

Stego anomaly detection in images exploiting the curvelet higher order statistics using evolutionary support vector machine

S. Muthuramalingam¹ · N Karthikeyan² · S. Geetha³ ·
Siva S. Sivatha Sindhu⁴

Received: 11 February 2015 / Revised: 9 September 2015 / Accepted: 5 October 2015 /
Published online: 13 November 2015
© Springer Science+Business Media New York 2015

Abstract Steganalysis is an important extension to existing security infrastructure, and is gaining more research focus of forensic investigators and information security researchers. This paper reports the design principles and evaluation results of a new experimental blind image steganalysing system. This work approaches the steganalysis task as a pattern classification problem. The detection accuracy of the steganalyser depends on the selection of low-dimensional informative features. We investigate this problem as a three step process and propose a novel steganalyser with the following implications: a) Selection of the Curvelet sub-band image representation that offers better discrimination ability for detecting stego anomalies in images, than other conventional wavelet transforms. b) Exploiting the empirical moments of the transformation as effective steganalytic features c) Realizing the system using an efficient classifier, evolutionary-Support Vector Machine (SVM) model that provides promising classification rate. An extensive empirical evaluation on a database containing 5600 clean and stego images shows that the proposed scheme is a state-of-the-art steganalyser that outperforms other previous steganalytic methods.

✉ S. Geetha
geethabaalan@gmail.com

S. Muthuramalingam
smrit@tce.edu

N Karthikeyan
nkarthikkeyan@gmail.com

Siva S. Sivatha Sindhu
sivathasindhu@gmail.com

¹ Department of Information Technology, Thiagarajar College of Engineering, Madurai, Tamil Nadu, India

² Department of Computer Science and Engineering, Syed Ammal Engineering College, Ramanathapuram, Tamil Nadu, India

³ School of Computing Science and Engineering, VIT University, Chennai Campus, Chennai, Tamil Nadu, India

⁴ Shan Systems LLC, Jersey City, NJ, USA

Keywords Blind image steganalysis · Curvelet transformation · Evolutionary SVM model · Higher order statistics

1 Introduction

Steganography and steganalysis are new techniques of information security fields. Steganography is considered as the art of undetectable communication in which messages are embedded in innocuous looking objects, such as digital images, digital audio, TCP/IP data packets or even non-standard locations of Subscriber Identity Module/Universal Subscriber Identity Module (SIM/USIM) file system cards. The modified cover object is called stego object and the embedding process usually depends on a secret stego key shared between both communication parties. The goal of steganography is to communicate as many bits as possible without creating any detectable artifacts in the cover-object. If any suspicion about the secret communication is raised, then the goal is defeated. Steganography takes cryptography a step further taking the advantage of unused bits within the file structure or bits that are mostly undetectable if altered. A steganographic message rides secretly to its destination, unlike encrypted messages, which although undecipherable without the decryption key, can be identified as encrypted.

Steganalysis is taken as a countermeasure to steganography and is aimed at detecting the presence of hidden information from seemingly innocuous stego signals. Steganalysis can not only expedite the elevation of the steganography security by suitable quality evaluation criteria, but also can be used by lawbreakers for keeping out of the abuse of steganography. The study on steganalysis is focused on detection and attacking technique.

As is well known, steganography and watermarking constitute two main applications of information hiding techniques. Though both applications share many common principles in data embedding/extraction schemes, they differ in some criteria, such as robustness, embedding capacity, requirement of original messages, etc. In certain scenarios, content owners might need to determine the existence of hidden watermark in a multimedia object, when the authentication program fails to extract or match the targeted watermarks (due to inversion attack, geometric attacks, de-synchronization attacks etc.). In a possibly negative viewpoint, users may use this steganalytic feature to identify the existence of watermarks in an object. To summarize, steganalysis has promising applications to detect both the steganographic and watermarking schemes.

Steganalysis can be broadly classified into two categories. Active steganalysis deals with the estimation of the facts such as the embedded message length, locations of the hidden message, secret key used in embedding and finally the extraction of the entire message that is hidden. Passive steganalysis on the other hand detect only the presence or absence of a hidden message.

On the outset, deciding whether the cover media contains any secret message embedded in it or not is essential to steganalysis. Although it is uncomplicated to inspect suspicious objects and extract hidden messages by comparing them to the original versions, the restricted portability and accessibility of original cover-signals generally make blind steganalysis more attractive and reasonable in many practical applications. Blindness is meant to analyze stego-data without knowledge of the original signal and without exploiting the embedding algorithm. Hence, detecting the existence of hidden information becomes quite difficult and complex without exactly knowing which embedding algorithm, hiding domain, and steganographic keys were used. This motivates our current research: extracting low-dimensional, informative features that are significantly sensitive to data hiding process and devising a feature-based algorithm to classify multimedia objects as bearing hidden data or not. Our objective is not to

extract the hidden messages or to identify the existence of particular information (as it is in watermarking applications), but only to determine whether a multimedia object was modified by information hiding techniques. Once classified, the suspicious objects can then be inspected in detail by any particular data embedding/retrieving algorithms. This pre-process would particularly contribute to save time in active steganalysis.

1.1 Feature based Steganalysis

There are various methods to steganalyze data. Hiding information within electronic media will alter some of the media properties that may introduce few degradation or unusual characteristics. These characteristics may act as signatures that broadcast the existence of the embedded message, thus defeating the purpose of steganography. But such distortions cannot be detected easily by the human perceptible system. This distortion may be anomalous to the 'normal' carrier that when discovered may point to the existence of hidden information. Statistical steganalysis exploit these irregularities to provide the best discrimination power between the steganograms and the cover files.

Images steganalysis can be performed utilizing the texture operator to examine the pixel texture patterns within neighborhoods across the color planes. Steganographic and watermarking information inserted into a color image file, regardless of embedding algorithm, causes disturbances in the relationships between neighboring pixels and hence produce varying histograms which can also be used for steganalysis.

Feature based steganalysis can be considered as a pattern recognition process to decide which class a test image belongs to. The basic idea is that the various features calculated on cover images and on stego are statistically different. Thus the key issue for steganalysis is feature extraction. The features should be sensitive to the data hiding process. In other words, the features should be rather different for the signal without hidden message and for the stego-signal. The larger the difference, the better the features are. The features should be as general as possible, i.e., they are effective to all different types of images and different data hiding schemes. Often in practice it is very hard to achieve a high recognition rate with a single feature when the classification process such as steganalysis is complicated in nature. Therefore, multi-dimensional feature vectors should be used under those circumstances. It is desirable to have features in individual dimensions of the feature vector independent or at least less related to one another.

1.2 Machine learning for steganalysis

Machine learning or supervised learning based methods construct a classifier to differentiate between stego and non-stego objects using training examples. The features extracted from the image samples are given as training inputs to a learning machine. This includes both stego as well as non-stego documents. The learning classifier iteratively updates its classification rule based on its prediction and the ground truth. The final stego classifier is obtained upon convergence. When we train the classifier for a specific embedding algorithm a reasonably accurate detection can be achieved. Since the classifier is given multiple examples there is no need to assume prior statistical models for the images. The classifier learns a model by averaging over multiple examples.

In this paper we propose to steganalyse using the higher order statistics derived from the curvelet sub-band representation of the images. The derived features possess strong

discriminatory power which is very helpful in the distortion measurement process. Unlike previous work in image steganalysis that used the traditional image quality metrics, such as signal-to-noise ratio (SNR), correlation quality [1], and other such metrics, the proposed feature is designed specifically to detect modifications to pure image content. The paper employs the evolutionary-SVM as the machine learning component of the steganalyser. Experimental results with the chosen classifier, feature set and popular watermarking and steganographic strategies indicate that our approach is very accurate and promising in steganalysing image data.

2 Related work

2.1 Steganographic domains

In recent years, there has been significant research effort in steganalysis with primary focus on digital images. All the popular data hiding methods can be divided into two major classes: spatial domain based and transform domain based. Spatial domain based techniques are easy to implement providing high payload capacity but their robustness is weaker than its counter part. Least Significant Bit (LSB) addition [5, 45] or substitution [36, 38] method is the most popular hiding technique. These techniques operate on the principle of tuning the parameters (e.g., the payload or disturbance) so that the difference between the cover signal and the stego signal is little and imperceptible to the human eyes. Yet, computer statistical analysis is still promising to detect such a distinction that is difficult for humans to perceive. Some tools, such as StegoDos, S-Tools, and EzStego, provide spatial-domain-based steganographic techniques [31, 48].

Hiding can also be performed in the transform domain, e.g., Discrete Cosine Transform (DCT) [10, 12, 23, 28, 40, 43, 46], or Discrete Wavelet Domain (DWT) domain [33, 51]. Regardless of which domain, “significant” transform coefficients are often selected to mix with secret/perturbing signal in a way such that information hiding or watermarking is transparent to human eyes. For instance, Lie et al. [40] proposed a two-level data embedding scheme, in principle of additive spread spectrum and spectrum partition, for applications in copy control, access control, robust annotation, and content-based authentication. Cheng et al. [10] proposed an additive approach to hiding secret information in the DCT and DWT domains.

Passive and active warden styles Xu et al. [61] have viewed active steganalysis as blind sources separation (BSS) problem and solved it with Independent Component Analysis (ICA) algorithm under the assumption that embedded secret message is an independent, identically distributed (i.i.d) random sequence and independent to cover image. Passive, in contrast to active steganalysis detects only the presence or absence of a hidden message. Jessica Fridrich [17] described an improved version of passive steganalysis in which the features for the blind classifier are calculated in the wavelet domain as higher-order absolute moments of the noise residual. The features are calculated from the noise residual because it increases the features’ sensitivity to embedding, which leads to improved detection results. Geetha et al. [21] presented a LSB passive steganalysis approach for image steganalysis using close color pair signature and image quality metric as threshold.

Specific steganalysis is dedicated to only a given embedding algorithm. It may be very accurate for detecting images embedded with the given steganographic algorithm but it fails to detect those embedded with another algorithm. Techniques developed in [4, 20] are specific where they target to attack wavelet-, Outguess-, and LSB-based stego systems respectively. **Universal steganalysis** enables to detect stego images whatever the steganographic system be used. Because it can detect a larger class of stego images, it is generally less accurate for one given steganographic algorithm. Methods presented in [13, 41] are universal.

Several **multi-class steganalyzers** have been proposed in the recent years by various authors. Savoldi et al. [54] presented an effective multi-class steganalysis system, based on high-order wavelet statistics, capable of attributing stego images to four popular steganographic algorithms. Authors of [16, 49] constructed a practical forensic steganalysis tool for JPEG images that can properly analyze both single- and double-compressed stego images and classify them to selected current steganographic methods. **Binary steganalysis** has only two classes - the input is either tested positive (stego) or negative (pure).

Targeted attacks use the knowledge of the embedding algorithm [52], while blind approaches are not tailored to any specific embedding paradigm. **Blind approaches** can be thought of as practical embodiments of Cachin's [7] definition of steganographic security. It is assumed that 'natural images' can be characterized using a small set of numerical features. The distribution of features for natural cover images is then mapped out by computing the features for a large database of images. Using methods of artificial intelligence or pattern recognition, a classifier is then built to distinguish in the feature space between natural images and stego images. Avcibas et al. [1] were the first who proposed the idea to use a trained classifier to detect and to classify robust data hiding algorithms. They have proposed an image steganalytic system using image quality metrics as features. Avcibas et al. [2] also proposed a different set of features based on binary similarity measures between the lowest bit-planes.

2.2 Steganalysis through signal processing

Some steganalytic methods [4, 16, 18, 42] were proposed in the DCT domain. Manikopoulos et al. [42] estimate the probability density function (PDF) of DCT coefficients for the test image, and calculate its difference with respect to a reference PDF, which is then used as a feature input to a trained two-layer neural network for classification (identifying the existence of a hidden message in the test image). In their work, the reference PDF derived by averaging PDFs from all plain images in the database is required for this similarity measure. Generally, representation of a set of plain images in terms of a reference PDF is questionable for the Joint Photographic Experts Groups (JPEG) image format. The detection first starts by decompressing the JPEG stego image, geometrically distorting it and recompressing. The paper investigates the use of macroscopic statistics that also predictably changes with the embedded message length. The details of the detection methodology are explained on the F5 algorithm and Outguess.

Also Fridrich et al. [16] described that a modified image block will most likely become saturated (i.e., at least one pixel with the gray value 0 or 255) in a JPEG-format stego-image after information hiding. If no saturated blocks can be found, there will be no secret messages therein. Otherwise, a spatial-domain steganalytic method [4] mentioned earlier can be used to

analyze these saturated blocks. In [4], the author modeled the common steganographic schemes as a linear transform between the cover and stego images, which can be estimated after at least two copies of a stego image. This is similar to a blind source separation problem that can be solved by using the independent component analysis (ICA) [41] technique. In [13], a steganalytic scheme was devised to deal with information hiding schemes mixing a secret and a cover signal in an addition rule. The phenomenon, that the centre of mass of the histogram characteristic function in a stego image moves left or remains the same to that of the cover image, was observed and exploited to distinguish stego images from plain ones. Jessica Fridrich [19] presented general methodology for developing attacks on steganographic systems

2.3 Steganalysis using distortion measures

Our research is based on the extension of the fact that hiding information in digital media requires alterations of the signal properties that introduce some form of degradation, no matter how small. The schematic description of the additive noise model of steganography is shown in Fig. 1. These degradations can act as signatures that could be used to reveal the existence of a hidden message. The idea that the addition of a watermark or message leaves unique artifacts, and which can be detected using the various distortion metrics i.e., Image Quality Measures (IQM) is introduced in [1].

It is noticed that most of the steganalytic schemes were designed either in specific operating domain, or even for particular steganographic algorithm. Building a universal steganalytic system is, up to now, a challenging exercise. In [39], Lie et al. has modelled a universal steganalyzer that operates to distinguish stego images from clean images using two features

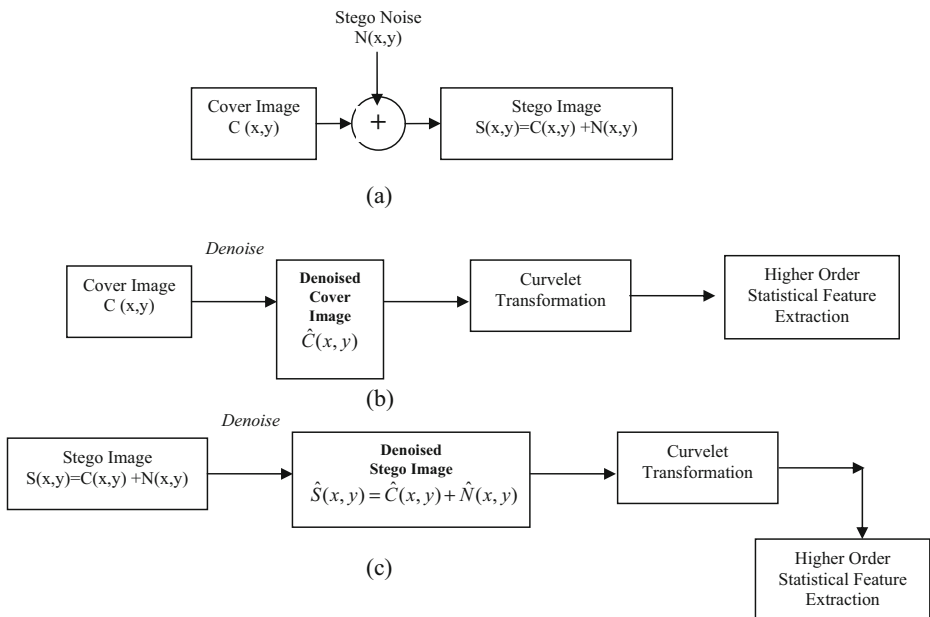


Fig. 1 Schematic descriptions of (a) Additive noise audio-steganography model, (b) denoising a cover-image object, and (c) denoising a stego-image object

only: namely, gradient energy and statistical variance of the Laplacian parameter. The system lacks the ability to strongly attack a wavelet based stego system. But that can be solved by using a feature that is more sensitive to such embedding strategy

2.4 Steganalysis using high-order statistics

There are many works reporting that high-order statistics are very effective in differentiating stego-images from cover-images. In [15], Farid proposed a general steganalysis algorithm based on image higher-order statistics. In this method, a statistical model based on the first (mean) and higher-order (variance, skewness, and kurtosis) magnitude statistics, extracted from wavelet decomposition, is used for image steganography detection. In [25], a steganalysis method based on the moments of the histogram characteristic function was proposed. It has been proved that, after a message is embedded into an image, the mass center (the first moment) of histogram characteristic function will decrease. In [27], Holotyak et al. used higher-order moments of the PDF of the estimated stego-object in the finest wavelet level to construct the feature vectors. Due to the limited number of features used in the steganalysis technique proposed in [25], Shi et al. proposed the use of statistical moments of the characteristic functions of the wavelet sub-bands [56]. Because the n^{th} statistical moment of a wavelet characteristic function is related to the n^{th} derivative of the corresponding wavelet histogram, the constructed 39-dimensional feature vector has proved to be sensitive to embedded data. Usually, the steganalysis algorithms based on the high-order statistics can achieve satisfactory performance on image files, regardless of the underlying embedding algorithm. In [60], the authors modeled the secret message embedded by LSB matching as an independent noise to the image. They employed the co-occurrence matrix to model the differences with the small absolute value in order to extract features. A classifier model is built using support vector machine which is trained with the features so as to identify a test image either an original or a stego image. The experimental results were very promising. The authors of [59] also followed a similar model to destroy LSB matching stego systems. They revealed that the histogram of the differences between pixel gray values is smoothed by the stego bits regardless of a large distance between the pixels. Features are extracted from characteristic function of difference histogram and are calibrated with an image generated by average operation. Finally a support vector machine (SVM) classifier is trained with these features. The experimental results proved that this system detected LSB matching promisingly well. However, both of these systems are targeted steganalysers. They operate remarkably well for LSB matching based algorithms only.

However, since the data-embedding method is typically unknown prior to detection, we focus on the design of a unified blind steganalysis algorithm to detect the presence of steganography independent of the steganography algorithms used. Moreover, we focus on passive detection as opposed to active warden steganalysis [1] which aim to detect and modify the hidden content. In this work, we employed the higher order statistical moments as features that were collected from an effective image sub-band representation i.e., Curvelet transform domain.

3 Design of the proposed steganalyser

For a group of given data samples (e.g., coefficients in any sub-band of the image multi-resolution representation), the first important step of machine-learning-based image

steganalysis is to choose representative features. Then, a decision function is built based on the feature vectors extracted from the two classes of training images: photographic cover images and stego-images with hidden information. The performance of the steganalyser depends on the discrimination capabilities of the features. Also, if the feature vector has low dimension, the computational complexity of learning and implementing the decision function will decrease. In summary, we need to find informative, low-dimensional features extraction.

The critical component for the success of the steganalyser's performance is the feature extraction phase. In this paper we investigate the feature extraction problem for image steganalysis from the following perspectives.

- 1) **Image sub-band decomposition.** We select an appropriate image sub-band representation for a given image. For instance, Lyu's image representation includes wavelet sub-band coefficients and their cross-sub-band prediction errors [41]. However, we have discovered that decomposing the image based on curvelet transformation is more beneficial than others in the steganalysis view. (see Section 3.1)
- 2) **Choice of features.** We consider empirical probability density function (PDF) and characteristic function (CF) moments as steganalytic features. These moments are good at capturing different statistical changes caused by data embedding process; (see Section 3.2). These features act as telltale evidences in classifying the image as stego-bearing or not.
- 3) **Feature evaluation and selection.** All features are not equally valuable to the learning system. Also, using too many features is undesirable in terms of classification performance due to the curse of dimensionality [14]. Also, if the feature vector has low dimension, the computational complexity of learning and implementing the decision function will decrease. In summary, we need to find informative, low-dimensional features. In Section 3.3, we apply evolutionary algorithm for feature dimensionality reduction and employ SVM algorithm for classification, thereby improving classification performance.

Finally, the proposed image steganalyser is implemented and the results are reported in Section 4.

3.1 Image Sub-band decomposition: choice of curvelet transforms

The decomposition of images using basis functions that are localized in spatial position, orientation and scale have proven extremely useful in image compression, image coding, noise removal and texture synthesis. One reason is that such decompositions exhibit statistical regularities that can be exploited. The last two decades have seen tremendous activity in the development of new mathematical and computational tools based on multi-scale ideas. New transforms may be very significant for practical concerns. For instance, the potential for sparsity of wavelet expansions led the way to very successful applications in areas such as signal/image compression or denoising and feature extraction/recognition. A special member of this emerging family of multi-scale geometric transforms is the curvelet transform [8, 57] which was developed in the last few years in an attempt to overcome inherent limitations of traditional multi-scale representations such as wavelets.

To process 2-D image signals, the 2-D wavelet transform, composed of the tensor product of two 1-D wavelet basis functions, takes advantage of the separable transform kernels to

realize the wavelet transform in horizontal firstly and then in vertical. Such kernels of the 2-D wavelet transform are isotropic, leading to that the local transform modulus maxima only reflect the positions of those maxima are across edge, instead of along edge. However, singularities in most of images are characterized by lines and curves, which seriously reduces the approximation efficiency of wavelet. In this circumstance, the traditional wavelet transform is limited in the field of image processing. To overcome this difficulty, Donoho et al. propose curvelet transform theory whose anisotropic feature is very helpful to effectively express the edges. Curvelet transform can sparsely characterize the high-dimensional signals which have line, curve or hyper-plane singularities and the approximation efficiency is one magnitude order higher than wavelet transform [57].

Conceptually, the curvelet transform is a multi-scale pyramid with many directions and positions at each length scale, and needle-shaped elements at fine scales. This pyramid is nonstandard, however. Indeed, curvelets have useful geometric features that set them apart from wavelets and the likes. For instance, curvelets obey a parabolic scaling relation which says that at scale 2^{-j} , each element has an envelope which is aligned along a “ridge” of length $2^{-j/2}$ and width 2^{-j} .

The curvelet transform is a higher dimensional generalization of the wavelet transform designed to represent images at different scales and different angles. Curvelets enjoy two unique mathematical properties, namely:

- Curved singularities can be well approximated with very few coefficients and in a non-adaptive manner - hence the name “curvelets”.
- Curvelets remain coherent waveforms under the action of the wave equation in a smooth medium.

The application of curvelet statistics for image steganalysis is relatively unexplored. The experimental results made clear that the curvelet transform is very relevant for steganalysis applications. The following potentiality of curvelet transforms provide substantial amount of evidence supporting our claim:

Sparse representations by curvelets The curvelet representation is far more effective for representing objects with edges than wavelets or more traditional representations. In fact, [8] proves that curvelets provide optimally sparse representations of C2 objects with C2 edges. Figure 2 illustrates the decomposition of the original image into sub-bands followed by the spatial partitioning of each sub-band and later the ridgelet transform is applied to each block. Thus they are candidates for informative, low-dimensional features. Hence for image steganalysis, curvelet coefficients may be beneficial than other transform coefficients like wavelets. In general, improved sparsity leads to reduced time complexity in calculating the features. Hence, steganalysis based on the curvelet coefficients may benefit from provably superior asymptotic properties.

Sparse component analysis In computer vision, there has been an interesting series of experiments whose aim is to describe the ‘sparse components’ of images. Of special interest is the work of Olshausen [47] who set up computer experiments for empirically discovering the basis that best represents a database of 16 by 16 image patches.

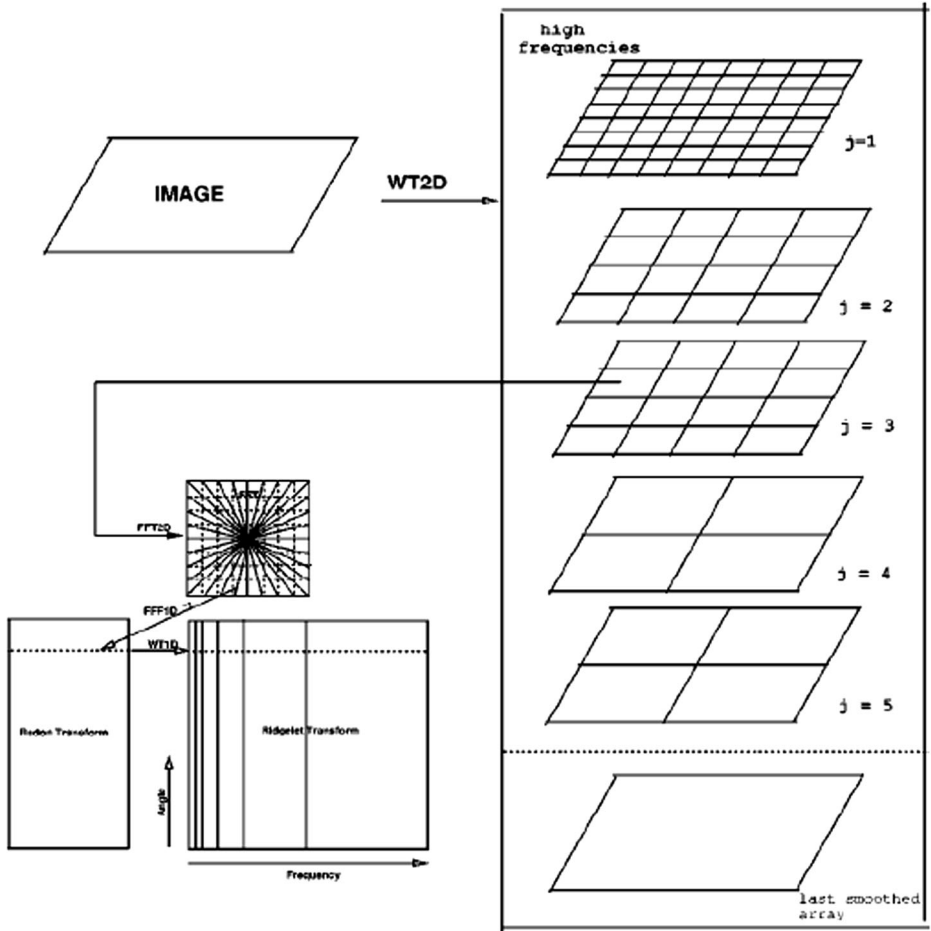


Fig. 2 Curvelet transform flow graph

Although this experiment is limited in scale, they discovered that the best basis is a collection of needle shaped filters occurring at various scales, locations and orientations. They resembled the curvelets. Similarly, when steganalysis is handled like a pattern matching task, the curvelet coefficients provide sub-band representations that respond significantly to the distortions induced due to data embedding.

Numerical experiments Huo [29] studies sparse decompositions of images in a dictionary of waveforms including wavelet bases and curvelets. They apply the Basis Pursuit (BP) in this setting and obtained sparse syntheses. BP gives an ‘equal’ chance to every member of the dictionary and yet, Donoho and Huo observe that BP preferably selects curvelets over wavelets, except for possibly the very coarse scales and the finest scale. Their experiment seems to indicate that curvelets are better for representing image data than pre-existing mathematical representations. Our experiments also defend the same idea that curvelets yield better features.

The basic process of the digital realization for curvelet transform is given as follows [57]. The transformation yields six sub-bands, a multi-scale representation across scale, orientation and phase.

We propose to decompose the given image in to six sub-bands $B_i, i=1,2,3,4,5,6$ as in [29] using curvelet transformation as in Listing 1. Let us denote by \mathfrak{R}_1 the set of these 6 curvelet sub-bands plus the image itself. The noise residual component for a cover image and its stego-image possess different statistics, which are useful in steganalysis. Since curvelet coefficients possess strong intra and inter subband dependencies, we propose to construct a set \mathfrak{R}_2 of six noise residual sub-bands to exploit these dependencies as follows. Take a sub-band coefficient $B_i(m,n)$ as an example, where (m,n) denoted the spatial co-ordinates at band i . The magnitude of the denoised component of this band can be computed by applying Wiener filter over these coefficients.

Listing 1. Algorithm for Digital Curvelet Transform

Algorithm : *The digital curvelet transform.*

1. *Sub-band Decomposition: A bank of filters is defined. The image f is filtered into sub-bands with à trous algorithm*

$$f \rightarrow (P_0f, \Delta_1f, \Delta_2f, \dots) \tag{1}$$

The different sub-bands $\Delta_s f$ contain details about 2^{-2s} wide.
2. *Smooth Partitioning: Each sub-band is smoothly windowed into "squares" of an appropriate scale*

$$\Delta_s f \rightarrow (w_Q \Delta_s f), Q \in Q_s \tag{2}$$

where w_Q is a collection of smooth window localized around dyadic squares

$$Q = [k_1 / 2^s, (k_1 + 1) / 2^s] \times [k_2 / 2^s, (k_2 + 1) / 2^s] \tag{3}$$
3. *Renormalization: Each resulting square is renormalized to unit scale*

$$g_Q = T_Q^{-1} (w_Q \Delta_s f), Q \in Q_s \tag{4}$$

where $T_Q f(x_1, x_2) = 2^s f(2^s x_1 - k_1, 2^s x_2 - k_2)$ is a renormalization operator.
4. *Ridgelet Analysis: Each square is analyzed in the orthonormal ridgelet system. This is a system of basis elements P_λ making an orthonormal basis for $L^2(\mathbb{R}^2)$:*

$$\alpha_\mu = \langle g_Q, P_\lambda \rangle \tag{5}$$

3.2 Feature extraction

Given this image decomposition and sub-band construction, the statistical model is composed of the higher order statistics – empirical PDF moments and empirical CF moments as steganalytic features. These statistics characterize the basic image’s coefficient distributions. The second set of statistics is based on the noise component of the stego-image in the curvelet domain. The noise component was obtained using the denoising filter as in [57]. We reiterate that the denoising step increases the SNR between the stego image and the cover image, thus making the features calculated from the noise residual more sensitive to embedding and less sensitive to image content. The denoising filter is designed to remove Gaussian noise from images under the assumption that the stego image is an additive mixture of a non-stationary Gaussian signal (the cover image) and a stationary Gaussian signal with a known variance (the noise). As the filtering is

performed in the curvelet domain, all our features (statistical moments) are calculated as higher order moments of the noise residual in the curvelet domain. The functional framework of the proposed steganalyser is given in Fig. 3 for an overall understanding.

3.2.1 PDF moments

For a sequence $x=(x_1,x_2,x_3,\dots,x_N)$ of independent and identically distributed (i.i.d.) samples drawn from an unknown PDF $p(x)$, a natural choice of descriptive statistics is a set of empirical PDF moments. The n^{th} empirical PDF moment is given by

$$\hat{\mu}_n = \frac{1}{N} \sum_{i=1}^N x_i^n, n \geq 1 \tag{6}$$

which is an unbiased estimate of the n^{th} PDF moment

$$\mu_n = EX^n = \int_{-\infty}^{\infty} p(x)x^n dx \tag{7}$$

Mean, variance, skewness, and kurtosis of the PDF $p(x)$ form the first four moments, respectively. Empirical PDF moments were used by Lyu [41] and Goljan et al. [22].

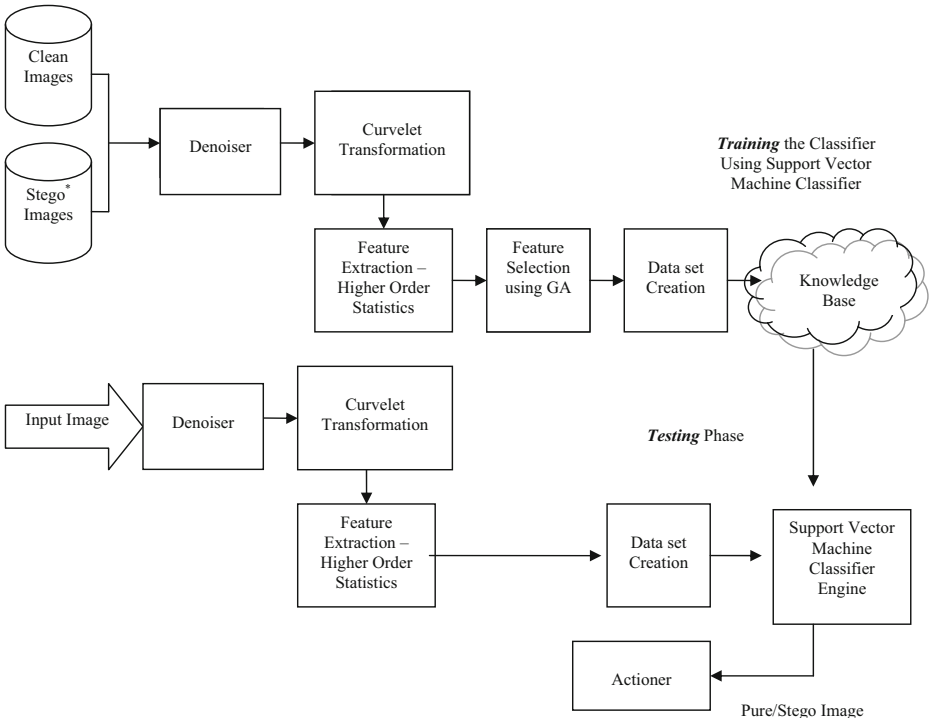


Fig. 3 Functional model of the proposed Image Steganalyzer

The n^{th} empirical absolute PDF moment given by

$$\hat{\mu}_n^A = \frac{1}{N} \sum_{i=1}^N |x_i|^n, n \geq 1 \tag{8}$$

which is an estimate of the n^{th} absolute PDF moment

$$\mu_n^A = E|X|^n = \int_{-\infty}^{\infty} p(x)|x|^n dx \tag{9}$$

From (9) and (11), $p(x)$ is weighted by x^n and $|x|^n$, respectively, and any change in the tails of $p(x)$ is polynomially amplified in PDF moments. As is well known, $\hat{\mu}_n$ and μ_n in (8) and (9) relate to the n^{th} derivative of the $CF\Phi(t)$ of the PDF $p(x)$ at $t=0$ by

$$\hat{\mu}_n \approx \mu_n = j^{-n} \frac{d^n}{dt^n} \Phi(t)|_{t=0} \tag{10}$$

Moreover

$$\hat{\mu}_n \approx \mu_n \geq |\mu_n| = \left[\frac{d^n}{dt^n} \Phi(t) \right]_{t=0} \tag{11}$$

For a heavy-tailed PDF, μ_n is large and it follows from (12) that $\Phi(t)$ has large derivatives at the origin (i.e., it is peaky).

3.2.2 CF moments

Analogously, for the $CF\Phi(t)$, its n^{th} moment is defined by

$$\varphi_n = \int_{-\infty}^{\infty} \Phi(t)t^n dt \tag{12}$$

and its n^{th} absolute moment is

$$\varphi_n^A = \int_{-\infty}^{\infty} |\Phi(t)||t|^n dt \tag{13}$$

In the above integral, $|\Phi(t)|$ is weighted by $|t|^n$. Any change in the tails of $|\Phi(t)|$, which correspond to the high-frequency components of $p(x)$, is thus polynomially amplified. Similar to (12) and (13), the CF moments φ_n and φ_n^A relate to the n^{th} derivative of $p(x)$ at $x=0$ by

$$\varphi_n = j^n 2\pi \frac{d^n}{dx^n} p(x)|_{x=0} \tag{14}$$

and

$$\varphi_n^A \geq |\varphi_n| = 2\pi \left| \frac{d^n}{dx^n} p(x) \right|_{x=0} \tag{15}$$

If a $CF\Phi(t)$ has heavy tails and φ_n^A is large, then the corresponding PDF $p(x)$ is peaky. Equations (10), (11), (14), and (15) reveal a duality between PDF moments and CF moments that follows from the duality between the PDF $p(x)$ and its $CF\Phi(t)$.

To obtain the corresponding empirical CF moments from a sample sequence x , we first estimate the PDF $p(x)$ using an M-bin histogram $\{h(m)\}_{m=0}^{M-1}$. Let $Z = 2^{\lceil \log_2 M \rceil}$. The Z -point discrete $CF\{\Phi(z)\}_{z=0}^{Z-1}$ is then defined as

$$\Phi(z) = \sum_{m=0}^{M-1} h(m) \exp\left\langle \frac{j2\pi mz}{Z} \right\rangle, 0 \leq z \leq Z-1 \tag{16}$$

which is analogous to $\Phi(t)$ defined in (1) and can be easily computed using the fast Fourier transform (FFT) algorithms. Similar to (2), the histogram

$$h(m) = \frac{1}{z} \sum_{z=0}^{Z-1} \Phi(z) \exp\left\langle \frac{-j2\pi mz}{Z} \right\rangle, 0 \leq m \leq M-1 \tag{17}$$

can be recovered from the discrete $CF\Phi(z)$.

The above mentioned features are extracted as per the algorithm given in Listing 2.

We set the parameter $\sigma_0^2=0.5$, which corresponds to the variance of the stego signal for an image fully embedded with ± 1 embedding [27].

The algorithm yields 30 (6 bands and 5 statistics from each band) statistics from noise residue and 30 statistics from the image. Combining these statistics we get a total of 60 statistics that form a feature vector which is used to discriminate between images that contain hidden messages and those that do not.

Listing 2. Algorithm for Feature Calculation

Algorithm: Feature Calculation

Step 1: Calculate the first level curvelet decomposition of the stego image as in [1]. Denote the six sub-bands as $B_k(m, n)$ where k runs through the index 1 to 6 and (m, n) runs through some index J

Step 2: In each sub-band, estimate the local variance of the cover image for each wavelet coefficient using the MAP estimator for 4 sizes of a square $N \times N$ neighbourhood, for $N \in \{3, 5, 7, 9\}$

$$\hat{\sigma}_N^2(m, n) = \max(0, \frac{1}{N^2} \sum_{(m,n) \in N} w^2(m, n) - \sigma_0^2), (m, n) \in J \tag{18}$$

Take the minimum of the 4 variances as the final estimate,

$$\hat{\sigma}^2(m, n) = \min(\hat{\sigma}_3^2(m, n), \hat{\sigma}_5^2(m, n), \hat{\sigma}_7^2(m, n), \hat{\sigma}_9^2(m, n)), (m, n) \in J \tag{19}$$

Step 3: The denoised curvelet coefficients for the first sub-bands are obtained using the Wiener filter

$$b_{1_den}(m, n) = b_1(m, n) \frac{\hat{\sigma}^2(m, n)}{\hat{\sigma}^2(m, n) + \hat{\sigma}_0^2}, (m, n) \in J \tag{20}$$

and similarly for the other five sub-bands also.

Step 4: Calculate the noise residual in each sub-band

$$r_{1_den}(m, n) = b_1(m, n) - b_{1_den}(m, n), (m, n) \in J \tag{21}$$

and similarly $r_{k_den}(m, n), (m, n) \in J$ and $k = 2$ to 6

Step 5: Calculate the empirical PDF moments and CF moment values of the six sub-band coefficients of the image and also of each noise residual. (60 features altogether)

3.3 Evolutionary SVM classifier (E SVMC) as the machine learning paradigm

Genetic Algorithms (GAs) [26] have been successfully applied to solve search and optimization problems. The basic idea of a GA is to search a hypothesis space to find the best hypothesis. A pool of initial hypotheses called a population is randomly generated and each hypothesis is evaluated with a fitness function. Hypotheses with greater fitness have higher probability of being chosen to create the next generation. Some fraction of the best hypotheses may be retrained into the next generation, the rest undergo genetic operations such as crossover and mutation to generate new hypotheses. The size of a population is the same for all generations in our implementation. This process is iterated until either a predefined fitness criterion is met or the preset maximum number of generations is reached. Various machine learning techniques, starting from linear regression techniques [1] to neural networks [55] have been applied for steganalysis. The problem of steganalysis falls under linearly non-separable category. The application of SVM has proved to be beneficial in many works [15, 24, 27]. Input vector is converted into a high dimensional feature space, which enables to separate non-linear separable spaces into proper classes. Next, when it formulates the boundary between classes, it determines the effectiveness of each feature, in order to find optimal boundary. It makes optimal boundary between classes.

The fundamental idea of the proposed stego-classifier system is to employ GA to explore efficiently the feature space of all possible subsets of the 60-dimension feature set so as to identify the feature subsets which possess low order and high discriminatory power. The dimensionality reduction process produces a greater impact both on enhancing the detection accuracy as well as minimizing the computational complexity of the classifier. In order to achieve this objective, the fitness evaluation should involve feature size and classification performance as direct measures rather than measures such as the ranking methods as used in conventional systems. The flow of the hybrid model is shown in Fig. 4.

For a group of given data samples (e.g., coefficients in any sub-band of the image multi-resolution representation), the first important step of machine-learning-based image steganalysis is to choose representative features. Then, a decision function is built based on the feature vectors extracted from the two classes of training images: photographic cover

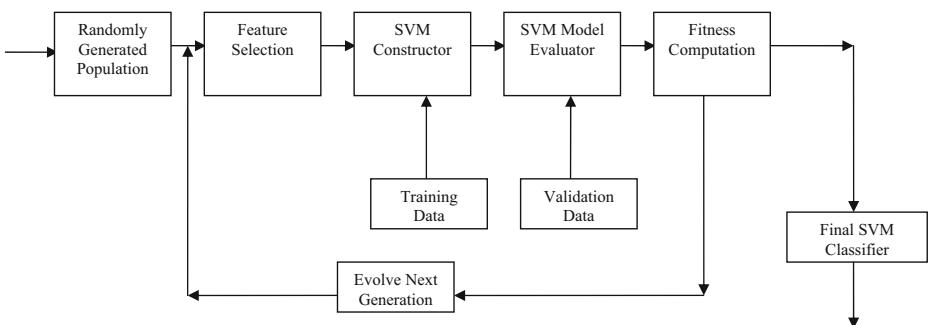


Fig. 4 Evolutionary-SVM Classifier

images and stego-images with hidden information. The performance of the decision rule depends on the discrimination capabilities of the features. Also, if the feature vector has low dimension, the computational complexity of learning and implementing the decision function will decrease. In summary, we need to find informative, low-dimensional features.

To start with, the training data set with the 60 features extracted from image files (clean as well as stego) corresponding to examples of concepts, is provided as inputs for which the support vectors have to be induced. GA is used to explore the complete solution space of all feature subsets of the given feature set where features sets which achieve better classification performance using smaller dimensionality feature sets are preferred. Each of the selected feature subsets is evaluated (its fitness measured) by testing the support vector model produced. The above process is iterated along evolutionary lines and the best feature subset observed is then recommended to be used in the actual design of the image stego classification system.

The proposed hybrid learning model will perform better by identifying better feature subsets than that of any other feature selection methods owing to two primary reasons – (i) The power of GA is exploited efficiently to investigate the non-linear interactions of the selected sub set of features; (ii) By using SVM in the evaluation loop, we have an effective mechanism for measuring the directly classification's accuracy.

3.3.1 Chromosome's encoding

For a GA to efficiently search such large spaces, the representation/encoding and the fitness function – both are chosen carefully. In the present case of image steganalysis, there exists a very natural representation of the space of all possible subsets of a feature set, namely, a binary string of fixed-length (60) representation in which the value of the i^{th} gene either $\{0,1\}$ indicates whether or not the i^{th} feature ($i=1, \dots, 25$) from the complete feature set is included in that specified feature subset. Hence, each individual chromosome in the GA population is encoded as a fixed-length string i.e., 60-bit binary string denoting a particular subset of the given feature set. This encoding procedure offers us an advantage of directly using a standard and well understood GA without any major modifications.

3.3.2 Fitness function

Each chromosome member of the current GA population denotes a competing feature subset which has to be evaluated for fitness feedback in the evolutionary process. This can be realized by invoking SVM classifier with the specified feature subset of that iteration and a corresponding training data set reduced to include only those selected features. The SVM evolved is then tested for classification accuracy on a set of unseen evaluation data.

We target to both enhance the detection accuracy of the steganalyser as well as minimise the number of features which could be indirectly achieved by maximizing the specificity and sensitivity scores of the classifier. Hence this knowledge is fairly imparted into the model in the form of fitness function in the GA module. Accordingly the fitness function is designed as

$$Fitness = w_1 \times Specificity + w_2 \times Sensitivity + w_3 \times \frac{1}{Count_of_ones} \quad (22)$$

Where

$$\text{Specificity} = \frac{TP}{TP + FN}$$

$$\text{Sensitivity} = \frac{TN}{TN + FP}$$

Fitness of a given chromosome is thus evaluated based upon the sensitivity and specificity scores from the validation dataset and number of features present in a chromosome. Here True Positive (TP) and True Negative (TN) are the number of images correctly classified in stego and clean image classes respectively. Similarly FP and FN are the number of records incorrectly classified in stego and clean image classes respectively. Count of ones is the total number of ones present in the given chromosome. If two feature subsets attain equal performance, while they have different number of features, obviously the subset with fewer features will have to be chosen. Among specificity, sensitivity and number of features, number of features is least concerned, so more weightage is given to specificity ($w_1 = 0.4$) and sensitivity ($w_2 = 0.4$) while the number of selected features is weighed only $w_3 = 0.2$.

3.3.3 Genetic operators

The other genetic operators like selection, cross-over and mutation used are that of the general simple GA's viz., tournament selection, uniform cross over and the simple mutation.

Several criteria from the pattern recognition and machine learning literature may be used to evaluate the usefulness of a feature in discriminating between classes [14]. In this paper, we use a non-linear SVM as adapted in [15, 27] as the classifier that provides the best classification accuracy. However, as a pre-processing step, the feature reduction phase is performed. For feature selection, genetic algorithm (GA) is utilized. The operational flow of the GA-ensemble SVM classifier is depicted as in Fig. 4.

Listing 3 shows an abstracted description of the algorithm execution. As a whole, the execution of the combined GA and SVM algorithm is an iterative procedure (GA-SVM procedure). Each iteration results in a group of support vectors. After n iterations, a collection of SVMs will be obtained from which the best could be used to classify. The SVM model with the highest specificity and sensitivity is identified to be the best model (Figs. 6 and 7).

The initial population is randomly generated. Every individual of the population has 60 genes, each of which represents a statistics of the input data and can be assigned to 1 or 0. 1 means the represented feature is selected during constructing SVM classifier; 0 means it is not selected. As a result, each individual in the population represents a choice of available features. For each individual in the current population, a SVM classifier is built using the program [9]. This resulting SVM model is then tested over clean and stego data sets. The specificity, sensitivity and number of features selected (i.e., number of ones) are measured. The fitness of this individual is the linear sum of these components. They are given weight values depending upon the requirement of the system.

Sensitivity and specificity are well-established statistical measures of the performance of a binary classification test. Sensitivity measures the proportion of stego images that are correctly identified and Specificity measures the proportion of clean images that are correctly identified. Summarily sensitivity quantifies the avoiding of false negatives, as specificity does for false positives. An ideal predictor would be characterised to be 100 % sensitive and 100 % specific.

Hence the weight values in the fitness function are adjusted appropriately so as to achieve perfect prediction in terms of the specificity and sensitivity values. The main aim of this analysis is to find the relationship between specificity and sensitivity and to find optimal weight values for w_1 and w_2 so as to achieve 100 % accurate prediction. The fitness function also includes a third component namely number of stego sensitive features selected. Among all these components, specificity and sensitivity are treated as the highest priority characteristics while the number of features is least concerned. The summation of all the weight values should equate to 1 so as to facilitate efficient GA search. Considering all these factors, more weightage is given to specificity ($w_1 = 0.4$) and sensitivity ($w_2 = 0.4$) while the number of selected features is weighed only $w_3 = 0.2$. Further the weight values assigned as $w_1 = 0.4$, $w_2 = 0.4$ and $w_3 = 0.2$ yielded good classification values of all individuals of the current population have been computed, the GA begins to generate next generation as follows: performance with fewer features. Depending on the requirement priorities, the weight values can be adjusted appropriately. The higher the accuracy, the better is the fitness of the individual. Once the fitness values of all individuals of the current population have been computed, the GA begins to generate next generation as follows:

- (1) Choose individuals according to Rank Selection method [3].
- (2) Use two point cross-over to exchange genes between parents to create offspring.
- (3) Perform a bit level mutation to each offspring.
- (4) Keep two elite parents and replace all other individuals of current population with offspring.

The procedure above is iteratively executed until the maximum number of generations is reached. Finally, the best individual of the last generation is chosen to build the final SVM classifier, which is tested on the test data set. A detailed description of the adapted algorithm is shown in Listing 3 and 4.

Listing 3. Algorithm for evolutionary feature selection

Algorithm : Evolutionary _Feature_Selection()	
Input :	Encoded binary string of length 60 (one bit for each statistical feature), number of generations=200, and population size=50, Cross over probability $P_c=0.8$, Mutation Probability $P_m=0.02$, $w_1 = 0.4$, $w_2 = 0.4$ and $w_3 = 0.2$
Output :	A set of optimized features that maximize the specificity and sensitivity of the model.
<ol style="list-style-type: none"> 1. Initialize the population randomly 2. $N =$ total number of records in the training set 3. For each chromosome in the new population 4. Apply uniform crossover operator to the chromosome with a probability of P_c. 5. Apply mutation operator to the chromosome with a probability of P_m. 6. Evaluate $Fitness = w_1 \times Specificity + w_2 \times Sensitivity + w_3 \times \frac{1}{Count_of_ones}$ 7. Select the top best 50% chromosomes into new population using Tournament Selection operator. 8. If number of generations is not reached, go to step 3. 	

Listing 4. Algorithm for evolutionary SVM**Input:**

Training_Set be the training set of instances to be classified.
Selected_Feature be a set of attributes for the description of each instance.
P be an initial population, each a 60-bit binary string of which is a set of features chosen at random. (a bit with value 1 represents the selection of that attribute, a 0 means deduction of that attribute)
Stop(Training_Set) be a stopping criterion of instance splitting procedure.
ESV(Selected_Features, Training_Set) be an evaluation function for attributes where $Selected_Feature_i \in Selected_Features$.

Output:

The optimal SVM model i.e., the SVM with the highest specificity and sensitivity scores as the steganalyser.

PROCEDURE Error-Complexity (SV)

1. False positive rate = $\frac{\text{No. of clean image files misclassified as stego - bearing}}{\text{Total No. of clean image files}}$
2. False negative rate = $\frac{\text{No. of stego image files misclassified as clean benign files}}{\text{Total No. of stego image files}}$
3. Error rate = $\frac{\text{False Positive Rate} + \text{False Negative Rate}}{2}$

Return error rate.

PROCEDURE GA- SVM (E)

// Based on the population, a SVM is generated by this procedure:

$$SV = SVMC(Training_Set)$$

IF *Error_Complexity(SV)* > 0.7 **THEN**

// It means that the SVM is not optimal and the algorithm has to generate a new population which may be better than the present one...

GA(P) ;

Go to the beginning of the algorithm.

else

Return from the algorithm, as the SVM is optimal.

PROCEDURE SVMC(Training_Set)

Training set is of the format (x_i, y_i) , where x_i indicates i^{th} instance's feature vector and y_i indicates the class label (stego or clean), $\alpha \Leftarrow 0$ or $\alpha \Leftarrow$ partially trained SVM (indicates the Lagrange multiplier)

1. $C \Leftarrow$ a random value (represents the no. of support vectors)
2. Repeat
3. For all pairs of instances $(x_i, y_i), (x_j, y_j)$ in the *Training_Set* do
4. Optimize α_i and α_j
5. End for
6. Until no changes in α or other resource constraint criteria met

Return only the support vectors with $(\alpha_i > 0)$

4 Experimental results

In our experiments, the discrimination performance of higher order statistics gathered from curvelet coefficients as features is analyzed first. Then the classification performance of our steganalyzer under the prepared test image set is reported. The impacts of embedding rate and the effectiveness of the selected are explored.

4.1 Preparation of test images and schemes

The design of experiments is important in evaluating our steganalytic algorithm. The key considerations include the following.

- 1) First, from the point of “generalization”, the proposed content independent image features and associated classifier should be capable of identifying the existence of hidden data which are possibly generated by using various kinds of embedding methods, regardless of steganography or watermarking, and regardless of spatial or transform-domain operations.
- 2) Second, in outlook of “performance”, the classifier should, on the one hand, detect hidden data as likely as possible (regardless of how transparent the embedded secret information is), and on the other hand, keep false alarms to as few as possible for plain images.
- 3) Third, in view of “robustness”, the classifier should be capable of differentiating the effect of ordinary image processing operations (such as filtering, enhancement, etc.) from that of data embedding.

On the grounds of the above considerations, six published methods based on two types of principles, LSB embedding and spread spectrum, were chosen for evaluation. Seventh scheme based on wavelet domain is chosen to validate the ability of the system to attack any new stego scheme. These systems are chosen for steganalysis since they include all possible data hiding mechanisms. The system is not trained with the stego patterns of scheme 7 and 8.

scheme #1: Digimarc [50]

scheme #2: PGS [35]

scheme #3: Cox et al.'s [35]

scheme #4: S-Tools [6]

scheme #5: Steganos [58]

scheme #6: JSteg [34]

They can be further categorized into:

- 1) steganography (#4, #5, #6) or watermarking (#1,#2,#3) purpose;
- 2) spatial (#2, #4, #5), or transform (#1, #3, #6) domain operation.

For further testing and to verify the effectiveness of the features selected, we select an extra scheme based on the wavelet domain:

3) scheme #7: Kim et al.'s method [], scheme #8: Solanki et al.'s YASS method [32]

It is expected that the difference between a cover image and its stego version can be easily detected when more secret messages are embedded. Hence the capacity of the payload of a steganography scheme should be taken into account in evaluating the detection capacity of a steganalytic classifier. To depict this, the payload capacity characterizing a scheme which is defined as the ratio between the number of embedded bits and the number of pixels in an image, is used.

To test the performance of the proposed method, our cover image dataset consists of 200 images with a dimension of 256×256 8-bit gray-level photographic images, including standard test images such as Lena, Baboon, and also images from [30]. Our cover images contain a wide range of outdoor/indoor and daylight/night scenes, including nature (e.g., landscapes, trees, flowers, and animals), portraits, manmade objects (e.g., ornaments, kitchen tools, architectures, cars, signs, and neon lights), etc. This database is augmented with the stego versions of these images using the above mentioned seven schemes, at various embedding rates. Some of the sample images used for experimentation are shown in Fig. 5. Also a separate image set was generated by applying the image processing techniques like JPEG compression (at several quality factors), low-pass filtering, image sharpening etc. Our generation procedure is aimed at making even contributions to database images from different embedding schemes, from original or stego, and from processed or non-processed versions, so that the evaluation results can be more reliable and fair. Three different ERs are attempted for each scheme in generating the database like (#1) 5 % (#2) 10 % (#3) 20 % of the maximum payload capacity prescribed by the techniques. The entire database contains $200 \times 4 \times 8 = 6400$ (No. of images) * (No. of varying payload sizes + 1 for Clean image set) * (No. of schemes evaluated) images on the whole.

4.2 Performance metrics

For measuring the performance of the proposed system, we use the following metrics. We present them in view of binary class problem which give two discrete outputs positive class and negative class. In binary classification, for a given classifier and instance, we have four possible outcomes.

True Positive (TP) – Positive instances correctly classified as positive outputs

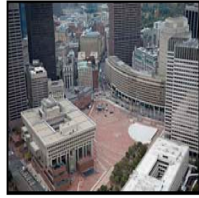
True Negative (TN) – Negative instances correctly classified as negative outputs

False Positive (FP) – Negative instances wrongly classified as positive outputs

False Negative (FN) – Positive instances wrongly classified as negative outputs

$$\text{True Positive Rate (TPR)} = \frac{\text{Positives correctly classified}}{\text{Total number of positives}} = \frac{TP}{TP + FN} \quad (23)$$

Aerial Category



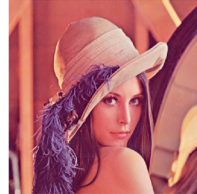
Lightning Variation Category



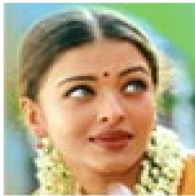
Underwater Category



Classical Category



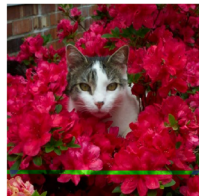
Face Category



Nature Category



Animal Category



Building Category



Fruit Category



Flower Category

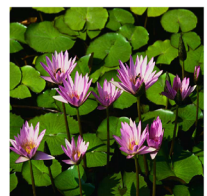


Fig. 5 Sample cover images used in performance evaluation

$$\text{False Positive Rate (FPR)} = \frac{\text{Negatives correctly classified}}{\text{Total number of negatives}} = \frac{FP}{FP + TN} \quad (24)$$

$$\text{Classification Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \quad (25)$$

$$\text{Precision} = \frac{\text{Number of True Positives}}{\text{Number of True Positives} + \text{False Positives}} \quad (26)$$

$$\text{Recall} = \text{Sensitivity} = \frac{\text{Postives correctly classified}}{\text{Total number of positives}} = \frac{TP}{TP + FN} \quad (27)$$

$$F\text{-Measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{(\text{Precision} + \text{Recall})} \quad (28)$$

4.3 Feature extraction and preprocessing

The features are calculated based on the algorithm in the Listing 1. We get an overall 60 statistical features that represent a file. Before proceeding to evaluate the performance of the classifier, discrimination capability of the proposed features is to be analyzed. The experiment involves breaking of different steganographic or watermarking strategies, which may adapt extremely different techniques for embedding ranging from LSB substitution to embedding inside the wavelet co-efficient.

Hence the feature set formed has to be normalized before feeding into the classifier for training to achieve a uniform semantics to the feature values. A set of normalized feature vectors as per the data smoothing function [44], $\tilde{f}_i = \frac{f_i - f_i^{\min}}{f_i^{\max} - f_i^{\min}}$, are calculated for each seed image to explore relative feature variations after and before it is modified. \tilde{f}_i , f_i^{\min} and f_i^{\max} represents the i^{th} feature vector value, the corresponding feature's minimum and maximum value respectively.

4.4 Evolutionary-SVM classifier

In the sequel, the model is incorporated in Java JGAP (<http://jgap.sourceforge.net/>) and the algorithm described in Listing 2 is implemented as per the method proposed. The ensemble classifier was trained and evaluated by using 4800 images out of the whole database, excluding those generated by using scheme #7 (employed as the test images to see how the proposed features behave when there is a mismatch between the operation domains). Here, two-thirds (3200) of images were randomly chosen as the training set and the others (1600 images) act as the validation set.

The GA parameters used were $w_1 = 0.4$, $w_2 = 0.4$ and $w_3 = 0.2$. The GA was run till 200 generations. There were 60 genes in the population; each gene representing a

feature as selected (1) or not (0). Two-point crossover with a rate of 0.8 and mutation with a rate of 0.02 were adapted. Radial Basis Function SVM model has been adopted with a 10-fold cross validation. Since the features are heterogeneous in scale, we performed the following operations in SVM parameter setting phase: We subtracted from each element in the input data the mean of the elements in that row, giving the row a mean of zero. We divided each element in the input data by the standard deviation of the elements in that row, giving the row a variance of one. Also the kernel matrix is normalized so as to get an enhanced performance. The convergence threshold was set as $1E-06$. Training halts when the objective function changes by less than this amount.

We have compared our results with some of the recent successful schemes Fig. 6. The classification and error rates obtained by using different values are listed in Table 1. Results show that the average classification rate does not change much (from 80 to 89 %). We are interested in analysing the detectability of proposed features and classifier against embedding schemes of different applications or principles. The system offers an appreciable range like 84.12 to 94.18 % sensitivity and 77.07 to 93.01 % for

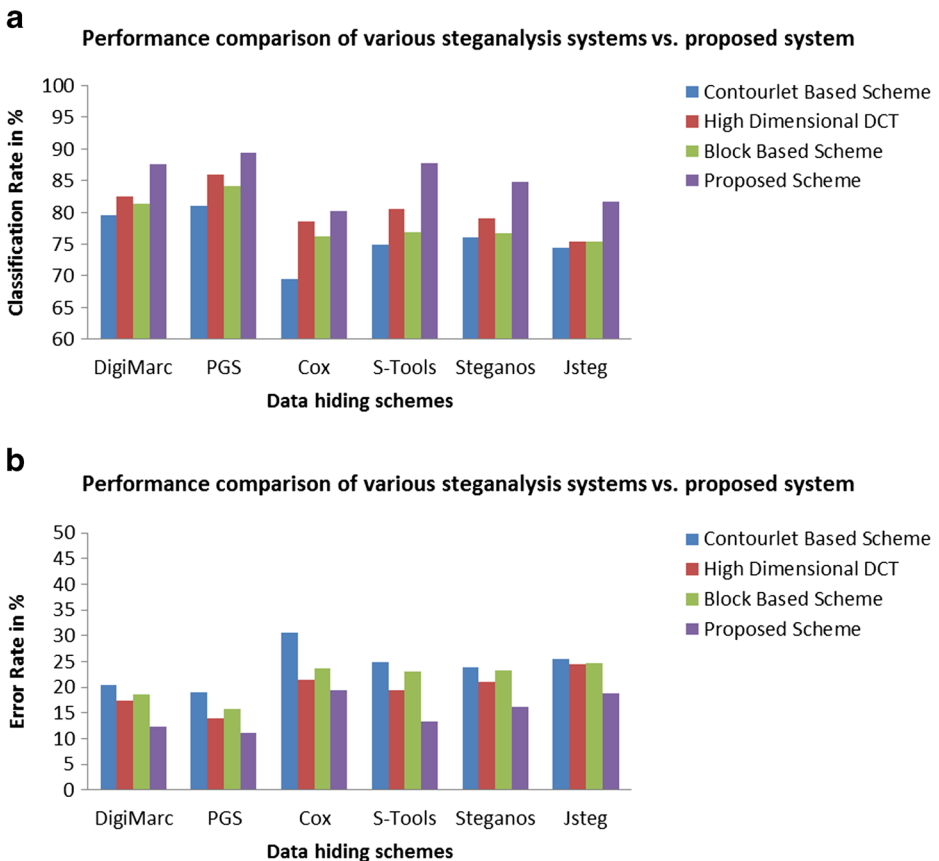


Fig. 6 Performance comparison curves depicting (a) Classification rate (b) Error rate of various steganalysis systems and the proposed system

Table 1 performance comparison of the classifiers

Scheme	True Positive (TP) in %	True Negative (TN) in %	Classification Rate (TP+TN)/2 in %									
			Block based Scheme [11]	Proposed Scheme	CBS [53]	High dimensional DCT [37]	Block based Scheme [11]	Proposed Scheme	CBS [53]	High dimensional DCT [37]		
DigiMarc	81.0	84.0	82.3	88.20	78.0	81.0	80.4	87.12	79.5	82.5	81.35	87.66
PGS	73.0	81.0	79.1	84.33	89.0	91.0	89.3	94.66	81.0	86.0	84.2	89.44
Cox	79.0	85.0	82.6	87.13	60.0	72.0	69.9	73.08	69.5	78.5	76.25	80.105
S-Tools	87.0	88.0	88.2	96.10	63.0	73.0	65.7	79.25	75.0	80.5	76.95	87.675
Steganos	86.0	82.0	85.3	90.21	66.0	76.0	68.3	79.50	76.0	79.0	76.8	84.855
Jsteg	78.0	79.0	78.1	85.27	71.0	72.0	72.6	78.04	74.5	75.5	75.35	81.655
Scheme	False Positive (FP) in %	False Negative (FN) in %	Error Rate (FP+FN)/2 in %									
	CBS [53]	High dimensional DCT [37]	Block based Scheme [11]	Proposed Scheme	CBS [53]	High dimensional DCT [37]	Block based Scheme [11]	Proposed Scheme	CBS [53]	High dimensional DCT [37]	Block based Scheme [11]	Proposed Scheme
DigiMarc	19.0	16.0	17.7	11.80	22.0	19.0	19.6	12.88	20.5	17.5	18.65	12.34
PGS	27.0	19.0	20.9	15.67	11.0	9.0	10.7	6.34	19.0	14.0	15.8	11.10
Cox	21.0	15.0	17.4	12.87	40.0	28.0	30.1	25.92	30.5	21.5	23.75	19.40
S-Tools	13.0	12.0	11.8	4.90	37.0	27.0	34.3	21.75	25.0	19.5	23.05	13.33
Steganos	14.0	18.0	14.7	10.79	34.0	24.0	31.7	21.50	24.0	21.0	23.2	16.15
Jsteg	22.0	21.0	21.9	14.73	29.0	28.0	27.4	22.96	25.5	24.5	24.65	18.85

specificity. Table 3 lists classification and error rates to see differentiation in performances between: 1) six targeted embedding schemes; 2) steganographic or watermarking applications; 3) spatial or DCT operation domain; and 4) types of processed non-stego images. We also analyzed the false negative rates for the original, smoothed, sharpened, and JPEG-compressed non-stego images. It is found that our system has a better performance in recognizing the plainness of JPEG-compressed images. The higher false negative rates for JPEG-compressed images is beneficial to real applications, since most images will be compressed in the JPEG form.

As for the detectability between different embedding schemes, we compare scheme #4 to #5 and scheme #1 to #3. Basically, embedding schemes #4 and #6 are similar in some aspects (both are in the spatial-domain, but for different applications), but the pixel change will be less for scheme #4 when embedding “0.” Accordingly, we got a higher true positive rate for scheme #4 than for scheme #5.

4.5 Influence of payload capacity on the Steganalyser’s performance

In this experiment, the images at various payload capacities were selected to see the influence on detectability. The ERs for the six embedding schemes were tried at 5, 10 and 20 % of the maximum hiding capacities in their proposed versions Fig. 7. The experimental results are listed in Table 4, which depicts that the average true positive rate still remains above 81.33 % for 20 %, 73.66 % for 10 % and 69.11 % for 5 % of maximum payload capacity. The results for steganographic schemes are more promising than for the watermarking schemes, as the steganographic schemes carry more hidden data than those of watermarking schemes, which makes the measured features more distinguishable for detection. The results reveal that clearly, our proposed content independent features and evolutionary-SVM classifier still yield reasonable results for stego images of less payload capacity.

4.6 Application on a completely new steganography scheme

In order to show that the system is dynamic i.e., adaptable to detect any new steganographic technique, the system was tested on scheme #7 which is based on the wavelet-domain techniques and scheme #8 – a steganography scheme that is resistive to blind steganalysis by embedding data in randomized locations in such a way that disables the self-calibration process. These schemes were chosen for testing to show the generalization ability of the proposed set of features. As expected, these systems were detected at a promising rate. We have tested with 12 different

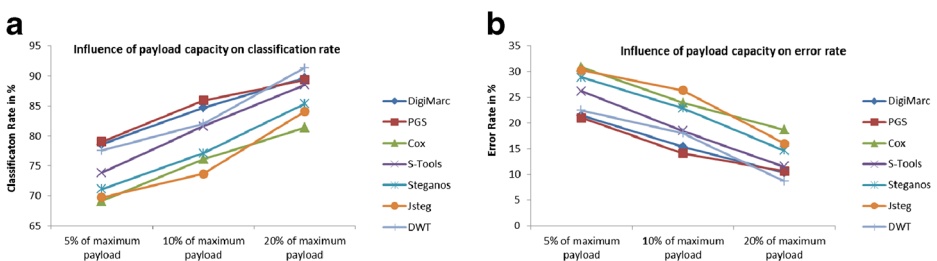


Fig. 7 Influence of payload capacity on performance of the steganalyzer (a) Classification rate (b) Error rate

Table 2 Parameter setting of the 12 variants of yass

Setting	Hiding Quality Factors	Decision Boundaries	Big Block Size	No. of iterations	bpac (bit per AC DCT coefficient)
YASS 1	65,70,75	3,7	9	0	0.110
YASS 2	75	–	9	0	0.051
YASS 3	75	–	9	1	0.187
YASS 4	65,70,75	2,5	9	0	0.118
YASS 5	50,55,60,65,70	3,7,12,17	9	0	0.159
YASS 6	75	–	10	0	0.031
YASS 7	65,70,75	3,7	10	0	0.078
YASS 8	75	–	10	1	0.138
YASS 9	65,70,75	3,7	9	2	0.237
YASS 10	75	–	10	2	0.159
YASS 11	75	–	11	1	0.114
YASS 12	65,70,75	3,7	11	0	0.077

configurations of Yet Another Steganography Scheme (YASS) including both the original and the extended versions. Table 2 shows the parameter settings of the 12 YASS variants. It was found that the true positive rate against scheme #7 is 88.23 %, and against scheme #8 is 82.66 % (average classification rates) as given in Table 3, 4 and 5. This proves that the identified features are sensitive to detect any new stego systems. The system is able to achieve a reasonably good true positive rate of 88.23 and 82.66 % because the data hiding process done in the DWT domain leaves

Table 3 average pd/nd rates for performance differentiation between different target schemes, different applications, different operation domains, and different types of nonstego images

Differentiation categories		Classification Rate
Schemes	#1	88.20 %
	#2	84.33 %
	#3	87.13 %
	#4	96.10 %
	#5	90.21 %
	#6	85.27 %
Applications	Watermarking	86.55 %
	Steganography	90.53 %
Operation domain	Spatial	90.21 %
	DCT	86.87 %
	DWT	88.23 %
	YASS	82.66 %
	Differentiation categories	Negative Detection rate
Type of processed non-stego images	Original	78.60 %
	JPEG-compressed	85.30 %
	Smoothed	79.50 %
	Sharpened	52.50 %

Table 4 classification and error rates for test sets at various embedding rates

Schemes	Classification rate in %			Error rate in %		
	5 % of maximum payload	10 % of maximum payload	20 % of maximum payload	5 % of maximum payload	10 % of maximum payload	20 % of maximum payload
#1	78.60	84.66	89.67	21.40	15.34	10.33
#2	79.00	85.90	89.33	21.00	14.10	10.67
#3	69.11	76.10	81.33	30.89	23.90	18.67
#4	73.83	81.55	88.50	26.17	18.45	11.50
#5	71.10	77.10	85.33	28.90	22.90	14.67
#6	69.70	73.66	84.00	30.30	26.34	16.00
#7	77.55	82.00	91.33	22.45	18.00	8.67

statistical artifacts in the higher order statistics of curvelet domain. Also, though the scheme #8 puts down the strength of self-calibration process that estimates the cover image from the stego image, the curvelet statistics capture the higher dimensional dependencies in the cover symbols. Table 6 compares the performance of the various classifiers used as steganalysers over the proposed features. Due to the presence of evolutionary algorithm component and the classifier design being a wrapper model, the training time is observed to be relatively higher than the other existing systems. However, the testing is with fewer relevant features, the testing time of the proposed model is lesser than the other classifiers. In a real-time scenario, faster detections of stego image flows in a network helps the security team to take measures faster before the harm is caused. Cover memory information which leaks out the important clue in steganalysis procedure is thus being incorporated into the feature vector. Further the proposed machine learning algorithm is powerful enough and hence when trained with thousands of images becomes capable of detecting even the slightest statistical

Table 5 steganalysis of yass using proposed system

Schemes	bpac	Classification Rate in %	Error Rate in %
YASS 1	0.110	83.00	17.00
YASS 2	0.051	78.00	22.00
YASS 3	0.187	85.00	15.00
YASS 4	0.118	73.60	26.40
YASS 5	0.159	86.30	13.70
YASS 6	0.031	82.60	17.40
YASS 7	0.078	85.40	14.60
YASS 8	0.138	80.50	19.50
YASS 9	0.237	82.20	17.80
YASS 10	0.159	85.20	14.80
YASS 11	0.114	86.10	13.90
YASS 12	0.077	86.50	13.50

Table 6 performance comparison of various classifiers as steganalyser

Schemes	Criteria	Decision Tree J48 Classifier	Naïve Bayes Classifier	SVM Classifier	Proposed Evolutionary SVM Classifier
DigiMarc	Accuracy %	84.00	82.65	84.66	87.66
	TP Rate	85.58	84.07	86.10	88.2
	TN Rate	82.16	81.23	83.21	87.12
	Precision	83.00	81.75	83.68	87.25
	Recall	86.00	84.07	86.10	88.2
	F-Measure	84.00	82.89	84.87	87.72
	Training Time (in sec)	192.34	188.31	244.41	258.3
	Testing Time (in sec)	1.25	1.21	1.33	1.16
PGS	Accuracy %	85.00	84.50	86.39	89.49
	TP Rate	80.53	79.51	81.41	84.33
	TN Rate	90.22	89.49	91.36	94.66
	Precision	89.00	88.00	90.41	94.04
	Recall	81.00	80.00	81.41	84.33
	F-Measure	85.00	84.00	85.67	88.92
	Training Time (in sec)	184.45	176.11	202.30	244.6
	Testing Time (in sec)	0.61	0.54	0.58	0.61
Cox	Accuracy %	84.00	74.00	75.60	80.10
	TP Rate	85.58	79.09	81.11	87.13
	TN Rate	82.16	68.20	70.08	73.08
	Precision	83.00	71.00	73.05	76.39
	Recall	86.00	79.00	81.11	87.13
	F-Measure	84.00	75.00	76.87	81.41
	Training Time (in sec)	186.23	171.22	190.01	200.03
	Testing Time (in sec)	0.42	0.39	0.44	0.38
S-Tools	Accuracy %	85.00	80.00	82.24	87.67
	TP Rate	80.53	88.97	90.31	96.10
	TN Rate	90.22	71.83	70.08	79.25
	Precision	89.00	76.00	77.75	82.24
	Recall	81.00	89.00	90.31	96.10
	F-Measure	85.00	82.00	83.56	88.63
	Training Time (in sec)	93.43	89.55	100.12	111.67
	Testing Time (in sec)	0.23	0.19	0.21	0.23
Steganos	Accuracy %	75.00	78.00	80.67	84.85
	TP Rate	80.99	85.21	87.66	90.21
	TN Rate	69.61	69.84	73.67	79.5
	Precision	73.00	74.00	77.75	81.48
	Recall	81.00	85.00	87.66	90.21
	F-Measure	77.00	79.00	83.56	85.62
	Training Time (in sec)	464.44	428.23	462.56	511.63
	Testing Time (in sec)	1.46	1.02	1.21	1.44
Jsteg	Accuracy %	81.00	74.00	76.46	81.65
	TP Rate	89.71	77.67	80.47	85.27
	TN Rate	72.73	70.04	72.44	78.04

Table 6 (continued)

Schemes	Criteria	Decision Tree J48 Classifier	Naïve Bayes Classifier	SVM Classifier	Proposed Evolutionary SVM Classifier
YASS	Precision	77.00	72.00	74.49	79.52
	Recall	90.00	78.00	80.47	85.27
	F-Measure	83.00	75.00	82.29	82.29
	Training Time (in sec)	308.45	298.66	303.04	365.4
	Testing Time (in sec)	0.57	0.48	0.56	0.49
	Accuracy %	79.00	74.00	77.38	82.67
	TP Rate	86.56	78.54	82.14	88.14
	TN Rate	70.98	68.81	72.61	77.21
	Precision	75.00	72.00	74.99	79.45
	Recall	87.00	79.00	82.14	88.14
	F-Measure	80.00	75.00	78.40	83.57
	Training Time (in sec)	703.12	698.56	711.24	724.23
	Testing Time (in sec)	2.04	1.97	2.00	2.02

variation. The system looks for these changes and thus is competent of capturing these differences and classifying the images as stego-bearing or not. This shows that the proposed features are competent enough to detect any new type of stego embedding schemes – irrespective of the logic they use to embed. This is capable of

Table 7 summarization of previous works and our proposed system

Steganalytic Systems	CBS [53]	High dimensional DCT [37]	Block based Scheme [11]	Proposed System
Number of features	64	15,700	274	60
Domains of Feature Extraction	Contourlet	DCT	Spatial – Block based	Spatial DCT DWT
Training/Classifier	Yes/Non-linear SVM	Yes/Bayesian Ensemble	Yes/Bayesian Ensemble	Yes/Evolutionary-SVM
Targeted embedding scheme	Arbitrary	Arbitrary	Model Based Steganography, Perturbed Quantization	Arbitrary
Number of test schemes	3	2	2	8
Payload of stego images	4000–15,000 bits	>0.03bpac	0.05 bpac	>0.01 bpp
Size of training database	600	2500	2264	6400
Number of test images	400	2500	565	3000
Average classification rate	75.91 %	80.33 %	78.48 %	85.23 %
Average error rate	24.08 %	19.67 %	21.51 %	14.76 %
Side information constraint for classifier	No	No	Block size and Block no.	No

capturing the cover statistics and also the disturbed statistics or the distortion perpetually introduced by the steganography scheme (Table 7).

5 Discussion and conclusion

Steganography is a dynamic tool with a long history and the capability to adapt to new levels of technology. As the steganographic tools become more advanced, the steganalyst and the tools they use must also advance. Like any tool, steganography (and steganalysis) is neither inherently good nor evil; it is the manner in which it is used which will determine whether it is a benefit or a detriment to the society. This paper presents a rationale for a blind image steganalytic model based on higher order statistics computed from curvelet coefficients. The feasibility of the proposed system is proved by systematic experiments. In our experiments, a database composed of processed plain images and stego images generated by using seven embedding schemes was utilized to evaluate the performance of our proposed features and classifier. Table 5 summarizes and compares characteristics of our proposed method with those of several other previous works in literature. In the table, not-reported (NRP) represents null information provided by the original work. The major findings of this work can be summarized as:

Higher order statistics computed from curvelet domain possess significant discriminatory power and proved to be useful, especially for steganographic data embedding, where the incurred distortions are much less pronounced than in watermarking.

A nonlinear classifier SVM that is easy to adapt to non-separable classes is adopted in our system. The classifier is constructed so as to minimize the false error rates. The GA component of the model incorporates this knowledge into the system. This has proved to be effective.

The average classification rate (85 %, including the true positive and true negative rates) for our proposed system is superior to many systems (Table 5) in blind steganalysis research. The future directions in this work can be concentrating more on the other statistics from curvelet domain like higher order moments and applying this system to videos and compressed images. The performance of the system can also be improved by using appropriate fusion techniques.

Acknowledgments This paper is based upon work supported by the All India Council for Technical Education - Research Promotion Scheme under Grant No. 20/AICTE/RIFD/RPS(POLICY-ID)65/2012-13.

References

1. AvciBAS I, Memon N, Sankur B (2003) Steganalysis using image quality metrics. *IEEE Trans Image Process* 12(2):221–229
2. AvciBAS I, Memon N, kharrazi M, Sankur B (2005) Image steganalysis with binary similarity measures. *J EURASIP, Appl Signal Process*, Hindawi
3. Baker JE (1985) Adaptive selection methods for genetic algorithms. In *Proc. 1st Int'l Conf. On Genetic Algorithms*. Pg 101–111

4. Bánoci V, Broda M, Bugár G, Levický D (2014) Universal image steganalytic method. *Radio Eng* 23(4): 1213–1220
5. Bender W, Gruhl D, Morimot N, Lu A (1996) Techniques for data hiding. *IBM Syst J* 35(3/4):313–336
6. Brown A. S-tools version 4.0. [Online]. Available: <http://members.tripod.com/steganography/stego/s-tools4.html>
7. Cachin C (1998) An information theoretic model for steganography, *Lecture Notes in Computer Science: 2nd Int'l Workshop on Information Hiding* 1525, pp. 306–318
8. Candes EJ, Donoho DL (1999) Curvelets - a surprisingly effective nonadaptive representation for objects with edges," In: Cohen A, Rabut C, Schumaker LL (eds) *Curve and surface fitting: Saint-Malo*
9. Chang C-C, Lin C-J: LIBSVM: a Library for Support Vector Machines, 2001, Available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
10. Cheng Q, Huang TS (2001) An additive approach to transform-domain information hiding and optimum detection structure. *IEEE Trans Multimedia* 3(3):273–284
11. Cho S, Cha B-H, Gawecki m, Jay Kuo C-C (2013) Block-based image steganalysis: algorithm and performance evaluation. *J Vis Commun Image Represent* 24(7):846–856
12. Cox JJ, Kilian J, Leighton FT, Shamoon T (1997) Secure spread spectrum watermarking for multimedia. *IEEE Trans Image Process* 6(12):1673–1687
13. Dai M, Liu Y, Lin J (2008) Steganalysis based on feature reducts of rough set by using genetic algorithm, *Proc. World Congress on Intelligent Control and Automation*
14. Duda RO, Hart PE, Stork DG (2001) *Pattern classification*. Wiley, New York
15. Farid H (2002) Detecting hidden messages using higher-order statistical models. In: *Image Processing, 2002. International Conference on*, Rochester, NY, USA 905–908
16. Fridrich J (2000) Miroslav Goljan and Dorin Hoge, *Attacking the OutGuess*, *Proc. ACM Intl. Conf. Information security*
17. Fridrich J (2004) Feature-Based Steganalysis for JPEG Images and its Implications for Future Design of Steganographic Schemes, *Proc. Int. Workshop. Information hiding*, vol.3200, *Lecture. Computer science*
18. Fridrich J, Goljan M (2002) Practical steganalysis of digital images-state of the art. *Proc SPIE* 4675:1–13
19. Fridrich J, Goljan M, Hoge D (2003) New methodology for breaking steganographic techniques for JPEGs, *Proc SPIE Electronic Imaging*, Santa Clara, CA, pp. 143–155
20. Fu J-W, Qi Y-C, Yuan J-S (2007) Wavelet domain audio steganalysis based on statistical moments and PCA, *Proc. IEEE Intl. Conf. Wavelet Analysis and Pattern recognition*
21. Geetha S, Sivatha Sindhu SS, Kamaraj N (2008) Steganalysis of LSB Embedded Images based on Adaptive Threshold Close Color Pair Signature, in *Sixth IEEE Indian Conference on Computer Vision, Graphics and Image Processing ICVGIP 2008*
22. Goljan, Fridrich J (2015) CFA-aware features for steganalysis of color images, *Proc. SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics XVII* San Francisco, CA
23. Gonzalez FP, Balado F, Martin JRH (2003) Performance analysis of existing and new methods for data hiding with known-host information in additive channels. *IEEE Trans Signal Process* 51(4): 960–980
24. Gu B, Sheng VS, Tay KY, Romano W, Li S (2015) Incremental support vector learning for ordinal regression. *IEEE Trans Neural Networks Learn Syst* 27(7):1404–1416
25. Harmsen JJ (2003) Steganalysis of additive noise modelable information hiding. Master's thesis, Rensselaer Polytechnic Institute, Troy, New York, USA
26. Holland JH (1975) *Adaptation in natural and artificial systems*. University of Michigan Press (reprinted in 1992 by MIT Press, Cambridge, MA)
27. Holotyak T, Fridrich J, Voloshynovskiy S (2005) Blind statistical steganalysis of additive steganography using wavelet higher order statistics. In: *Lecture Notes in Computer Science*, Springer Berlin, Heidelberg, pp. 273–274
28. Huang J, Shi YQ (1998) Adaptive image watermarking scheme based on visual masking. *Electron Lett* 34(8):748–750
29. Huo X (1999) Sparse image representation via combined transforms. PhD thesis, Stanford Univesity
30. Images. [Online]. Available: http://www.cl.cam.ac.uk/~fapp2/watermarking/benchmark/image_database.html
31. Katzenbeisser S, Petitcolas FAP (2000) *Information hiding techniques for steganography and digital watermarking*. Artech House, Norwood
32. Kaushal S, Anindya S, Manjunath BS (2007) YASS: Yet another steganographic scheme that resists blind steganalysis, *9th International Workshop on Information Hiding*, Saint Malo, Brittany, France, Jun
33. Kim Y-S, Kwon O-H, Park R-H (1999) Wavelet based watermarking method for digital images using the human visual system. *Electron Lett* 35(6):466–468
34. Korejwa J. Jsteg shell 2.0. [Online]. Available: <http://www.tiac.net/users/korejwa/steg.htm>

35. Kutter M, Jordan F JK-PGS (Pretty Good Signature). [Online]. Available: http://tswwww.epfl.ch/~kutter/watermarking/JK_PGS.html
36. Lee YK, Chen LH (2000) High capacity image steganographic model. *Proc Inst Elect Eng, Vis Image Signal Process* 147(3):288–294
37. Li F, Zhang X, Chen B, Feng G (2013) JPEG steganalysis with high-dimensional features and Bayesian ensemble classifier. *IEEE Signal Process Lett* 20(3):233–236
38. Lie W-N, Chang L-C (1999) Data hiding in images with adaptive numbers of least significant bits based on human visual system, in *Proc. IEEE Int. Conf. Image Processing*, pp. 286–290
39. Lie W-N, Lin G-S (2005) A feature-based classification technique for blind image steganalysis. *IEEE Trans Multimedia* 7(6):1007–1020
40. Lie W-N, Lin G-S, Wu C-L, Wang T-C (2000) Robust image watermarking on the DCT domain. *Proc IEEE Int Symp Circ System I*:228–231
41. Lyu S, Farid H (2006) Steganalysis using Higher-Order Image statistics. *Proc IEEE Trans Inf Forensic Secur*, Vol.1, no.1
42. Manikopoulos C, Shi Y-Q, Song S, Zhang Z, Ni Z, Zou D (2002) Detection of block DCT-based steganography in gray-scale images. in *Proc. 5th IEEE Workshop on Multimedia Signal Processing*, pp. 355–358
43. Marvel LM, Boncelet CG Jr, Retter CT (1999) Spread spectrum image steganography. *IEEE Trans Image Process* 8(8):1075–1083
44. Min F (2007) A novel intrusion detection method based on combining ensemble learning with induction-Enhanced Particle Swarm Algorithm *IEEE Third International Conference on Natural Computation (ICNC)*
45. Nikolaidis N, Pitas I (1998) Robust image watermarking in the spatial domain. *Signal Process* 66:385–403
46. Ogihara T, Nakamura D, Yokoya N (1996) Data embedding into pictorial with less distortion using discrete cosine transform. In: *Proc. ICPR'96*, pp. 675–679
47. Olshausen BA, Field DJ (1996) Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* 381:607–609
48. Peticolas AP, Anderson RJ, Kuhn MG (1999) Information hiding—A survey. *Proc IEEE* 87(7):1062–1078
49. Pevny T, Fridrich J (2008) Multiclass detector of current steganographic methods for JPEG format. *IEEE Trans Info Forensic Secur* 3(4):635–650
50. PictureMarc, Embed Watermark, v 1.00.45, Digimarc Corp Available: <http://avcibas.home.uludag.edu.tr/mmsp.pdf>
51. Podilchuk CI, Wenjun Z (1998) Image-adaptive watermarking using visual models. *IEEE J Select Areas Commun* 16(4):525–539
52. Ru X-M, Zhang H-J, Huang X (2005) Steganalysis of audio: Attacking the steghide. *Proc IEEE, Int Conf Mach Learn Cyberm* 7:3937–3942
53. Sajedi H, Jamzad M (2010) CBS: contourlet-based steganalysis method. *J Signal Process Syst* 61(1):367–373
54. Savoldi A, Gubian P (2007) Blind multi-class steganalysis system using wavelet statistics. In: *IEEE International Conference on Intelligent Information Hiding and Multimedia Signal Processing IHMSP'07*, IEEE Computer Society, pp. 93–96
55. Shaohui L, Hongxun Y, Wen G (2003) Neural Network based Steganalysis in Still images. *Proc Int'l Conf Multimedia Expo, ICME* 2:509–512
56. Shi YQ, Xuan G, Yang C, Gao J, Zhang Z, Chai P, Zou D, Chen C, Chen W (2005) Effective steganalysis based on statistical moments of wavelet characteristic function. In: *IEEE International Conference on Information Technology: Coding and Computing, ITCC&newapos;05*, IEEE Computer Society, pp. 768–773
57. Starck J, Candes EJ, Donoho DL (2001) The curvelet transform for image denoising. *IEEE Trans Image Process* 11(6):670–684
58. Steganos II Security Suite.. [Online]. Available: <http://www.steganos.com/english/steganos/download.htm>
59. Xia Z, Wang X, Sun X, Liu Q, Xiong N (2014) Steganalysis of LSB matching using differences between nonadjacent pixels”, *Multimedia Tools and Applications*, Springer Verlag, pp. 1–16. doi: 10.1007/s11042-014-2381-8
60. Xia Z, Wang X, Sun X, Wang B (2014) Steganalysis of least significant bit matching using multi-order differences. *Secur Commun Networks* 7(8):1283–1291
61. Xu B, Zhang Z, Wang J, Liu X (2007) Improved BSS based Schemes for Active steganalysis. *Proc. ACIS Int. Conf. Software Engineering, Artificial Intelligence, Networking and parallel distributed computing*



Dr. S. Muthuramalingam Received the B.E., degree in Computer Science and Engineering in 1997 and M.E., degree in Digital communication & Network Engineering in the year 2002 respectively, from the Bharathidasan University and Madurai Kamaraj University. He obtained his Ph.D. Degree from Anna University in 2012. He has rich teaching and research experience of 15+ years. He has published more than 20 papers in reputed International Conferences and refereed Journals. His research interests include computer networks, wireless networks and mobile computing.



Dr. N. Karthikeyan has received his B.E. degree from Syed Ammal Engineering College, M.E. degree from PSNA College of Engg. & Technology and Ph.D. from Anna University. He is heading the Computer Science and Engineering Department in Syed Ammal Engineering College. He is an IEEE and ACM member. He has organised many funded workshops, conferences and seminars. He has received grants for Faculty Development Programs from AICTE, UGC etc. His research interests include Image Security, Cloud computing, Medical Informatics. He has published many papers in IEEE conferences and journals. He has served as Organising Chair, Session Chair in IEEE conferences.



Dr. S. Geetha Received the B.E., and M.E., degrees in Computer Science and Engineering in 2000 and 2004, respectively, from the Madurai Kamaraj University and Anna University of Chennai, India. She obtained her Ph.D. Degree from Anna University in 2011. She has rich teaching and research experience of 15+ years. She has published more than 60 papers in reputed International Conferences and refereed Journals. She joins the review committee for IEEE Transactions on Information Forensics and Security and IEEE Transactions on Image Processing. Her research interests include steganography, steganalysis, multimedia security, intrusion detection systems, machine learning paradigms and information forensics. She is a recipient of University Rank and Academic Topper Award in B.E. and M.E. in 2000 and 2004 respectively. She is also the proud recipient of ASDF Best Academic Researcher Award 2013.



Dr. Siva S. Sivatha Sindhu Received her B.E and M.E degrees in Computer Science and Engineering from M.S. University and Anna University in 2002 and 2004 respectively. She obtained her Ph.D. Degree from Anna University 2011. Her research interests include Information security, Intrusion Detection Systems and soft computing approaches. She has published many papers in refereed IEEE conferences and reputed journals on areas like intrusion detection systems, steganography techniques and steganalysis. She has authored a book on Intrusion Detection Systems using Machine Learning Techniques. She had developed many security tools and services that are applied in e-commerce and online community.