2005

An Artificial Neural Network for Wavelet Steganalysis

Jennifer Davidson
Iowa State University

Clifford Bergman
Iowa State University, cbergman@iastate.edu

Eric Bartlett
Iowa State University

An Artificial Neural Network for Wavelet Steganalysis

Abstract

Hiding messages in image data, called steganography, is used for both legal and illicit purposes. The detection of hidden messages in image data stored on websites and computers, called steganalysis, is of prime importance to cyber forensics personnel. Automating the detection of hidden messages is a requirement, since the shear amount of image data stored on computers or websites makes it impossible for a person to investigate each image separately. This paper describes research on a prototype software system that automatically classifies an image as having hidden information or not, using a sophisticated artificial neural network (ANN) system. An ANN software package, the ISU ACL NetWorks Toolkit, is trained on a selection of image features that distinguish between stego and nonstego images. The novelty of this ANN is that it is a blind classifier that gives more accurate results than previous systems. It can detect messages hidden using a variety of different types of embedding algorithms. A Graphical User Interface (GUI) combines the ANN, feature selection, and embedding algorithms into a prototype software package that is not currently available to the cyber forensics community.

Keywords

steganalysis, blind steganalysis, steganography, artificial neural network, data hiding, ACL Toolkit

Disciplines

Electrical and Computer Engineering | Information Security | Mathematics | Systems and Communications

Comments

This article is from *Proceedings of SPIE--The International Society for Optical Engineering,* 5916, Mathematical Methods in Pattern and Image Analysis: 1-10. Posted with permission.

Proceedings of SPIE - The International Society for Optical Engineering, vol. 5916, Mathematical Methods in Pattern and Image Analysis, 2005, pp. 1-10.

An artificial neural network for wavelet steganalysis

Jennifer Davidson[1a], Clifford Bergman[2a], Eric Bartlett[3b]
[a]Department of Mathematics, [b]Department of Electrical and Computer Engineering
Iowa State University, Ames, Iowa, 50011

ABSTRACT

Hiding messages in image data, called *steganography,* is used for both legal and illicit purposes. The detection of hidden messages in image data stored on websites and computers, called *steganalysis,* is of prime importance to cyber forensics personnel. Automating the detection of hidden messages is a requirement, since the shear amount of image data stored on computers or websites makes it impossible for a person to investigate each image separately. This paper describes research on a prototype software system that automatically classifies an image as having hidden information or not, using a sophisticated artificial neural network (ANN) system. An ANN software package, the ISU ACL NetWorks Toolkit, is trained on a selection of image features that distinguish between stego and nonstego images. The novelty of this ANN is that it is a blind classifier that gives more accurate results than previous systems. It can detect messages hidden using a variety of different types of embedding algorithms. A Graphical User Interface (GUI) combines the ANN, feature selection, and embedding algorithms into a prototype software package that is not currently available to the cyber forensics community.

Keywords: steganalysis, blind steganalysis, steganography, artificial neural network, data hiding, ACL Toolkit

1. INTRODUCTION

During the past ten years there has been an intensified interest in the use of digital images and audio data for hiding information. Image and audio files are being used for covert channels, copyright protection, and authentication of data. *Steganography* is the study of techniques for hiding information in an innocuous carrier so that the existence of the message is concealed. The article in [23] gives an introduction to steganography. *Cryptography* [27] is different from steganography, as cryptography is concerned with obscuring the *content* of a message but not its *existence*. *Digital watermarking* [10] is another area of data hiding; it is concerned with issues related to content protection of the digital work itself, such as copyright control of audio and movie data, and intellectual property protection. When data is watermarked for content protection, it typically contains information about the data itself, such as the owner, copyright information, contact information, or ability to copy. Analyzing data to determine if information has been hidden in it is called *steganalysis*. Steganalysis techniques can be used to detect, extract, change or ultimately destroy the hidden information, and can be applied to suspect data for steganography, watermarking, or authentication purposes. The article by Fridrich in [13] gives a good survey on steganalysis techniques. This paper focuses mainly on steganalysis, although steganography and steganalysis are intimately related: knowing how data is embedded can lead to more effective detection techniques, which can in turn lead to less detectible embedding techniques.

Steganalysis is of growing interest to communities in law enforcement and counter-espionage. Hiding information in image and audio files, as well as other data types on digital computers, is increasingly common. Freeware for steganography is easy to download and easy to use (see www.stegoarchive.com, for example). Thus, tools for determining whether a file contains any hidden information are important for cyber forensics personnel. Law enforcement agencies at the local and national levels need good software programs that do a reliable job of identifying

[1] davidson@iastate.edu; phone (515) 294-2941
[2] cbergman@iastate.edu; phone (515) 294-8137
[3] eric@aclillc.com; phone (515) 294-1828

suspicious files on computers and websites. Sensitive information such as financial documents, files containing information concerning illegal business transactions or child pornography, as well as other unlawful activities, can be hidden in innocuous-looking image and audio files on a computer or website. Commercial tools exist to help find such hidden data, such as WetStone's StegoSuite (www.WetStone.com) or SoftwareShield's steganographic licensing algorithm (www.softwareshield.com). However, as commercial software vendors keep the specifics of their algorithms secret, and the cost is beyond the means of typical academic users, it is difficult for academic researchers to compare their results with those of commercial products. Anecdotal stories from active law enforcement workers cite a false positive rate high enough to be a concern to users of some commercial software. A need for more accurate identification of stego images is apparent. The project described in this paper addresses this question.

The remainder of the paper is organized as follows. Section 2 discusses steganalysis and previous related work. Section 3 discusses the features we selected to represent the types of data we want to classify. Section 4 presents the pattern recognition system we used for classification. Section 5 discusses the data used for the experiment, and section 6 presents the results. Section 7 gives conclusions and discusses future research, and section 8 gives the references.

2. STEGANALYSIS AND PREVIOUS WORK

Models for describing the data embedding and extraction processes can be posed as a communications channel, with the cover image serving the role of the communications channel through which the data is transmitted, and the message to be embedded playing the role of as the data stream to be sent. Thus, the cover image can be viewed as *noise*; simple initial efforts model the image as realization of Gaussian noise. See Fig. 1. Under these assumptions, signal detection theoretic concepts as well as information theoretic concepts may be applied to model embedding algorithms, extraction algorithms, and message detection algorithms. Pattern recognition models may also be applied, and in this research we apply an *artificial neural network* for pattern classification of feature data extracted from the cover image and stego image data.

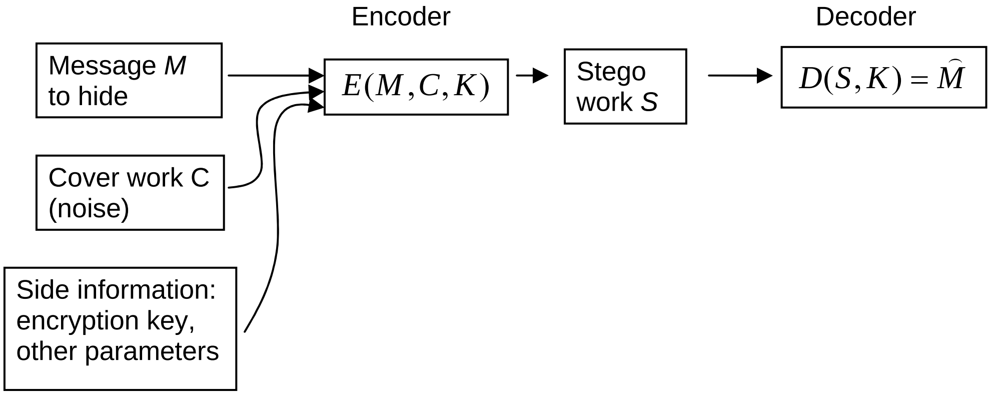

Figure 1. Communications model for steganography and steganalysis [20].

How does one determine if an image file has hidden information in it? First, the analyst determines if any *a priori* information is available, such as evidence of embedding software on the computer where the image was stored. Having a guess at what software was used to embed can help the analyst decide how to try to extract the information. Also, if there are multiple versions of a particular image file, one may be the original or *cover* image, and another may the image in which the information was hidden, the *stego* image. Using this and any other information, the analyst can run a detection algorithm, or extract the hidden data by running extraction software, or even write code to perform the process of extraction. If none of the *a priori* information is available, then we say the steganalyst is performing a *blind* analysis—that is, neither the embedding algorithm nor the cover image are known prior to analyzing the data. Blind steganalysis is most often performed on data that is attached to email, on websites, on flashdrives, on MP3 players, etc. The research presented in this paper is on blind steganalysis of image data using an artificial neural network.

Several basic approaches are used to detect hidden messages. Most of them manually identify a specific signature for a given embedding algorithm, based on the image data or features extracted from the image data, and then create a test, usually based on statistics, to identify whether or not an image contains hidden data. In addition, most techniques are applicable to data in one format—jpeg, gif, etc. A *signature* of an embedding algorithm is a distinctive characteristic of some feature or collection of features of an image that can be used to identify the existence of hidden data in the image. Signatures include statistical measures or properties that may give different results for an original image than a stego image (an image that has data hidden in it), such as the image histogram does for the jsteg embedding algorithm [17]. Color palettes in stego images can also be used as signatures, such as for S-Tools [16]. Our approach is to use the power of a nonlinear function approximator/ classifier, namely, an artificial neural network, to distinguish original images (no message hidden) from stegoimages (with message hidden). ANNs have been shown over the past decade or more to provide solutions to data mining-type problems that are not easily amenable to classical solutions [19].

In the literature, ANNs have been applied twice to steganalysis problems. In [24], Shaohui et al. describe an artificial neural network model designed for steganalysis applied to still images. The features they use for the input to their ANN are spectral measures of the discrete cosine transform (DCT) and the discrete Fourier transform (DFT), and four moments of the wavelet transform. The ANN they use is a three-layer, two-output, feed-forward network with simple backpropagation for training. They manually set the number of nodes for the one hidden layer and use a straight-forward, time-consuming backpropagation training algorithm. Their ANN detects only data embedded using the quantization index modulation method [9], and the authors do not give the data format they used. According to [13], the format of the data can be crucial to the success of the steganalysis technique. Their training data consisted of a total of 44 images, a very small number indeed. Thus, while their classification percentages ranged from 75% correct for images with no hidden data, and 85% correct for images with hidden data, it was a very limited experiment. In [5], Berg et al. demonstrate that three artificial intelligence techniques—decision tree, naïve Bayes classifier, and a simple backpropagation ANN—can perform in the same general range of correct classification of hidden messages as *stegdetect* [23] or even slightly better. For embedding algorithms, they use Jsteg-jpeg and GIFShuffle, a palette-shuffling technique. (These and other embedding algorithms can be found at www.stegoarchive.com.) Two different models are developed, not one as we propose, and they train on 150 images. While their best technique performs at 81% correct classification, to run their classifiers on the two different types of data or one of the two different embedding techniques, one must use different models. The ANN used in this project is a general purpose steganographic content detector that can be trained on a wide variety of embedding algorithms and different data types.

3. FEATURES

This section presents the features we selected to use for classification. We reviewed much of the literature on feature selection for steganography detection, and decided on the following feature values:

1. the first four moments of the wavelet transform subband coefficient values [11];
2. the first four moments of the errors in the linear prediction estimation of the wavelet transform subband coefficient values [11];
3. a selection of image quality metrics [1];
4. a selection of feature values based on distance between parameter values of calibrated images [12];
5. parameters estimated by fitting a generalized Gaussian distribution to wavelet subband coefficient values [25];
6. a feature based on the chi-square attack proposed by Westfeld [29], and extended in [18].

The feature extraction was implemented using Matlab 7.0.1 and Mathematica 5.1. Both platforms provide excellent prototyping capabilities, although execution time is not optimal. For the initial phase of the project, we selected a subset of these features to implement, namely set 1 above and set 3 above, for a total of 39 feature values. The features were coded and applied to two sets of data: original cover images with no hidden message (class 0), and stegoimages with hidden messages using the jsteg algorithm (class 1), message length approximately 50% to the total capacity.

The wavelet transform gives a decomposition of an image in terms of basis functions that give simultaneous information about frequency content and spatial position of the frequency. These properties have found successful applications in image compression, noise removal, image coding, and texture modeling. Such decompositions have statistical regularities that can be exploited. In additional, because the wavelet transform is hierarchical, statistical dependencies for natural image scenes exist across scale, orientation, and space. The set of features in 2 above should give additional information not available in 1. Due to time constraints, we have implemented only these 39 features in set 1 and will add the remaining features to present in future published works.

The features given in [11] have been used to identify images with stego content in previous works [24], [28]. Our work extends that in [11] by using an extended set feature values and producing a procedure that can be adapted to include features based on future unknown stego algorithms. While our initial results do not match those in [11], our ANN is more versatile and training on other data sets can easily be incorporated in. Our work extends that in [24] and [5] by giving more accurate results, as discussed in Section 6.

Following [1], several features were created based on *image quality metrics* (IQMs). The expectation is that the disruption to an image caused by low-pass filtering will be different for a cover image as compared to a stego-image (see Figure 2).

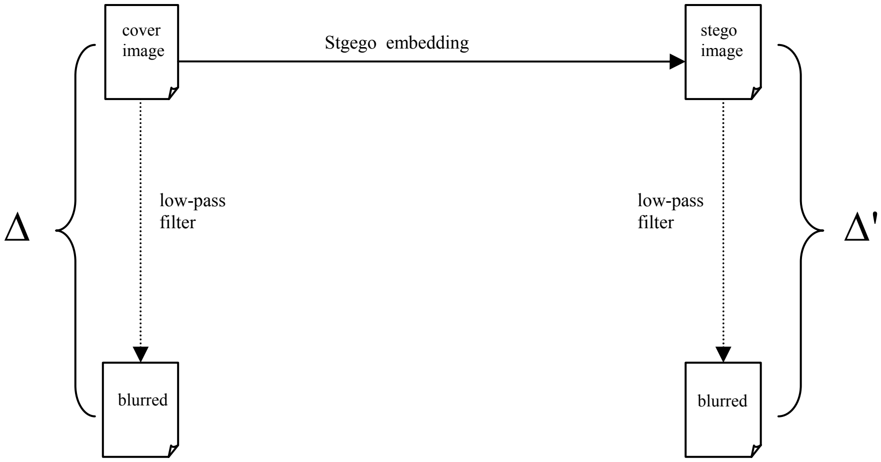

Figure 2: Comparing the distortion caused by blurring a cover image and a stego-image.

To be more precise, a Gaussian filter, H (see equation (1)), is applied to a candidate image, I, producing a blurred version of the image, H(I). An IQM, μ , is used to measure the difference between the image and its blurred counterpart, yielding a quantity $\Delta = \mu(I, H(I))$. By training the neural network on values of Δ obtained from both unmodified and stego images, we hope the neural net will later be able to distinguish between the two.

We selected three IQMs for this project, all taken from [1]. They are the median block spectral phase, the median weighted block spectral distortion and the normalized mean square HVS error. A description of these metrics is given in equations (2–4) below.

Here is a detailed description of the filter and the image quality metrics. We assume our image is a rectangular, single-band array $I_{i,j}$, $0 \le i \le m-1$, $0 \le j \le n-1$. The blurred image $H(I)$ is obtained by convolving I with the 3×3 matrix Kg where

$$g_{u,v} = \frac{2}{\pi}\exp\left(-2\left(u^2 + v^2\right)\right), \text{ for } -1 \le u,v \le 1 \text{ and } K = \left(\sum\nolimits_{u,v=-1}^{1}\left|g_{u,v}\right|^2\right)^{-1/2}. \qquad (1)$$

Let I and J be images of identical dimensions. The value of $\mu(I,J)$ where μ is one of the two block spectral metrics is computed as follows. I and J are each partitioned into blocks of size 32×32, enumerated as $I^{(k)}, J^{(k)}$, $k = 1,\ldots K$. For a block $I^{(k)}$, let $\hat{I}^{(k)}$ be its discrete Fourier transform. Then the median block spectral phase difference between I and J is defined as

$$\mu_1(I,J) = \underset{k}{\text{median}}\sqrt{\sum_{i,j=0}^{31}\left(\left|\arg\left(\hat{\mathrm{I}}_{\mathrm{i,j}}^{(\mathrm{k})}\right)\right| - \left|\arg\left(\hat{J}_{i,j}^{(k)}\right)\right|\right)^2}$$

where $\arg(z)$ denotes the phase angle of the complex number z. Similarly, the median weighted block spectral distortion is given as

$$\lambda \cdot \mu_2(I,J) + (1-\lambda)\cdot \mu_1(I,J). \qquad (2)$$

Here $\mu_2(I,J)$ is defined in the same manner as $\mu_1(I,J)$, but with the phase angle replaced with magnitude. The value of λ was chosen experimentally to be 0.000125.

Finally, the normalized mean square HVS metric is defined as

$$\mu_3(I,J) = \frac{\left\|U(I) - U(J)\right\|_F^2}{\left\|U(I)\right\|_F^2}$$

where $\|\bullet\|_F$ denotes the Frobenius norm, and $U(I) = D^{-1}\left(F\left(D(I)\right)\right)$. In this equation, $D(I)$ denotes the discrete cosine transform of I, and $F(J)$ is a band-pass filter applied to J, given (in polar coordinates) by

$$F(\rho) = \begin{cases} .05\exp\left(\rho^{0.554}\right) & \rho < 7, \\ \exp\left(-9\left[\left|\log\rho - \log 9\right|\right]^{2.3}\right) & \rho \ge 7. \end{cases}$$

See [1] or [22] for more details.

4. PATTERN RECOGNITION SYSTEM

This section describes the ANN software we used. The Adaptive Computing Laboratory (ACL) Toolkit (henceforth, Toolkit) is a comprehensive system developed at Iowa State University for advanced Artificial Neural Network (ANN) development and application, and is an open-source software package available at Iowa State University [4]. It can be used to design linear and nonlinear empirical models to solve complex applications involving pattern classification, process modeling, and forecasting. The Toolkit includes a data processing module with capabilities that enhance and speed ANN model development. The modules of the Toolkit include:

1. Data pre- and post-processing
2. Test set determination
3. Normalization
4. Input Ranking
5. Pattern Selector
6. ANN Training Engines
7. Dynamic Node Architecture
8. Network Recall
9. Sensitivity Analysis
10. Model Integration
11. Error Estimation
12. Help Module

The Toolkit is a sophisticated, advanced applications-oriented ANN software environment, which has been used in numerous application environments with excellent success [7], [2], [3]. The Toolkit allows the user to do extensive pre- and post-processing on the data, such as normalization of the data values, and linear, information theoretic, and nonlinear methods to determine the input importance ranking, and provides a full spectrum of analysis, manipulation, and filtering techniques for data preparation and quality assurance. It has methods for dealing with nonlinear, missing, and outlying data, to auto- and cross- correlations and power spectral densities. There are also capabilities for intelligent clustering and learning strategies.

We selected this software for our pattern recognition system because of its extensive and sophisticated proven capabilities at solving difficult data mining problems [3], [26]. The set of weight values and number of nodes for an ANN is called the ANN *architecture*. The Toolkit allows the ANN architecture to be simultaneously and automatically optimized using a dynamic node approach, an advanced technique not available in commercial ANN software. Thus, the user need only select the number of inputs and the number of layers, reducing the user's need for ANN expertise. The number of optimal nodes is determined by the ANN itself using information gleaned from the data. The weight values are optimized using a scaled conjugate gradient descent learning algorithm (SCGD). This learning algorithm is faster than the backpropagation learning algorithm by a factor of 1000 or more, and thus the ANN trained with SCDG can attempt bigger problems such as processing with image data with reasonable computer resources. Thus, with a better modeling capability than previously used ANNs, we have achieved initial results that outperform previous published results using ANNs [24], [5].

We use the features selected from the image data, as described in Section 3, as input to the ANN and classify the one ANN output as either having a hidden message (value one) or not (value zero). From the master set of features we selected 39 features, 36 of them the first four wavelet moments from the wavelet transform coefficient values, and the three IQM values. The Toolkit was able to find an optimal minimal feature set using three ranking schemes, consisting of the 16 features shown in Table 1. This reduced the number of inputs necessary and hence the complexity of the ANN.

Table 1. 16 feature values used for the ANN.

Feature Number	Feature Description
1-3	the mean value for the coefficients for the horizontal, vertical and diagonal bands, level 1
4	mean value for diagonal band, level 2
5	mean value for diagonal band, level 3
6	variance for diagonal band, level 1
7	variance for diagonal band, level 2
8	skewness value for vertical band, level 1
9	skewness for diagonal band, level 1
10	skewness for diagonal band, level 2
11-13	kurtosis for horizontal, vertical, and diagonal bands, level 1
14	kurtosis for diagonal band, level 2
15-16	IQM measures 1 and 2

One advanced capability that the Toolkit offers is that additional retraining on new data, containing, say, features from new embedding algorithms, can be performed without losing the classification on the old data set. This means that training on additional new data can be performed in much less time than it took to do the first training. Thus, new embedding algorithms whose signatures can be captured using the same feature set from new image data, can be included into the existing ANN without losing the ability to correctly classify old data. We expect to use this capability as we continue this research into detection of hidden data that use different embedding algorithms or different image formats.

5. DATA

The data we used for training and testing the ANN was generously provided by the Watermarking Evaluation Testbed (http://www.datahiding.org), courtesy of Dr. Edward Delp, Purdue University. There were a total of 1300 images, which had been cleared of copyright issues. The image data was originally formatted in PNG format, which we change to grayscale (BMP) if the embedding algorithm requires grayscale images, as jsteg does. The image scenes in these images include natural scenes (taken by a camera, for example), digitized photographs, maps, computer graphics images ("clip art" type), and other various types of images. We excluded six images due to their overly large size. The data ranges in size from 3 KB to 13 MB (excluding the largest files). Of the remaining images, we set aside 300 for future testing scenarios. Thus, we had 1000 images to use for training and testing the current ANN. We used Matlab and Mathematica for coding the feature values, and a Dell Precision 670 Windows server machine to run most of the feature extraction algorithms.

6. RESULTS

This section discusses the process of preprocessing the data and applying the artificial neural network to the feature values. As discussed in Section 4, the ANN will generally perform better if the user preprocesses the data. In our case, the data was normalized and was ranked by three input importance ranking algorithms. The feature values that were common to all three rankings were the 16 features as described in Table 1. The ANN was trained on 900 patterns of class 0 (no hidden message), and 900 patterns of class 1 (hidden message using jsteg, with a message length approximately 50% of the total capacity of the data). Then, feature values from 221 unseen images were calculated and put through the ANN. The graph shown in Figure 3 describes the actual ANN output values for these 221 images (wiggly line). The desired output for images numbered 1-98 was zero, and the desired output for images numbered 99-221 was one. The ANN misclassified 9 cover images as stego (false positives), and 10 stego images as cover (false negatives). Table 2 gives the rates of classification. These rates are very good when compared to the other ANN results (see above). The percent correctly classified is 91.5%, and the percent misclassified is 8.5%. We expect to see improvements as we add additional features that most likely will have other information in them.

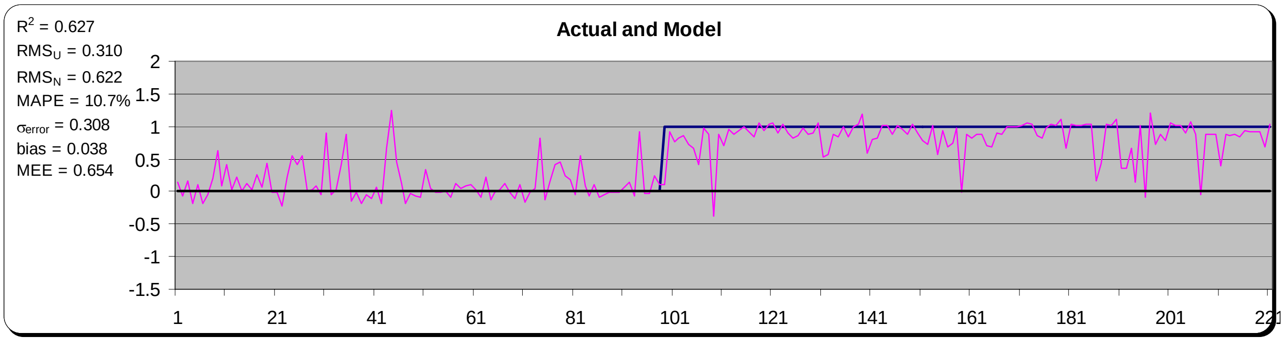

Figure 3. Graph of actual output values for features from 221 unseen images (wiggly line), and graph of desired model output for those 221 images (0 for images 1-98, 1 for images 99-221).

Table 2. Classification numbers for the ANN.

Classification Numbers		
	Message present	Message not present
Test output = 1	# True Positives = 113	# False Positive = 9
Test output = 0	# False Negatives = 10	# True Negatives = 89

Classification Rates		
	TPR = 92%	FPR = 10%
	FNR = 8%	TNR = 91%

7. CONCLUSIONS

We have presented a pattern recognition system using a sophisticated artificial neural network software package to classify steganography image data. Preliminary results are very good, with misclassification rate lower than other ANNs. We expect to get still better performance as more features with different information are incorporated.

8. ACKNOWLEGEMENTS

This work was partially funded by the National Institute of Justice, through the Midwest Forensics Resource Center at Ames Laboratory under interagency agreement number 2002–LP–R–083. The Ames Laboratory is operated for the US Department of Energy by Iowa State University, under contract No. W–7405–Eng–82.

9. REFERENCES

[1] I. Avcibaş, N. Memon and B. Sankur, "Steganalysis Using Image Quality Metrics," IEEE Trans. on Image Processing, vol. 12, no. 2, 221–229, 2003.

[2] E. B. Bartlett and R. G. Abboud, "Error estimates and model consolidation for time series data," Computational Intelligence for Financial Engineering, IEEE Neural Networks Council, pp. 174–177, New York City, March 2000.

[3] E. B. Bartlett and Adam Whitney, "On the use of various input subsets for stacked generalization," Intelligent Engineering Systems Through Artificial Neural Networks: vol. 9, Dagli, Akay, Chen, Fernandez, and Ghosh Editors, Artificial Neural Networks in Engineering, St. Louis, Missouri, American Society of Mechanical Engineers, pp. 111–116, November 1999.

[4] http://www.public.iastate.edu/~ebart/.

[5] G. Berg et al. "Searching for hidden messages: Automatic detection of steganography," Proc. 15th Innovative Applications of Artificial Intelligence Conf., Acapulco, Mexico, pp. 51–56, 2003.

[6] C. Cachin, "An information-theoretic model for steganography," Proc. Information Hiding: second international workshop, pp. 15–17, 1998.

[7] C. G. Carmichael and E. B. Bartlett "Stacking diverse models to achieve reliable error response distributions", International Journal of Smart Engineering Systems Design, vol. 4, pp. 55–63, April 2002.

[8] R. Chandramouli, "A mathematical framework for active steganalysis," ACM Multimedia Systems Journal, Special Issue on Multimedia Watermarking, Multimedia Systems vol. 9, pp. 303–311, 2003.

[9] B. Chen and G. W. Wornell, “Quantization index modulation: a class of provably good methods for digital watermarking and information embedding,” IEEE Trans. on Information Theory, vol. 47, pp.1423–1443, May 2001.

[10] I. Cox, M. Miller and J. Bloom, *Digital Watermarking,* Academic Press. 2002.

[11] H. Farid, “Detecting hidden messages using higher-order statistical models,” Proc. International Conference on Image Processing, vol. 2, pp. II-905–II-908, 2002.

[12] J. Fridrich, “Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes,” preprint, 2004.

[13] J. Fridrich and M. Goljan, “Practical Steganalysis—State of the Art,” Proc. SPIE Photonics Imaging 2002, Security and Watermarking of Multimedia Contents*,* vol. 4675, SPIE Press, pp. 1–13, 2002.

[14] J. Fridrich, M. Goljan and D. Hogea, “Steganalysis of JPEG images: breaking the F5 algorithm,” 5th Information Hiding Workshop, Noordwijkerhout, The Netherlands, pp. 310–323, 2002.

[15] J. Harmsen and W. Pearlman, “Capacity of Steganographic Channels,” ACM Multimedia and Security Workshop *MMSEC’05,* August 1–2, 2005, N.Y., New York, USA., to appear.

[16] N. Johnson and S. Jajodia, “Steganography: seeing the unseen”, IEEE Computer, pp. 26–34, February 1998.

[17] N. Johnson, Z. Duric, and S. Jajodia, *Information Hiding: Steganography and Watermarking: Attacks and Countermeasures*. Kluwer Academic, 2001.

[18] A. McAdams and T. McKay, “Evaluating the Chi-Square Attack for Steganography Using ROC Curves,” preprint, 2005.

[19] J. E. Meng et al., “Face recognition with radial basis function (RBF) neural networks,” IEEE Trans. on Neural Networks, vol. 13, pp. 697–710, May 2002.

[20] P. Moulin and M.K. Michak, “The parallel-Gaussian watermarking game”, IEEE Trans. on Information Theory, vol. 50 (2), pp. 272–289, February 2004.

[21] P. Moulin and Y. Wang, “New results on steganographic capacity,” Proc. CISS Conference, Princeton, NJ, Mar. 2004.

[22] N. Nill, “A Visual Model Weighted Cosine Transform for Image Compression and Quality Assessment,” IEEE Trans. on Communications, vol. com-33, no. 6, 551–557, June 1985.

[23] N. Provos, and P. Honeyman, “Hide and seek: an introduction to steganography,” IEEE Security & Privacy Magazine, Vol.1, Issue 3, pp. 32–44, 2003.

[24] L. Shaohui, Y. Hongxun and G. Wen, “Neural network based steganalysis in still images,” Proc. Int’l. Conf. on Multimedia and Expo, ICME 2003, vol. 2, pp.509–512, 2003.

[25] L. Shaohui, Y. Hongxun and G. Wen, “Steganalysis based on wavelet texture analysis and neural network,” Fifth World Congress on Intelligent Control and Automation (WCICA), vol. 5, pp. 4066–4069, June 2004.

[26] D. V. Sridhar, E. B. Bartlett, and R. C. Seagrave, “Information theoretic subset selection for neural network models of chemical processes,” Computers and Chemical Engineering, vol. 22, no. 4/5, pp. 613–626, 1998.

[27] D. Stinson, *Cryptography: Theory and Practice*, second edition, Chapman/CRC Press, 2002.

[28] R. Tzschoppe et al., “Steganographic system based on higher-order statistics,” Proc. EI SPIE Electronic Imaging, Santa Clara, pp. 156–166, 2003.

[29] A. Westfeld and A. Pfitzmann, “Attacks on Steganographic Systems,” 3rd International Workshop on Information Hiding, Lecture Notes in Computer Science, vol. 1768, pp.61–75, Springer-Verlag, 2000.

[30] M. Wu and B. Liu *Multimedia Data Hiding,* Springer, 2003.