB, PQ, and YASS steganography techniques and validated it using an image database. Experimental results indicate that evaluating images based on the discovered patterns of steganography methods increases the detection accuracy of steganalysis methods considerably.

The goal of this work is extraction rules that show the pattern of steganography methods and applying steganalysis method based on these rules.

The paper is organized as follows: Related works are introduced in Section 2. Steganography pattern discovery is presented in Section 3. Experiments are reported in Section 4 and Section 5 concludes this paper.

2. Related works

Most of the existing works in steganalysis assumed that the employed steganography method is not known like Fridrich and Holub (2015); Fridrich et al. (2013); Li et al. (2008); Sajedi and Jamzad (2010b); Solanki et al. (2007); and Westfeld (2001). For example, the innovation of the research (Fridrich and Holub, 2014) is proposing low complexity features for JPEG steganalysis using undecimated DCT. The features are engineered as first-order statistics of quantized noise residuals obtained from the decompressed JPEG image using 64 kernels of the discrete cosine transform (the so-called undecimated DCT). Considering the great advances in steganalysis of grayscale images, it is rather surprising that steganalysis that uses the more complex structure of color images has largely been neglected by the research community. In this regard, the research in Fridrich and Goljan (2015) introduces features for steganalysis of color images. The research in Sajedi and Jamzad (2010b) proposed extracting features in Contourlet transform to increase the accuracy of steganalyzers.

In this paper, firstly, the type of steganography method is predicted and then a suitable model is employed for analyzing the images.

2.1. *Image feature selection*

To detect the existence of a hidden message in an image various steganalysis methods consider different features of images.

Commonly, feature selection can be divided into two groups: filtering and wrapper methods. Filtering methods select feature subsets independently from the learning classifiers and do not include learning (Hofmann, 2004; Ishibuchi et al., 2001). The flaw of filtering methods is that they only consider the single feature in isolation and disregard the possible interaction of features among them. However, the combination of these features may have a combined effect that does not necessarily follow from the separate performance of features in the group. However, if there is a limit on the number of features to be selected, no informative features may be included. The wrapper methods wrap around a certain learning algorithm that can assess the selected feature subsets in terms of estimated classification errors and then build the final classifiers (Inza et al., 2002).

In Liu and Sung (2007) and Liu et al. (2008), schemes based on feature mining and pattern classification are presented to detect Least Significant Bit (LSB) matching steganography in grayscale images.

In Ghareh Mohammadi and Saniee Abadeh (2012), several data mining approaches on steganalysis of images, audio, video, text and protocol are studied. The main aim of this study is to present the efficiency of using data mining techniques in steganalysis in comparison to the model based steganalysis approaches.

Using a large number of features is unattractive in terms of classification performance due to the curse of dimension (Wang and Moulin, 2007). Additionally, performing feature selection in steganalysis offers some advantages (Miche et al., 2006) as follows:

- Pruning the meaningless features
- Improvement of classification performance
- Reducing the complexity for both feature generating and classifier training
- Help 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 employ a learning approach in Section 3 to find accurate and interpretable fuzzy if–then rules.

2.2. *Fuzzy rule generation*

Employing EAs to construct fuzzy rule-based systems is referred to as Evolutionary Fuzzy Systems (EFS), each of which can be classified into Michigan, Pittsburgh, and Iterative Rule Learning approaches (Cordon et al., 2004). In Michigan approach, a single fuzzy if–then rule is considered as an individual (Ishibuchi and Nakashima, 1999; Ishibuchi et al., 2005). In Pittsburgh approach, a set of fuzzy if–then rules is considered as an individual (Ishibuchi et al., 2001; Rouwhorst and Engelbrecht, 2000). In Iterative Rule Learning approach, each individual codes one rule and in each iteration of Genetic Algorithm (GA) a new rule is adapted and added to the rule set, iteratively (Hofmann, 2004; Ozyer et al., 2007).

The most important characteristic of EFS that inspires us to use them as a rule generation tool is their notable capability to produce precise and interpretable knowledge (Cordon et al., 2004). 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, which is a 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 stego images and, finally, pattern of steganography method. We will discuss about the details of our approach in the following section.

3. Steganography pattern discovery

In this paper, we utilize an iterative evolutionary fuzzy algorithm for Steganography Pattern Discovery (SPD). In this utilization, the proposed algorithm extracts a fuzzy rule base for steganalysis problem. Introducing the pattern of each steganography method, our proposed approach classifies clean and stego images with high accuracy and introduces the used steganography method based on fuzzy rules.

Fig. 1 shows the block diagram of steganography pattern discovery. The following subsections describe the details of this approach. In SPD, we have S_i, $i = 1,\ldots,I$ steganography methods, the amount of embedded secret data(payload) can be L_j, $j = 1,\ldots,J$ and we have A_k, $k = 1,\ldots,K$ steganalysis method. Accordingly, the number of steganographic patterns P_{ijk} is $I \times J \times K$ and we require to construct $I \times J \times K$ models M_{ijk} for investigating cleanness of an image. Therefore, instead of showing a test image to all of the models, we can match its feature vector with all the patterns and after finding the most possible one, the corresponding model is employed for the final evaluation.

3.1. *Feature vector generation*

This section deals with the generation of feature vector from the image database. A 636-dimensional feature vector is produced by appending the features of four efficient and famous steganalyzers. Table 1 shows that the features are computed according to the proposed features of Pevný and Fridrich (2007), Chen et al. (2006), Lyu and Farid (2002), and YASS steganalysis methods. In the following, the features are briefly reviewed, which are used by these steganalyzers.

1. Pevný et al. (Pevný and Fridrich, 2007) merge 193 extended DCT features with 81 averaged calibrated Markov features to provide a 274-dimensional feature vector. In this method, Markov features model intra block DCT dependencies and DCT features (first 193 features) model inter block relations. However, many of the 274 features may be highly correlated to each other. In this paper, we called this method as 274-dim steganalyzer.
2. In Chen et al. (2006), 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. Lyu and Farid proposed Wavelet-based steganalysis method in Lyu and Farid (2002). This steganalyzer builds a model for clean images by using higher order statistics, and then considers deviation of stego images from the built 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 computed from wavelet coefficients of each high-

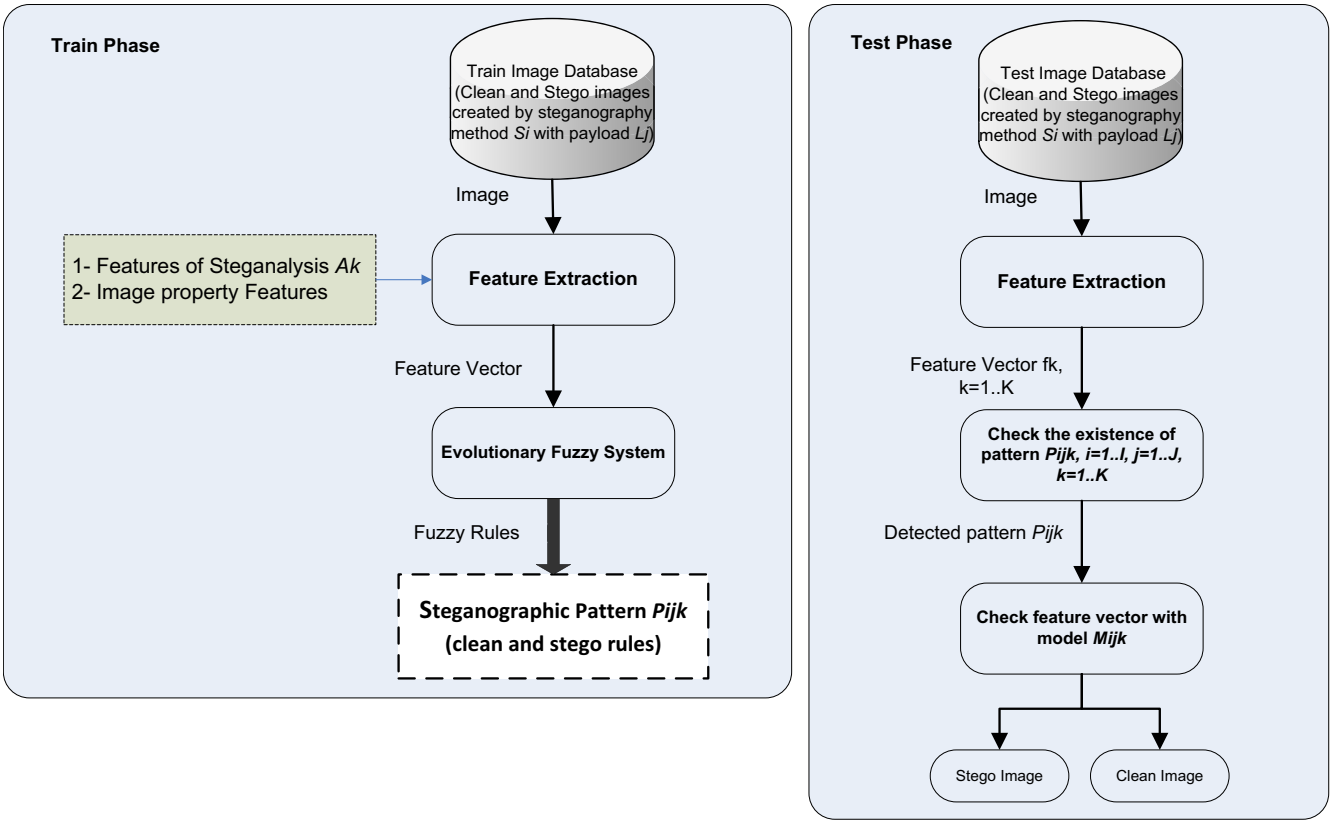

Fig. 1 – The block diagram of steganography pattern discovery.

Table 1 – Four groups of 636 image features and types of 636 image features.

Steganalysis method	Feature group (number of features)	Number of features	Feature type
Pevný–Fridrich (Pevný and Fridrich, 2007)	274	11	Global histogram
		66	5 AC histograms
		99	11 dual histograms
		1	Variation
		2	Blockiness
		25	Co-occurrence matrix
		81	Markov features
Chen (Chen et al., 2006)	324	39	Histogram of spatial representation and discrete wavelet transform (DWT) representation
		39	Histogram of prediction error and DWT of error
		39	Histogram of JPEG representation and its DWT
		78	Horizontal 2-D histogram of JPEG representation and its DWT
		78	Vertical 2-D histogram of JPEG representation and its DWT
		78	Diagonal 2-D histogram of JPEG representation and its DWT
		39	Histogram obtained from prediction error of JPEG representation and its DWT
Lyu–Farid (Lyu and Farid, 2002)	24	24	Higher order statistics of each wavelet subband
YASS (Solanki et al., 2007)	14	14	Frequency of zeros

frequency subband to form one set of features. Another set of features is in the same way formulated from the prediction errors of wavelet coefficients of each high-frequency subband. We referred to this method as 24-dim steganalyzer in the rest of the paper.

4. In Solanki et al. (2007), a steganalysis method is proposed that employs a 14-dimensional feature vector for analysis. It is based on the average frequency of zeros in possible locations, average frequency of zeros in impossible locations and their differences. In the following, this method is referred to as 14-dim steganalyzer.

The types of all 636 features are given in Table 1. 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 the unit interval [0,1] in order to use the same membership function for them in the fuzzy rule generation. In the next subsection, this step is explained in detail.

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 the previous subsection. Each fuzzy if–then rule 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 similar to Sajedi and Jamzad (2009b) are as follows:

Rule R_j: *If* (x_1 is A_{j1} and ... 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. The rectangular shape of DC membership function means if a feature gets the value (*DC*), the value of membership function is 1. CF_j is the certainty factor of the fuzzy if–then rule R_j. Each fuzzy rule has a certainty factor 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 the 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., steganalysis based on $n = 636$ features). Therefore, our evolutionary method searches for a relatively small number of fuzzy 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 of stego images with high accuracy. This pattern is

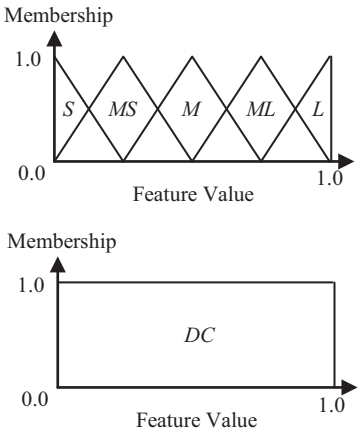

Fig. 2 – Membership Functions. *S*: small; *MS*: medium, small; *M*: medium; *ML*: medium, large; *L*: large; and *DC*: don't care.

extracted according to the training samples of clean and stego images.

The following three steps are applied to calculate the certainty factor 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 Eq. (1):

$$\mu_j(x_p) = \mu_{j1}(x_{p1}) \times \ldots \times \mu_{jn}(x_{pn}), \quad P = 1, 2, \ldots, M. \tag{1}$$

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 rule R_j:

$$\beta_{Clean}(R_j) = \frac{\sum_{x_p \in Clean} \mu_j(x_p)}{N_{Clean}} \tag{2}$$

$$\beta_{Stego}(R_j) = \frac{\sum_{x_p \in Stego} \mu_j(x_p)}{N_{Stego}} \tag{3}$$

where $\beta_{Clean}(R_j)$ and $\beta_{Stego}(R_j)$ are the relative sums of the compatibility grades of training samples that represent clean and stego images, respectively. Note that N_{Clean} and N_{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 for clean images is determined as Eq. (4):

$$CF_j = \frac{(\beta_{Clean}(R_j) - \beta_{Stego}(R_j))}{(\beta_{Clean}(R_j) + \beta_{Stego}(R_j))} \tag{4}$$

By the proposed heuristic procedure, we can specify the certainty factor 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 the evolutionary fuzzy algorithm in detail.

3.3. *Evolutionary fuzzy algorithm*

Our employed evolutionary fuzzy algorithm learns rules iteratively by optimizing one fuzzy rule in each iteration of the algorithm. Firstly, all the training samples have 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 the 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 current training samples, which are uncovered by the rules achieved in the 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.

In our learning system, we have used a fitness function in the evolutionary process, which is computed for example for discovering pattern of clean images according to Eqs. (5)–(7).

$$f_P = \frac{\sum_{x^k \in Clean} w^k \mu_{R_i}(x^k)}{\sum_{x^k \in Clean} w^k} \tag{5}$$

$$f_N = \frac{\sum_{x^k \in Stego} w^k \mu_{R_i}(x^k)}{\sum_{x^k \in Stego} w^k} \tag{6}$$

$$fitness(R_j) = w_P f_P - w_N f_N \tag{7}$$

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 and w_N is the weight of rule's negative power.

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

1. *Initialization*: Produce an initial population of fuzzy if–then rules based on the weight of training samples.
2. *Generation*: Generate new fuzzy if–then rules by genetic operations.
3. *Replacement*: Replace a part of the current population with the newly generated rules.
4. *Inner Cycle Termination Test*: Terminate the inner cycle (Step 2 and Step 3) of the algorithm if a stopping condition is satisfied, otherwise go to Step 2.
5. *Outer Cycle Termination Test*: Terminate the outer cycle (Step 1 to Step 6) if a stopping condition is satisfied, otherwise go to Step 6.
6. *Weight Adjustment*: Reduce the weight of training samples that cover the new obtained fuzzy rule. Go to Step 1.

The output of each outer cycle of the above algorithm is one fuzzy if–then rule. Each step of the presented learning algorithm is described as follows:

Step 1: N_{pop} denotes the number of rules in the population of genetic algorithm. To produce an initial population, N_{pop} rules are generated according to the features of random samples in the training database. The probability of each training sample to be chosen in this step is relative to its current weight. In this regard, 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 a random sample is measured by Eq. (1). The certainty factor of each fuzzy if–then rule is determined according to the heuristic method, explained in the 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 Eq. (7).

Step 2: A pair of rules is nominated from the current population to generate new rules for the following population. Each rule in the current population is selected using the tournament selection mechanism. This process is repeated until a certain number of pairs of rules are selected. Crossover operation is then applied to a selected random pair of rules with a certain crossover probability. We have used uniform crossover in our computer simulations. With a certain mutation probability, each antecedent of rules is randomly replaced with a different antecedent fuzzy set after the crossover operation. The probability of changing to don't care value, P_{DC}, is more than the other five linguistic values. After performing selection, crossover, and mutation operators, the fitness value of each of the generated rules is evaluated according to Eq. (7).

Step 3: A pre-specified 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 Eq. (7).

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

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 the 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 introduced into the primary 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. This adjustment prevents the relearning of correctly classified instances and therefore provides the opportunity for misclassified instances of the previous iteration to be learned in the new iteration. Note that initially $w^k = 1$. After this step, the algorithm jumps to Step 1.

3.4. *Steganography pattern discovery*

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 Fig. 1 shows, each pair of stego image database and clean image database is fed to the evolutionary fuzzy rule generation stage (we set the parameters of MB, PQ, and YASS to construct stego image databases with a variety of payloads). After this stage, an image rule set has 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. All the rules of images are put in an image rule base.

For a given rule base S, in order to determine whether or not an image with feature vector $x_p = (x_{p1}, x_{p2}, \ldots, x_{pn})$ is stego using the steganography method with index i, the parameter τ^i_{Stego} is computed using Eq. (8).

$$\tau_{Stego} = \sum_{R_j \in S} \mu_j(x_{ps}) CF_j \tag{8}$$

Discovered Pattern (*DP*), the employed steganography method/cleanness of image *I*, is determined based on Eq. (9):

$$DP(I) = argmax\left(\left\{\tau^n_{Stego}, \tau_{Clean}\right\}\right) \quad for \quad n = 1 \text{ to } N \tag{9}$$

SPD in the current paper is a 4-class (MB, PQ, YASS and Clean) classification problem. Therefore, if another steganography method will be considered in the future, the problem is changed to a 5-class classification problem but the number features for extracting SPD are fixed.

4. Experiments

In this section, experimental results of the proposed method are presented.

The experiments were executed on a personal computer with a 4 GB RAM, Intel(R) Pentium (R), 3.00 GHz processor, and Matlab R2010b was used for program writing. In the following, the way of providing the code to make stego datasets and evaluating the results is explained.

The database of 4959 images was prepared for evaluation. For performance evaluation of the proposed approach different experiments were done. We obtained 1000 JPEG images from Washington University image database (Li, 2005) and 3959 images were taken with six cameras with different resolutions. 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. Accordingly, each classifier is built by using 1000 stego and 1000 clean images. Table 2 shows the source of codes to produce such data sets.

In this paper, 4 different payloads, and 3 steganography and 4 steganalysis methods are considered. Experiments can be done in 4 situations:

- Situation 1 (Blind Steganalysis): For traditional evaluation without considering SPD, 4 models should be built based on 4 steganalysis methods. In this situation, type of the used

Table 2 – Provided steganography and steganalysis codes.

Algorithm		Available from
Steganography method	MB	http://www.codeforge.com/article/157206#introduction
	PQ	http://dde.binghamton.edu/download/pq/
	YASS	Get privately from the authors of Solanki et al. (2007)
Steganalysis method	274-dim steganalyzer	http://dde.binghamton.edu/download/ccmerged/
	324-dim steganalyzer	http://dde.binghamton.edu/download/feature_extractors/
	24-dim steganalyzer	http://www.cs.dartmouth.edu/farid/#jumpTo
	14-dim steganalyzer	Implemented in Matlab

steganography method and the amount of the payload are unknown.

- Situation 2 (Partially Blind Steganalysis): If we assume that the employed steganography method is known, then we can build 12 models based on 4 steganalysis and 3 steganography methods while the amount of the payload is unknown. In this situation, the achieved accuracy for steganalysis is usually higher than Situation 1 because the models are built more specifically than Situation 1. In the real world, we are not aware of the employed steganography method.
- Situation 3 (Unblind): If we assume that the employed steganography method and the payload are known, then we can build 48 models based on 4 steganalysis and 3 steganography methods and 4 different amounts for the payloads. This situation is more specific than situation 2 and the models are more accurate. In the real world, we are not aware of the employed steganography method and the amount of the payload. Usually in the papers, steganalyzers are built specially for a certain size (range) of payload for a particular steganography method.
- Situation 4 (Blind in nature but Operate Partially Blind): If we use SPD, which tries to guess the employed steganography method, we can build 12 models based on 4 steganalysis methods and 3 steganography methods. In the real world, we do not know the employed steganography method and the amount of the payload but we can use SPD to guess the employed steganography method. When a new image is given for analysis, after analyzing the results of SPD we deliver the new observed image to the true classifier for further analysis.

The experiments were run in 10-fold cross validation way. We quantified the steganalysis performance according to the detection accuracy (Solanki et al., 2007). 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:

Table 3 – Value of parameters in simulations.

Parameter	Value
Population size (N_{pop})	500
Don't care replacement rate	0.5
Crossover probability	0.9
Mutation probability	0.5
Fitness positive weight	0.1
Fitness negative weight	0.9
Replacement percentage	10
Maximum number of the iteration	50

$$D_a = P_d \times 100$$

$$P_d = 1 - P_{error}$$

$$\begin{aligned} P_{error} &= P(X_0)P(Y_1|X_0) + P(X_1)P(Y_0|X_1) \\ &= 1/2\,P_{FA} + 1/2\,P_{miss}; \quad for \quad P(X_0) = P(X_1) = 1/2 \end{aligned} \tag{10}$$

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 toward 100.

We normalized the features extracted from the databases, where each numerical value in the data set is normalized between 0 and 1 according to Eq. (11):

$$x_{normalized} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{11}$$

Hence, 636 numeric features are constructed and normalized to the interval [0, 1].

Table 3 shows the parameter specification that we have used in our computer simulations for the evolutionary fuzzy rule generation algorithm.

4.1. *Relation between detection rate of steganalysis and embedding rate of steganography*

This experiment is done in Situation 3. To investigate the relation between the embedding rate and the detection rates of steganalyzers, we adjust the embedding rate of steganography methods to different values. Then we measure the average detection rates of four used steganalyzers on YASS, PQ, and MB steganography methods. In this experiment, SPD is not employed . The results in Fig. 3 show that in most of the cases the detection accuracy has a direct relation with embedding rate. Nevertheless, using YASS as the steganalysis algorithm, this relation is not direct in all the cases. In this experiment, we report the results of 48 classifiers. Each classifier has been trained using 1000 clean and 1000 stego images.

4.2. *Detection rate of steganalysis without SPD*

This experiment is done in Situation 2. In this experiment, we report the results of 12 classifiers. Each classifier has been

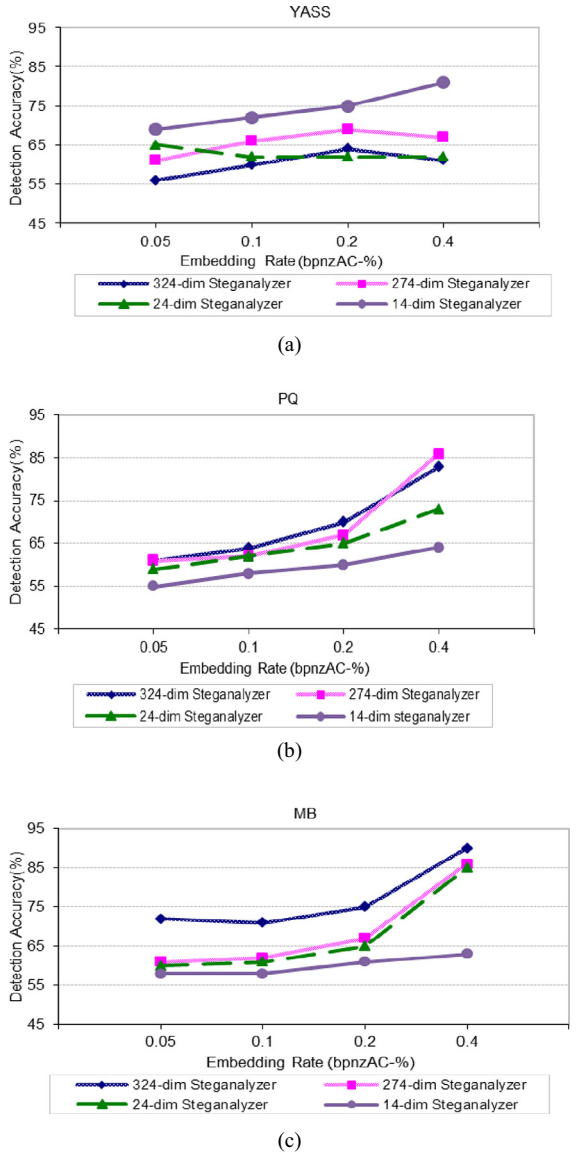

Fig. 3 – The relation between detection rate and embedding rate: (a) YASS, (b) PQ, and (c) MB steganography methods.

trained using 4000 clean and 4000 stego images. Stego images have different payloads. The results are shown in Table 4.

4.3. *Evaluation of the proposed method*

This experiment is done in Situation 4. In this experiment, we evaluate the effect of SPD. In the first part we investigate the accuracy of each SPD. In this regard, we measure the accuracy of each model in recognizing the patterns. The results of three models are shown in Table 5. In this experiment the fuzzy rules base is used to predict the hype of employed steganography method. Actually SPD performs as a blind steganalysis.

Table 4 – Average accuracy of steganalysis methods in detection of stego images with different payloads, produced by different steganography methods (in percent, %).

	Steganography method	MB	PQ	YASS
Steganalysis method (general classifier)	274-dim	77	69.5	60
	324-dim	78	69	66
	24-dim	68	65	63
	14-dim	60	59	74
Average		71	66	66

Table 5 – Average accuracy of steganography patterns in detection of employed steganography methods (in percent, %).

Real images	SPD estimation		
	MB	PQ	YASS
MB	91	—	—
PQ	—	81	—
YASS	—	—	88
Clean	90	79	89

Table 6 shows the detection accuracy of four steganalysis methods on MB, PQ, and YASS steganography methods after applying SPD. In this application SPD helps simplify a blind steganalysis problem to a partially blind steganalysis problem. In this experiment after SPD, 12 classifiers are used to report the detection rates. As the results demonstrate, steganography pattern discovery approach on average decreases the undetectability of steganography methods compared to Table 4, which shows the results of traditional steganalysis. Consequently, we can comprehend that our proposed approach enhances the performance of steganalysis methods considerably.

In Table 6, the accuracy is 55% for MB by 14-dim, and 58% for YASS by 324-dim, which are lower than the previous results in Table 4. This difference is not important because when the pattern of steganography is discovered, the best steganalyzer can be used. For example, for MB pattern, the 324-dim classifier is the best choice. In other words, SPD is a coarse scale analysis and based on the output of SPD, the fine scale analysis is done by the proper steganalysis.

As a result, we can see that if we use only SPD, since it uses features from four strong steganalysis methods and in detection of each steganography method, some special features are discriminative, and we can use only SPD as a steganalysis.

As we can conclude from the process of the proposed method, the proper steganalysis method is recognized after SPD execution. This process is done during the modeling. For testing, we know the proper steganalyzer for testing an input image. Based on the results in Table 6, it is obvious that 324-dim steganalyzer is the proper method if the used steganography method is MB, 274-dim steganalyzer is the best for detecting PQ, and 14-dim steganalysis method is the best one for detecting YASS steganography.

4.4. *Evaluation of the proposed method as a blind steganalysis method*

This experiment is done in Situation 4. In this experiment, a data set that consists of clean and stego images with various payloads and produced by different steganography methods is used. As Table 7 shows, the accuracy of SPD in predicting the type of the used steganography method is less than 20 percent. By having the results of SPD, the accuracy of

Table 6 – Average accuracy of steganalysis methods in detection of stego images with different payloads, produced by different steganography methods (in percent, %).

	Steganography method					
	MB (with SPD)		PQ (with SPD)		YASS (with SPD)	
	Accuracy (%)		Accuracy (%)		Accuracy (%)	
Steganalysis method (steganography pattern discovery)	274-dim	82	274-dim	80	274-dim	70
	324-dim	91	324-dim	77	324-dim	58
	24-dim	79	24-dim	68	24-dim	66
	14-dim	55	14-dim	60	14-dim	92
Average		77		71		72

Table 7 – Average accuracy of steganalysis methods with employing SPD, in detection of stego images (in percent,

Steganography method										
MB			PQ			YASS			MB + PQ + YASS	
Proper steganalysis	Accuracy of SPD (%)		Proper steganalysis	Accuracy of SPD (%)		Proper steganalysis	Accuracy of SPD (%)		Accuracy of SPD (%)	
324-dim	90		274-dim	80		14-dim	92		87	
	TP	FP	Accuracy	TP	FP	Accuracy	TP	FP	Accuracy	
Average final steganalysis accuracy (%)	88	20	84	61	35	62	87	31	78	74

Table 8 – Average execution time of SPD, steganalyzers and SPD application.

Model		Time (secs)
SPD extraction		311
Steganalysis method	274-dim	7
	324-dim	6
	24-dim	2
	14-dim	3.2
SPD application (without time of steganalysis)		2

steganalysis methods is shown in the last row of the table. The final accuracy is about 74%, while without using SPD this value is less than 70% (from Table 4).

4.5. *Computational complexity*

Due to the considerable computation of the evolutionary method, the execution time or computational complexity of the proposed scheme is explained in the following. Extracting fuzzy rules via an evolutionary algorithm has a high complexity. But in steganalysis as a process of checking an image to find either it has secret data or not, only prepared fuzzy rules are checked. Therefore being time consuming of rule extraction can be tolerated. The average execution time of Table 6 is shown in Table 8.

5. Conclusions

Generally, this research proposes an approach for steganography pattern discovery which shows an approach of evolutionary method, which extracts the signature of stego images against clean images via fuzzy if–then rules. Basically, the proposed approach consists of two stages: firstly, an analysis of image database to discover the pattern or signature of stego images, and secondly, the steganalyzer trained to detect only one steganography method at once. In other words, in this paper, to increase the accuracy of steganalysis images, we proposed a novel step before steganalysis process. After discovering the used steganography method, a suitable model for steganalysis is employed. In this regard, an evolutionary fuzzy algorithm is proposed to generate fuzzy rules from features of the stego images. These rules are used to form the pattern of steganography methods.

According to the obtained results, our approach increases the detection rate of steganalyzers compared to the classical use of steganalysis methods. The advantage of our proposed approach is that in appearance of new steganography method, the fuzzy rule base can be upgraded and the pattern of this method can be discovered for using by the steganalysis.

Conflict of interests

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgements

This research has been supported by a Grant from Iran National Science Foundation (INSF) and the grant number is 87041894.

REFERENCES

Chen C, Shi YQ, Chen W, Xuan G. Statistical moments based universal steganalysis using JPEG-2D array and 2-D characteristic function. In: Proceedings of ICIP image processing. Atlanta, GA, USA; 2006. p. 105–8.

Cordon O, Gomide F, Herrera F, Hofmann F, Magdalena L. Ten years of genetic fuzzy systems current framework and new trends. Fuzzy Sets Syst 2004;41(1):5–31.

Fridrich J, Goljan M. CFA-aware features for steganalysis of color images. In: Proceedings of SPIE, electronic imaging, media watermarking, security, and forensics. San Francisco; 2015.

Fridrich J, Holub V. Low complexity features for JPEG steganalysis using undecimated DCT. IEEE Trans Inf Forensics Secur 2014;10(2).

Fridrich J, Holub V. Phase-aware projection model for steganalysis of JPEG images. In: Proceedings of SPIE, electronic imaging, media watermarking, security, and forensics. San Francisco; 2015.

Fridrich J, Goljan M, Soukal D. Perturbed quantization steganography with wet paper codes. In: Proceedings of ACM, multimedia workshop. Germany; 2004.

Fridrich J, Ker AD, Bas P, Boehme R, Cogranne R, Craver S, et al. Moving steganography and steganalysis from the laboratory into the real world. In: ACM workshop on information hiding and multimedia security. Montpellier, France; 2013.

Fridrich J, Denemark T, Sedighi V, Holub V, Cogranne R. Selection-channel-aware rich model for steganalysis of digital images. In: Proceedings of IEEE workshop on information forensics and security. Atlanta, GA; 3–5 December 2014.

Ghareh Mohammadi F, Saniee Abadeh M. A survey of data mining techniques for steganalysis. In: Sajedi H, editor. Recent advances in steganography. InTech Publisher; 2012. ISBN 978-953-51-0840-5.

Hofmann F. Combining boosting and evolutionary algorithms for learning of fuzzy classification rules. Fuzzy Sets Syst 2004;141(1):47–58.

Holub V, Fridrich J. Designing steganographic distortion using directional filters. In: Proceedings of 4th IEEE international workshop on information forensics and security. Tenerife, Spain; 2–5 December 2012.

Hu Y, Chen R, Tzeng G. Finding fuzzy classification rules using data mining techniques. Pattern Recognit Lett 2003;24:509–19.

Inza I, Sierra B, Blanco R, Larranaga P. Gene selection by sequential search wrapper approaches in microarray cancer class prediction. J Intell Fuzzy Syst 2002;12(1):25–33.

Ishibuchi H, Nakashima T. Improving the performance of fuzzy classifier systems for pattern classification problems with continuous attributes. IEEE Trans Ind Electron 1999;46(6).

Ishibuchi H, Nakashima T, Murata T. Three-objective genetics-based machine learning for linguistic rule extraction. Inf Sci (Ny) 2001;136:109–33.

Ishibuchi H, Yamamoto T, Nakashima T. Hybridization of fuzzy GBML approaches for pattern classification problems. IEEE Trans Syst Man Cybern B Cybern 2005;35(2):359–65.

Kahn D. The codebreakers. 2nd ed. New York: Macmillan; 1996.

Kahn M. The history of steganography. In: Information hiding, vol. 1174. Lecture Notes in Computer Science. Berlin: Springer; 1996. p. 183–206.

Li B, Shi Y, Huang J. Steganalysis of YASS. In: ACM workshop on information hiding and multimedia security. 2008. p. 139–48.

Li Y. Object and concept recognition for content-based image retrieval, Doctoral Dissertation, University of Washington, 2005. <http://www.cs.washington.edu/research/imagedatabase> [accessed 05.27.15].

Liu Q, Sung AH. Feature mining and neuro-fuzzy inference system for steganalysis of LSB matching steganography in grayscale images. In: Proceedings of international joint conference on artificial intelligence. 2007.

Liu Q, Sung AH, Chen Z, Xu J. Feature mining and pattern classification for steganalysis of LSB matching steganography in grayscale images. Pattern Recognit 2008;41(1):56–66.

Lyu S, Farid H. Detecting hidden messages using higher-order statistics and support vector machines. In: Proceedings of 5th international workshop on information hiding. 2002.

Miche Y, Roue B, Lendasse A, Bas P. A feature selection methodology for steganalysis. In: Proceedings of international workshop on multimedia content representation, classification and security. Istanbul, Turkey; 2006. p. 49–56.

Munuera C. Steganography and error-correcting codes. Signal Processing 2007;87:1528–33.

Norman B. Secret warfare. Washington, DC: Acropolis Books; 1973.

Ozyer T, Alhajj R, Barker K. Intrusion detection by integrating boosting genetic fuzzy classifier and data mining criteria for rule pre-screening. J Netw Comput Appl 2007;30(1):99–113.

Pevný T, Fridrich J. Merging Markov and DCT features for multi-class JPEG steganalysis. In: Proceedings of SPIE, electronic imaging, security, steganography, and watermarking of multimedia contents. 2007.

Qin C, Zhang X. Effective reversible data hiding in encrypted image with privacy protection for image content. J Vis Commun Image Represent 2015;31(C):154–64.

Qin C, Chang C, Huang Y, Liao L. An inpainting-assisted reversible steganographic scheme using a histogram shifting mechanism. IEEE Trans Circuits Syst Video Technol 2013;23(7):1109–18.

Qin C, Chang C, Chiu Y. A novel joint data-hiding and compression scheme based on SMVQ and image inpainting. IEEE Trans Image Process 2014;23(3):969–78.

Qin C, Chang C, Hsu T. Reversible data hiding scheme based on exploiting modification direction with two steganographic images. Multimed Tools Appl 2015;74(15):5861–72.

Rouwhorst SE, Engelbrecht AP. Searching the forest: using decision trees as building blocks for evolutionary search in classification databases. In: Proceedings of IEEE congress on evolutionary computation, vol. 1. 2000. p. 633–8.

Sajedi H, Jamzad M. Adaptive steganography method based on contourlet transform. In: Proceedings of 9th international conference on signal processing. 2008. p. 745–8.

Sajedi H, Jamzad M. ContSteg: contourlet-based steganography method. Wirel Sens Netw 2009a;1(3):163–70.

Sajedi H, Jamzad M. Evolutionary rule generation for signature-based cover selection steganography. Neural Netw World 2009b;20(3):297–316.

Sajedi H, Jamzad M. BSS: boosted steganography scheme with cover image preprocessing. Expert Syst Appl 2010a;37:7703–10.

Sajedi H, Jamzad M. CBS: contourlet-based steganalysis method. J Signal Process Syst 2010b;61(3):367.

Sajedi H, Jamzad M. Contourlet-based steganography using cover selection. Int J Inf Secur 2010c;9(5):337–45.

Sajedi H, Jamzad M. HYSA: hybrid steganographic approach using multiple steganography methods. Secur Commun Netw 2010d;4:1173–84.

Sajedi H, Jamzad M. Selecting a reliable steganography method. In: 2010 international conference on multimedia computing and information technology (MCIT). 2010e. p. 69–72.

Sallee P. Model-based steganography. In: Proceedings of international workshop on digital watermarking. Seoul, Korea; 2003.

Simmons GJ. Prisoners' problem and the subliminal channel. In: Proceedings of international conference on advances in cryptology. 1984. p. 51–67.

Solanki K, Sarkar A, Manjunath BS. YASS: yet another steganographic scheme that resists blind steganalysis. In: Proceedings of 9th international workshop on information hiding. 2007.

Wang Y, Moulin P. Optimized feature extraction for learning-based image steganalysis. IEEE Trans Inf Forensics Secur 2007;2(1):31–45.

Westfeld A. F5—a steganographic algorithm: high capacity despite better steganalysis. In: Proceedings of the 4th international workshop on information hiding. 2001.

Westfeld A, Pfitzmann A. Attacks on steganographic systems breaking the steganographic utilities EzStego, Jsteg, Steganos, and S-tools and some lessons learned. In: Proceedings of the 3rd international workshop on information hiding. Dresden, Germany; 1999. p. 61–76.

Wu YT, Shih FY. Genetic algorithm based methodology for breaking the steganalytic systems. IEEE Trans Syst Man Cybern B Cybern 2006;36(1).

Zhang X, Wang S. Efficient steganographic embedding by exploiting modification direction. IEEE Commun Lett 2006;10(11):781–3.