Hybrid SOFM-MLP Neural Network for Steganalysis to Detect Stego-Contents in Corporate Emails

Dr. Anitha Thangasami¹, Dr. M.I. Thariq Hussan²

Associate Professor, Department of Informatics, Wollega University, Nekemte, Ethiopia¹.
Professor, Department of Information Technology, Guru Nanak Institutions Technical Campus, Hyderabad².
anithapt74@gmail.com¹, thariq.gnitc@gmail.com²

Abstract - Steganalysis is the art of detecting and often decoding hidden data within a given medium. The path of seeking the hidden data is treacherous and uncertain. Information theory and statistical analysis, the two major tools in steganalysis, reveal in clear terms the tremendous potential for hidden information in internet data. There is no room for hidden data within the medium as long as the set of data can be compressed to a smaller size.

Keywords – Steganalysis, LSB, DCT, SOFM, Neural Network, stego-content.

I. INTRODUCTION

Steganography and cryptography are cousins in the spycraft family. Cryptography scrambles a message so it cannot be understand whereas steganography hides the message so it cannot be seen. A message in ciphertext, for instance, might arouse suspicion on the part of the recipient while an invisible message created with steganographic methods will not.

1.1. E-MAIL SECURITY

The Internet, manifest as the World Wide Web and as e-mail, is revolutionary. It overshadows the personal computer, enticing more “regular folks” with the ability to browse, shop, and communicate than ever wanted to format a document or balance a checkbook. It has brought the issue of censorship versus protection to a new level of complexity. It is accused of turning people away from the real world of interpersonal relationships. It is even luring people away from their televisions.

1.1.1. Image files

An image is an array of numbers that represents light intensities at various points or pixels. These pixels make up the image’s raster data. A common image size is 640x480 pixels and 256 colors or 8 bits per pixel. Such an image could contain about 300 kilobytes of data.

Digital images are typically stored in either 24 bit or 8 bit files. A 24-bit image provides the most space for hiding information; however, it can be quite large (with the exception of JPEG images). All color variations for the pixels are derived from three primary colors: red, green, and blue. Each primary color is represented by 1 byte; 24-bit images use 3 bytes per pixel to represent a color value. These 3 bytes can be represented as hexadecimal, decimal, and binary values. The background color is represented by a six digit hexadecimal number in many web pages.

Digital images are typically stored in either 24-bit or 8-bit files. A 24-bit image provides the most space for hiding information; however, it can be quite large (with the exception of JPEG images). All color variations for the pixels are derived from three primary colors: red, green, and blue. Each primary color is represented by 1 byte; 24-bit images use 3 bytes per pixel to represent a color value. These 3 bytes can be represented as hexadecimal, decimal, and binary values. In many Web pages, the background color is represented by a six-digit hexadecimal number - actually three pairs representing red, green, and blue. A white background would have the value FFFFFF: 100 percent red (FF), 100 percent green (FF), and 100 percent blue (FF). Its decimal value is 255, 255, 255, and its binary value is 11111111, 11111111, 11111111, which are the three bytes making up white.

This definition of a white background is analogous to the color definition of a single pixel in an image. Pixel representation contributes to file size. For example, suppose we have a 24-bit image 1,024 pixels wide by 768 pixels high - a common resolution for high resolution graphics. Such an image has more than two million pixels, each having such a definition, which would produce a file exceeding 2 Mbytes. Because such 24-bit images are still relatively uncommon on the Internet, their size would attract attention during transmission. File compression would thus be beneficial, if not necessary, to transmit such a file.
1.1.2. Concealment in Digital Images

Information can be hidden many different ways in images. To hide information, straight message insertion may encode every bit of information in the image or selectively embed the message in “noisy” areas that draw less attention - those areas where there is a great deal of natural color variation. The message may also be scattered randomly throughout the image. Redundant pattern encoding “wallpapers” the cover image with the message.

1.2. MASKING AND FILTERING

Masking and filtering techniques, usually restricted to 24-bit and gray-scale images, hide information by marking an image, in a manner similar to paper watermarks. Watermarking techniques may be applied without fear of image destruction due to lossy compression because they are more integrated into the image. Visible watermarks are not steganography by definition. The difference is primarily one of intent. Traditional steganography conceals information; watermarks extend information and become an attribute of the cover image. Digital watermarks may include such information as copyright, ownership, or license. In steganography, the object of communication is the hidden message. In digital watermarks, the object of communication is the cover.

To create the watermarked image, we increased the luminance of the masked area by 15 percent. If we were to change the luminance by a smaller percentage, the mask would be undetected by the human eye. Now we can use the watermarked image to hide plaintext or encoded information. Masking is more robust than LSB insertion with respect to compression, cropping, and some image processing. Masking techniques embed information in significant areas so that the hidden message is more integral to the cover image than just hiding it in the “noise” level. This makes it more suitable than LSB with, for instance, lossy JPEG images.

Algorithms and transformations: LSB manipulation is a quick and easy way to hide information but is vulnerable to small changes resulting from image processing or lossy compression. Such compression is a key advantage that JPEG images have over other formats. High color quality images can be stored in relatively small files using JPEG compression methods; thus, JPEG images are becoming more abundant on the Internet. One steganography tool that integrates the compression algorithm for hiding information is Jpeg-Jsteg. Jpeg-Jsteg creates a JPEG stego-image from the input of a message to be hidden and a lossless cover image. According to the Independent JPEG Group, the JPEG software we tested has been modified for 1-bit steganography in JFIF output files, which are composed of lossy and non-lossy sections. The software combines the message and the cover images using the JPEG algorithm to create lossy JPEG stego-images.

JPEG images use the discrete cosine transform to achieve compression. DCT is a lossy compression transform because the cosine values cannot be calculated exactly, and repeated calculations introduce rounding errors into the final result. Variances between original data values and restored data values depend on the method used to calculate DCT.

In addition to DCT, images can be processed with fast Fourier transformation and wavelet transformation. Other image properties such as luminance can also be manipulated. Patchwork and similar techniques use redundant pattern encoding or spread spectrum methods to scatter hidden information throughout the cover images (“patchwork” is a method that marks image areas, or patches). These approaches may help protect against image processing such as cropping and rotating, and they hide information more thoroughly than by simple masking. They also support image manipulation more readily than tools that rely on LSB.

In using redundant pattern encoding, you must trade off message size against robustness. For example, a small message may be painted many times over an image, so that if the stegoimage is cropped, there is a high probability that the watermark can still be read. A large message may be embedded only once because it would occupy a much greater portion of the image area.

Other techniques encrypt and scatter the hidden data throughout an image. Scattering the message makes it appear more like noise. Proponents of this approach assume that even if the message bits are extracted, they will be useless without the algorithm and stego-key to decode them. For example, the White Noise Storm tool is based on spread spectrum technology and frequency hopping, which scatters the message throughout the image. Instead of having x channels of communication that are changed with a fixed formula and passkey, White Noise Storm spreads eight channels within a random number generated by the previous window size and data channel. Each channel represents 1 bit, so each image window holds 1 byte of information and many unused bits. These channels rotate, swap, and interlace among themselves to yield a different bit permutation. For instance, bit 1 might be swapped with bit 7, or both bits may rotate one position to the right. The rules for swapping are dictated by the stego-key and by the previous window’s random data (similar to DES block encryption).
Scattering and encryption helps protect against hidden message extraction but not against message destruction through image processing. A scattered message in the image’s LSBs is still as vulnerable to destruction from lossy [2] compression and image processing as is a clear-text message inserted in the LSBs.

Steganography’s niche in security is to supplement cryptography, not replace it. If a hidden message is encrypted, it must also be decrypted if discovered, which provides another layer of protection.

II. PROBLEM SPECIFICATION

Steganalysis is the science of detecting the presence of hidden data in the cover media files and is emerging in parallel with steganography. Steganalysis has gained prominence in national security and forensic sciences since detection of hidden messages can lead to the prevention of disastrous security incidents. While it is possible to design a reasonably good steganalysis [5] technique for a specific steganographic algorithm, the long term goal is to develop a steganalysis framework that can work effectively at least for a class of steganography methods, if not for all. The other problems concerned with the steganalysis techniques are:

• Steganalysis techniques are not adequate to find the contents in image accurately.
• The precision value of the results is very low.
• Training time and execution time of the neural network techniques are extremely high.
• Classification error is also a major issue.
• Steganalysis techniques do not perform well in noisy images.

So, a novel steganalysis technique is very much necessary to overcome the above given problems.

III. PROPOSED METHOD

Accurate prediction of stego-contents in corporate emails is a complex task because of the fast development of security attacks. Several techniques like linear regression, auto regression, Multi Layer Perceptron have been used for prediction. Due to the factors like image quality, strength of the technique used to hide the information behind image and image noise, it is not easy to get an accurate prediction result. So, a novel approach is very much necessary for establishment of detecting stego-contents in corporate emails with effective accuracy and performance. Hybrid network is illustrated, which integrates a Self-Organizing Feature Map (SOFM) and a Multilayer Perceptron Network (MLP) to understand a much better prediction system [6].

Thus, ultimately, results in a network system that can provide significant prediction result with very few inputs.

The Fig 1 shows the architecture of self-organizing network (SOM), which consists of input layer, and Kohonen or clustering layer.

The shadowed units in the Fig 2 represent the processing units. SOM network may cluster the data into N number of classes. An input vector is presented at each step of the self organizing network. These vectors comprise the “setting” of the network. Each new input creates an adaptation of the parameters [6]. If such alterations are properly controlled, the network can construct a type of internal representation of the settings.
IV. MULTI-LAYER PERCEPTRON (MLP)

Multilayer Perceptrons (MLPs) correspond to the most significant class of ANNs in classification, implementing a feed forward, supervised and hetero-associative paradigm. MLPs comprises of various layers of nodes, interconnected via weighted acyclic arcs from each previous layer to the following layer, without lateral or feedback connections. Each node computes a transformed weighted linear combination of its inputs of the form, with the vector of output activations from the preceding layer, the transposed column vector of weights, and a bounded non-decreasing non-linear function, such as the linear threshold or the sigmoid, with one of the weights acting as a trainable bias connected to a constant input. The Fig 3 represents the three layered MLP topology [6] showing the information processing within a node, using a weighted sum as input function, the logistic function as sigmoid activation function and an identity output function.

4.1. SOFM NETWORK

In numerous applications such as in [7] the Kohonen’s self-organizing feature map has been successfully used. SOFM possesses the interesting property of achieving a distribution of weight vectors that approximates the distribution of the input data. This property of the SOFM can be used to generate prototypes which in turn can partition the data into homogeneous groups. This property is used in the proposed methodology.

4.1.1. SOFM Network Architecture

The SOFM is an algorithmic transformation $A^P_{SOFM}: R^p \rightarrow V(R^q)$ that is frequently advocated for visualization of metric-topological relationships and distributional density properties of feature vectors (signals) $X = \{x_1, x_2, ..., x_N\}$ in $R^p$.

SOFM is deployed through a neural architecture as shown in Fig 4 and it is believed to be analogous in some ways to the biological neural network. The visual display generated by SOFM can be used to form hypotheses about topological structure present in X. It is concentrated on (m×n) displays in $R^2$, but in principle X can be transformed onto a display lattice in $R^q$ for any q.

Input vectors $x \in R^p$ are distributed by a fan-out layer to each of the $(m \times n)$ output nodes in the competitive layer as shown in Figure 4.2. Each node in this layer has a weight vector (prototype) $v_{ij}$ attached to it. Let $O_p = \{v_{ij}\} \subset R^p$ denote the set of weight vectors. $O_p$ is (logically) connected to a display grid $O_2 \subset V(R^2)$. (i,j) in the index set $\{(1,2, ..., m) \times (1,2, ..., n)\}$ is the logical address of a cell. There is a one-to-one correspondence between the $m \times np$ vectors $v_{ij}$ and the $m \times n$ cells (i,j), i.e., $O_p \leftrightarrow O_2$.

With a random initialization of the weight vectors $v_{ij}$ the feature mapping algorithm starts. For notational clarity, the double subscripts are suppressed.

Now, let $x \in R^p$ enter the network, and let denote the current iteration number. Find $V_{r,s-1}$ that best matches $x$ in the sense of minimum Euclidean distance in $R^p$.
This vector comprises a (logical) “image” that is the cell in \( O_2 \) with subscript \( r \). Next, a topological (spatial) neighborhood \( N_r(s) \) centered at is defined in \( O_2 \), and its display cell neighbors are located. A 3\( \times \)3 window, \( N(r) \), centered at \( r \) corresponds to the updating of nine prototypes in \( R^p \).

Finally, \( v_{i,s-1} \) and other weight vectors associated with cells in the spatial neighborhood \( N_s(r) \) are updated using the rule as given below:

\[
v_{i,s} = v_{i,s-1} + H_r(s)(x - v_{i,s-1})
\]  

(4.1)

Here, \( r \) is the index of the “winner” prototype and \( \|\*\| \) is the Euclidean norm on \( R^p \).

\[
r = \arg \min_i \left\{ \|x - v_{i,s-1}\| \right\}
\]  

(4.2)

The strength of interaction between cells \( r \) and \( i \) in \( O_2 \) usually decreases with \( s \), and for a fixed \( s \) it decreases as the distance in \( O_2 \) from cell to cell \( i \) increases which is expressed by the function \( H_r(s) \). \( H_r(s) \) is usually expressed as the product of a learning parameter and a lateral feedback function \( g_s(\text{dist}(r,i)) \).

A common choice for \( g_s \), is \( g_s(\text{dist}(r,i))=e^{-\text{dist}^2(r,i)/\sigma_s^2} \). \( \sigma_s \) and \( \sigma_g \) both decrease with \( s \). \( N_r(s) \) is the topological neighborhood which also decreases with \( s \). This method, when repeated long enough, usually preserves spatial order in the sense that weight vectors which are metrically close in \( R^p \) have visually close images in the viewing plane. The SOFM is repeated for \((500\times m\times n)\) steps.

### 4.2. PERFORMANCE EVALUATION

To evaluate the hybrid SOFM-MLP system for detecting the stego-contents in corporate emails against k-Nearest Neighbour classifier (k-NN) and Support Vector Machine (SVM), experiments were carried out using the similar experimental setup and parameters as discussed in the chapter 3.

#### 4.2.1. NRCS Photo Gallery Dataset

**Accuracy**

Accuracy is calculated for k-NN, SVM and the hybrid SOFM-MLP based steganalysis framework for detecting the stego-contents in corporate emails in NRCS photo gallery.

Table 4.1 shows the comparison of the accuracy for the hybrid SOFM-MLP based steganalysis framework with k-NN and SVM.

<table>
<thead>
<tr>
<th>Steganalysis Techniques</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>k-NN</td>
<td>89.80</td>
</tr>
<tr>
<td>SVM</td>
<td>92.65</td>
</tr>
<tr>
<td>Hybrid SOFM-MLP based Steganalysis Framework</td>
<td>98.97</td>
</tr>
</tbody>
</table>

**TABLE 4.1: COMPARISON OF HYBRID SOFM-MLP BASED STEGANALYSIS FRAMEWORK ACCURACY IN NRCS PHOTO GALLERY DATASET**

**Precision**

Table 4.2 shows the comparison of the precision of for the hybrid SOFM-MLP based steganalysis framework with k-NN and SVM in NRCS photo gallery dataset.

<table>
<thead>
<tr>
<th>Steganalysis Techniques</th>
<th>Precision (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>k-NN</td>
<td>72.36</td>
</tr>
<tr>
<td>SVM</td>
<td>79.82</td>
</tr>
<tr>
<td>Hybrid SOFM-MLP based Steganalysis Framework</td>
<td>88.27</td>
</tr>
</tbody>
</table>

**TABLE 4.2: COMPARISON OF HYBRID SOFM-MLP BASED STEGANALYSIS FRAMEWORK PRECISION IN NRCS PHOTO GALLERY DATASET**

**Classification Error**

Table 4.3 shows the comparison of the classification error of the hybrid SOFM-MLP based steganalysis framework with k-NN and SVM in NRCS photo gallery dataset.

<table>
<thead>
<tr>
<th>Steganalysis Techniques</th>
<th>Classification Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>k-NN</td>
<td>10.20</td>
</tr>
<tr>
<td>SVM</td>
<td>7.35</td>
</tr>
<tr>
<td>Hybrid SOFM-MLP based Steganalysis Framework</td>
<td>1.04</td>
</tr>
</tbody>
</table>

**TABLE 4.3: COMPARISON OF HYBRID SOFM-MLP BASED STEGANALYSIS FRAMEWORK CLASSIFICATION ERROR IN NRCS PHOTO GALLERY DATASET**
Execution Time

Table 4.4 shows the execution time taken by the k-NN, SVM and the hybrid SOFM-MLP based steganalysis framework for detecting the stego-contents in corporate emails in NRCS photo gallery. It can be observed that the time required for execution using the hybrid SOFM-MLP based steganalysis framework in NRCS photo gallery dataset is 19.2 seconds, whereas more time is needed by other clustering techniques for execution.

<table>
<thead>
<tr>
<th>Steganalysis Techniques</th>
<th>Execution Time (Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>k-NN</td>
<td>37.2</td>
</tr>
<tr>
<td>SVM</td>
<td>35.6</td>
</tr>
<tr>
<td>Hybrid SOFM–MLP based Steganalysis Framework</td>
<td>19.2</td>
</tr>
</tbody>
</table>

**TABLE 4.4: COMPARISON OF HYBRID SOFM-MLP BASED STEGANALYSIS FRAMEWORK EXECUTION TIME IN NRCS PHOTO GALLERY DATASET**

4.2.2. Camera Images

Accuracy

Accuracy is calculated for k-NN, SVM and the hybrid SOFM–MLP based steganalysis framework for detecting the stego-contents in corporate emails in camera images dataset. Table 4.5 shows the comparison of the accuracy of for the hybrid SOFM-MLP based steganalysis framework with k-NN and SVM.

<table>
<thead>
<tr>
<th>Steganalysis Techniques</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>k-NN</td>
<td>89.97</td>
</tr>
<tr>
<td>SVM</td>
<td>93.24</td>
</tr>
<tr>
<td>Hybrid SOFM–MLP based Steganalysis Framework</td>
<td>98.74</td>
</tr>
</tbody>
</table>

**TABLE 4.5: COMPARISON OF HYBRID SOFM-MLP BASED STEGANALYSIS FRAMEWORK ACCURACY IN CAMERA IMAGES DATASET**

Precision

Table 4.6 shows the comparison of the precision of for the hybrid SOFM-MLP based steganalysis framework with k-NN and SVM in camera images dataset.

<table>
<thead>
<tr>
<th>Steganalysis Techniques</th>
<th>Precision (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>k-NN</td>
<td>74.39</td>
</tr>
<tr>
<td>SVM</td>
<td>81.26</td>
</tr>
<tr>
<td>Hybrid SOFM–MLP based Steganalysis Framework</td>
<td>88.24</td>
</tr>
</tbody>
</table>

**TABLE 4.6: COMPARISON OF HYBRID SOFM-MLP BASED STEGANALYSIS FRAMEWORK PRECISION IN CAMERA IMAGES DATASET**

Classification Error

Table 4.7 shows the comparison of the classification error of the hybrid SOFM-MLP based steganalysis framework with k-NN and SVM in camera images dataset.

<table>
<thead>
<tr>
<th>Steganalysis Techniques</th>
<th>Classification Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>k-NN</td>
<td>10.03</td>
</tr>
<tr>
<td>SVM</td>
<td>6.76</td>
</tr>
<tr>
<td>Hybrid SOFM–MLP based Steganalysis Framework</td>
<td>1.26</td>
</tr>
</tbody>
</table>

**TABLE 4.7: COMPARISON OF HYBRID SOFM-MLP BASED STEGANALYSIS FRAMEWORK CLASSIFICATION ERROR IN CAMERA IMAGES DATASET**

Execution Time

Table 4.8 shows the execution time taken by the k-NN, SVM and the hybrid SOFM-MLP based steganalysis framework for detecting the stego-contents in corporate emails in camera images dataset. It can be observed that the time required for execution using the hybrid SOFM-MLP based steganalysis framework in camera images dataset is 19.1 seconds, whereas more time is needed by other clustering techniques for execution.

<table>
<thead>
<tr>
<th>Steganalysis Techniques</th>
<th>Execution Time (Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>k-NN</td>
<td>35.8</td>
</tr>
<tr>
<td>SVM</td>
<td>33.2</td>
</tr>
<tr>
<td>Hybrid SOFM–MLP based Steganalysis Framework</td>
<td>19.1</td>
</tr>
</tbody>
</table>

**TABLE 4.8: COMPARISON OF HYBRID SOFM-MLP BASED STEGANALYSIS FRAMEWORK EXECUTION TIME IN CAMERA IMAGES DATASET**
V. SUMMARY

This research work provides the complete discussion of the third proposed methodology that uses Self-Organizing Feature Map (SOFM) and a Multilayer Perceptron Network (MLP) for steganalysis framework to detect stego-contents in corporate emails. Its performance has been evaluated against the k-NN and SVM using the same datasets and performance measures as discussed in chapter III, and it is found that this method using hybrid SOFM-MLP performs better than the k-NN and SVM.

REFERENCES


[10] Samir K Bandyopadhyay, and Debnath Bhattacharyya, 2008, A Tutorial Review on Steganography, University of Calcutta, Senate House, 87 /1 College Street, Kolkata, UFL & JIITU.

Authors:

Dr. Anitha Thangasami having 20 years of teaching experience and presently working as an Associate Professor in the Department of Informatics, Wollega University, Nekemte, Ethiopia. She was awarded her Doctorate degree in Anna University, Chennai, Tamilnadu, India. Her research interest includes Network Security, Cryptography and Steganography. She is interested in guiding research students to shine in their research fields.

Dr. M. I. THARIQ HUSSAN is working as a Professor in Guru Nanak Institutions Technical Campus, Hyderabad, India. He was awarded his Doctoral Degree in the Faculty of Information and Communication Engineering from Anna University, Chennai, Tamilnadu. He has 4 International Journal publications and 2 National publications. He has presented papers in 25 International/National conferences and attended 31 Seminars/Workshops/FDP. He is a life member of ISTE (Indian Society for Technical Education) and various other International Computer Societies and organization for knowledge exchange and enhancement. He is having 2 years of industry and 15 years of teaching experience. His research areas include Sequential Pattern Mining and Mobile Computing.