Volume 5 ▪ Issue 5 ▪ December 2011

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE AND SECURITY (IJCSS)

ISSN : 1985-1553

Publication Frequency: 6 Issues / Year

CSC PUBLISHERS
http://www.cscjournals.org

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE AND SECURITY (IJCSS)

VOLUME 5, ISSUE 5, 2011

EDITED BY
DR. NABEEL TAHIR

ISSN (Online): 1985-1553

International Journal of Computer Science and Security is published both in traditional paper form and in Internet. This journal is published at the website http://www.cscjournals.org, maintained by Computer Science Journals (CSC Journals), Malaysia.

IJCSS Journal is a part of CSC Publishers
Computer Science Journals
http://www.cscjournals.org

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE AND SECURITY (IJCSS)

Book: Volume 5, Issue 5, December 2011
Publishing Date: 15 - 12- 2011
ISSN (Online): 1985 -1553

IJCSS Journal is a part of CSC Publishers
http://www.cscjournals.org

Published in Malaysia

Typesetting: Camera-ready by author, data conversation by CSC Publishing Services – CSC Journals, Malaysia

CSC Publishers, 2011

EDITORIAL PREFACE

This is fifth issue of volume five of the International Journal of Computer Science and Security (IJCSS). IJCSS is an International refereed journal for publication of current research in computer science and computer security technologies. IJCSS publishes research papers dealing primarily with the technological aspects of computer science in general and computer security in particular. Publications of IJCSS are beneficial for researchers, academics, scholars, advanced students, practitioners, and those seeking an update on current experience, state of the art research theories and future prospects in relation to computer science in general but specific to computer security studies. Some important topics cover by IJCSS are databases, electronic commerce, multimedia, bioinformatics, signal processing, image processing, access control, computer security, cryptography, communications and data security, etc.

The initial efforts helped to shape the editorial policy and to sharpen the focus of the journal. Starting with volume 5, 2011, IJCSS appears in more focused issues. Besides normal publications, IJCSS intend to organized special issues on more focused topics. Each special issue will have a designated editor (editors) – either member of the editorial board or another recognized specialist in the respective field.

This journal publishes new dissertations and state of the art research to target its readership that not only includes researchers, industrialists and scientist but also advanced students and practitioners. The aim of IJCSS is to publish research which is not only technically proficient, but contains innovation or information for our international readers. In order to position IJCSS as one of the top International journal in computer science and security, a group of highly valuable and senior International scholars are serving its Editorial Board who ensures that each issue must publish qualitative research articles from International research communities relevant to Computer science and security fields.

IJCSS editors understand that how much it is important for authors and researchers to have their work published with a minimum delay after submission of their papers. They also strongly believe that the direct communication between the editors and authors are important for the welfare, quality and wellbeing of the Journal and its readers. Therefore, all activities from paper submission to paper publication are controlled through electronic systems that include electronic submission, editorial panel and review system that ensures rapid decision with least delays in the publication processes.

To build its international reputation, we are disseminating the publication information through Google Books, Google Scholar, Directory of Open Access Journals (DOAJ), Open J Gate, ScientificCommons, Docstoc and many more. Our International Editors are working on establishing ISI listing and a good impact factor for IJCSS. We would like to remind you that the success of our journal depends directly on the number of quality articles submitted for review. Accordingly, we would like to request your participation by submitting quality manuscripts for review and encouraging your colleagues to submit quality manuscripts for review. One of the great benefits we can provide to our prospective authors is the mentoring nature of our review process. IJCSS provides authors with high quality, helpful reviews that are shaped to assist authors in improving their manuscripts.

Editorial Board Members
International Journal of Computer Science and Security (IJCSS)

EDITORIAL BOARD

TABLE OF CONTENTS

Volume 5, Issue 5, December 2011

Pages

Non-Specialized File Format Extension

Blake W. Ford blake.wford@gmail.com
Department of Computer Science
Texas State University – San Marcos
San Marcos, 78666, USA

Khosrow Kaikhah kk02@txstate.edu
Department of Computer Science
Texas State University – San Marcos
San Marcos, 78666, USA

Abstract

The study expands upon previous work in format extension. The initial research purposed extra space provided by an unrefined format to store metadata about the file in question. This process does not negatively impact the original intent of the format and allows for the creation of new derivative file types with both backwards compatibility and new features. The file format extension algorithm has been rewritten entirely in C++ and is now being distributed as an open source C/C++ static library, roughdraftlib. The files from our previous research are essentially binary compatible though a few extra fields have been added for developer convenience. The new data represents the current and oldest compatible versions of the binary and values representing the scaling ratio of the image. These new fields are statically included in every file and take only a few bytes to encode, so they have a trivial effect on the overall encoding density.

Keywords: Steganography, CAD, Metadata, Compatibility.

1. BACKGROUND

Simple interactions between engineers and their clients can at times be surprisingly difficult, because most contemporary drafting programs use proprietary technologies to archive data in application specific formats. When sharing data with clients, source files either need to be converted to a format in common usage or the client will need to install an application specific viewing program. In either case, the maintenance associated with keeping the client up to date can be tedious. To resolve this issue we created a single sourcing steganography library called roughdraftlib.

This library integrates high-level design data into a standardized file format while maintaining backwards compatibility. Steganography is used to create an outlet for adding additional hidden information to the standardized file. Using our software, it is possible to build a single source format that is convenient for clients and workable for developers.

2. PORTABLE NETWORK GRAPHICS

Language unifying the code and developing a new internal architecture have made it much easier to expand the usefulness of the product. The software now has a pluggable target interface which allows for greater target diversity than the previous architecture. The most important improvement over the previous system to date is the addition of the Portable Network Graphics format as a possible target made possible by these changes. The PNG files exported by roughdraftlib are of the same quality as the old 24-bit bitmap targets, but with a much smaller disk footprint. This format uses the DEFLATE algorithm discussed previously to compress the image data. However, the effective compression of the DEFLATE algorithm on the raster data for these images is far greater than it was on the vector data according to our findings. In one test case, the PNG file produced was 1% the size of the already compressed 256-color bitmap produced by the older

RoughDraft application. DEFLATE's underlying LZ77 compression algorithm works well with repetitive datasets and the wire frames we have been testing are highly repetitive. In addition, roughdraftlib uses a feature of the PNG file format to hide data within an image without having to manipulate the visible data.

The information within a PNG file is broken up into small sections known as chunks. Chunks can either be critical or ancillary. As a rule, all PNG decoders are required to process critical chunks, but opt in to processing ancillary chunks. If a decoder cannot process an ancillary chunk, it typically ignores that piece of data. In roughdraftlib, all of the CAD data is stored in an ancillary chunk instead of in the least significant bits of the image data. Though either method is possible to implement using the PNG format, this method was chosen because it does not restrict the amount of data that can be embedded within the image.

For these reasons, PNG is likely to be the most popular target of roughdraftlib. Noting this, we had to revisit our previous choices regarding the layout and the future roadmap for the technology. Because the PNG format is more flexible and so different from the bitmap files studied before, some of our optimizations may no longer make sense when propositioning this format to outside vendors, while other considerations from our previous research still hold true in the PNG environment. The goal of this research is to verify the effectiveness of our format design and reevaluate the old bitmap format specific decisions that may hold back the adoption of vector bitmaps.

3. FEATURE EVALUATION

One universal feature that adds value to all of our potential targets is secondary format compression. While this may provide software developers with some additional overhead, the consumer facing experience is vastly improved in every domain. There is no tradeoff to be made here in terms of developer efficiency and disk footprint, because ample developer resources exist to attack and understand the problem of DEFLATE compression. Libraries and documentation on the algorithm are available for a variety of programming languages. Stylistically, few changes are foreseen in regards to the basic process by which roughdraftlib processes data, embeds its payload and targets individual file formats though some areas will require change.

In our initial research, we discounted the use of off the shelf file types due to their comparatively large size. However, now that a target exists that allows for boundless secondary format size we have called into question the importance of an extremely small but limited secondary data format. For instance, in the original research project, we found that embedding a standard DXF file in place of our custom format limits the maximum number of shapes possible in bitmap targets to about one tenth their previous value. The importance of this seemly negative statistic can now be reevaluated, considering that bitmaps will likely not be the most popular target of this library. Therefore, we are not optimizing this use case and using DXF as the backend format for all targets in an effort to increase adoption.

4. QCAD INSTRUMENTATION

We extended an open source version of the QCAD drafting application. With these additions to the code base, it is now possible to export CAD drawings in any of the formats available in roughdraftlib, PNGs with a compressed DXF chunk, or raw text based DXF files natively supported by QCAD. Through this exercise, we have been able to gather important size and integration data from which we will base the future roadmap of roughdraftlib.

QCAD was chosen as the host platform for a variety of reasons. Not only is the QCAD application industry capable, it is also open source and relatively easy to extend for our purposes. In addition, the creators of the application have also produced a suite of tools to convert DXF files into standard image formats. We assume since these tools exist, QCAD users are dealing with the duplicated source issues we are trying to solve. For this reason, QCAD users may be interested in converting to our approach. Lastly, QCAD ships with a standard DXF part library. The library contains CAD drawings of common blocks used in industry, like doors and appliances. We used these blocks as our baseline to ensure that our technology supports both the features and performance expected by CAD designers.

Using our instrumented QCAD program, we were able to produce some convincing size numbers for migration to either the highly compressed vector bitmap format or the easy to use DXF chunk derivative. We start each experiment with a standard part from the shipping library. This part is exported into each possible format discussed above. Then a qualitative assessment is made regarding the resulting image's quality followed by some file size comparisons.

For now, because all of the resulting images are lossless, there are only three static buckets of qualitative assessment. Our DXF chunk images have the highest quality, because they represent each CAD image verbatim. Vector bitmaps lag behind, because it does not yet support all of the shapes possible in a given DXF drawing, and pure DXF files are considered last as they cannot be viewed using standard imaging programs.

There are two primary size comparisons being made from each exported file. The first is raw file size to compare the disk footprint of the bitmaps, PNGs, and DXF files produced. This gives developers some indication of how standardizing on the end product would affect their systems. The second metric is a comparison of the two competing secondary data formats. The size of the CAD data in the final product determines how useful bitmaps and other steganographically limited file extensions will be in different applications.

File Type	File Size
24-bit bitmap	921.00 KB
256-color bitmap	307.00 KB
Raw DXF	21.00 KB
DXF Chunk PNG	5.25 KB
Roughdraftlib PNG	2.50 KB

TABLE 1: Raw File Size

The referenced tables were generated using an appliance drawing that ships with QCAD as a sample DXF file. The numbers represent the real world performance of the different encoding methods. Table 1 shows how each variation compares in terms of file size. Bitmaps produced under either the previous or DXF extension mechanisms have identical disk size and are generally larger than the source DXF file. The PNGs differ in size and the roughdraftlib variant is significantly smaller. It should be noted that both PNG exports are smaller than the original DXF source file.

Looking at the data size and capacity statistics helps demonstrate our conflict of interests. Table 2 depicts the size of each secondary format and Table 3 displays the embedding capacity of all possible targets. Table 4 illustrates the compatibility matrix derived from the previous two tables.

File Type	Format Size
Raw DXF	21.00 KB
DXF Chunk	3.00 KB
Roughdraftlib	300 B

TABLE 2: Data Format Size

File Type	Capacity
DXF, PNG types	Unlimited KB
24-bit bitmap	77 KB
DXF Chunk 256 bmp	1 KB
Roughdraft 256 bmp	960 B

TABLE 3: Data Capacity

File Type	Raw DXF	DXF Chunk	RoughDraft
DXF, PNG types	OK	OK	OK
24-bit bitmaps	OK	OK	OK
DXF Chunk 256 bmp	NP	NP	OK
Roughdraft256 bmp	NP	NP	OK

TABLE 4: Aggregate Data

This support matrix is representative of most files tested from the QCAD sample library. The roughdraftlib data format is the only option that allows us to support all of the target configurations currently available for this use case. When using the compressed DXF chunk as the secondary data backend, the resulting file typically fits into all but the 256-color bitmap target's available embedding space. For more complex designs, it should be noted that while the designs tested worked with our 24-bitmap targets the overall capacity for that format was greatly reduced when the DXF chunk method was used; approximately one-tenth the capacity of the roughdraftlib representation in this test.

5. NEW VALUE EQUATIONS

We established a relationship between this growth in data size and the number of shapes possible to embed in a file. This helps to clarify our arguments in regards to each of the secondary formats. In our original research, we used a static equation to determine the size of a secondary format in bitmap files.

Original Algorithm
File Size < 54(Header) + (Shapes)*72 (1)

Using this equation, we determined that 24-bit bitmap with the dimensions 500x500 should be able to store around 10,000 shapes. Using compression, we increased this number conservatively by 30% for a final total of approximately 13,000 shapes when using the roughdraftlib format.

Compressed Algorithm
File Size < 54(Header)+(Shapes)*50 (2)

Because DXF files scale in a slightly different fashion we need to derive a second equation for competitive analysis. First, using QCAD we noticed that an empty DXF file defaults to roughly 11KB in size. With each additional shape, the file grows by an average of 130 bytes. When compressed, we are observing about an 80% decrease in the empty file size plus a 40 byte size increase per shape. Using this information, we derived the following corresponding equations.

DXF Algorithm

File Size < 54+88K+(Shapes)*1.2K (3)

Compressed DXF Algorithm
File Size < 54+16.8 K+(Shapes)*320 (4)

Applying our overhead tax and shape penalty rules for the DXF chunk format, we estimate that designs 2,400 or few shapes will be possible for an equal sized 24-bit bitmap encoded using our previous research.

For 256-color bitmaps a similar equations can be derived. In our earlier research, we chose not to noticeably change any of the 256 colors from the bitmaps palette. Since then, we have updated our approach. In our latest design, the color palette is artificially limited to 16 colors. The remaining dictionary space is filled with compressed secondary data. This increases the data size available for embedding shapes from 320 to 960 bytes; this is reflected in the tables [1]-[4]. For this format, we no longer hide data in the least significant bits of the color palette, so the overhead for each shape goes down as well. The following equations represent our current strategy for 256-color bitmaps.

Original Algorithm
960 Bytes < (Shapes)*9(Raw Size) (5)

Compressed Algorithm
960 Bytes < (Shapes)*6(Raw Size) (6)

DXF Algorithm
960 Bytes < 11KB+(Shapes)*130 (7)

Compressed DXF Algorithm
960 Bytes < 2.1KB+(Shapes)*40 (8)

Using these equations, linear placement would yield 105 possible shapes, compression would increase this number to around 135 and the DXF algorithms would be impossible.

With these results, we either have to consider either dropping the DXF chunk representation for 256-color bitmaps or improving our embedding technique. As there are two other file format choices, losing this target to gain the flexibility of the DXF chunk type does not seem like an unreasonable tradeoff, however, we also explored ways to more efficiently embed into 256-color targets. Our most recent approach involves encoding data into the bit field by duplicating the restricted color dictionary.

256-color bitmaps allow identical colors to be defined multiple times in their dictionary. Using this feature, we would expand our current dictionary usage from 16 colors to 32 by redefining the original set. This would limit free space in the dictionary from 960 bytes to 896, an initial loss of 64 bytes. However, this would allow us to assign logical 0s and 1s to the data in the image's bit field. If a color reference in the bit field pointed to a value from the first set of 16 colors, it would indicate a 0 in the secondary file format. Likewise, a color pointing to a value in the second set would represent a 1. With this mechanism, we would recover the initial loss of 64 bytes in 512 pixels. If the area of the image is larger than 512 pixels, this method would allow for more encoding space than the previous version. The new algorithm would yield 32KB worth of encoding space from an image with our baseline dimensions of 500x500 pixels.

Original Algorithm
File Size < 54+32(Dict.)+(Shapes-100)*72 (9)

Compressed Algorithm
File Size < 54+32(Dict.)+(Shapes-100)*50 (10)

DXF Algorithm
File Size < 54+32+87K+(Shapes)*1.2K (11)

Compressed DXF Algorithm
File Size < 54+32+15.8K+(Shapes)* 320 (12)

6. SOFTWARE STACK COMPARISON

Taking into consideration all of this information, we intend to push our current software stack in a slightly new direction. Figure 1 shows the stack as it currently exists with the QCAD application.

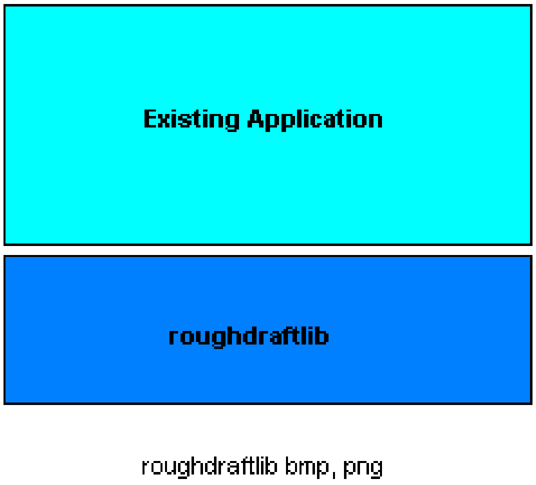


FIGURE 1: Stack for QCAD application

We copied and slightly modified the critical source from roughdraftlib in order to test the new standardized file approach to the single source problem. The resulting code is structured according to Figure 2. This code base has been name-spaced in order to allow two similar code paths to exist side by side for comparison. Currently the new DXF based single source utility is dependent upon the existing applications ability to produce one of the three carrier file types supported by our research.

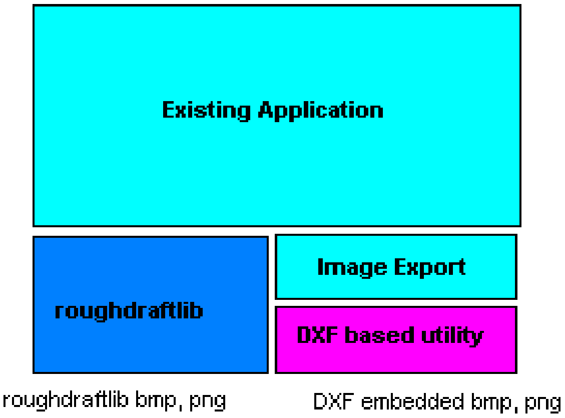


FIGURE 2: New Standardized File

We propose a new independent library derived from the embedding portion of the roughdraftlib code base. Instead of duplicating this code as we are now, roughdraftlib will become dependent upon the lower level-embedding library as depicted in Figure 3.

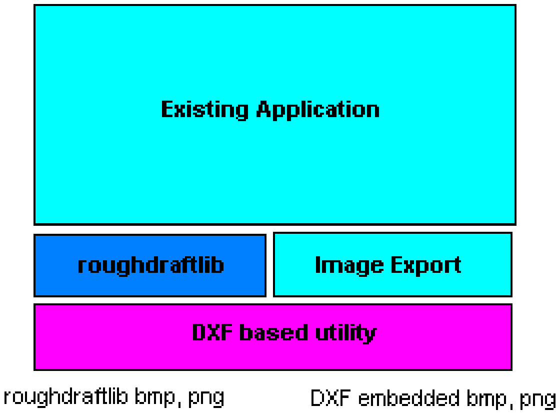


FIGURE 3: New Independent File

By moving the steganography code into a new independent layer, we give developers the option of using their own image file creation routines and standardized formats while preserving the integrity of our industry optimized approach. This will provide more flexibility when both adopting and continuing to add support for new features related to our research. With this software stack, existing applications can plug in with almost no effort and grow into our optimized format at their leisure. In addition, other industries can also take advantage of single sourcing immediately without the overhead of defining a fully optimized single source version.

7. CONCLUSIONS AND FUTURE GOALS

Following the previously mentioned code restructuring, we would like to target additional file types. The next likely candidate would be the JPEG file format. This format is a popular web format like PNG and can also be used in steganography. While the JPEG format can out perform PNG compression on certain images types, it is relatively ineffective when compressing diagrams, graphs, text, icons, and monochrome images. CAD images exhibit many of these qualities and we have observed that the size of the JPEG images we could produce through roughdraftlib would be around 500% larger than the comparable PNG version. Though this seems like a very large difference, keep in mind a JPEG image produced from roughdraftlib would still be drastically smaller than either of the bitmap files we currently export.

In this research iteration, we also leverage the work of an open source application to define and improve the usability of our product. This is a tread that will be repeated in the future of this product as well. One project of interest is Steghide source integration. Steghide is an open source steganography application that supports some of the formats targeted by roughdraftlib. Steghide has many interesting features that roughdraftlib does not like embedding capacity reporting and encryption and so long as we keep the appropriate licensing terms we can also take advantage of those features.

8. REFERENCES

[1] B. W. Ford and K, Kaikhah. “File Format Extension Through Steganography,” presented at the *International Conference on Software Engineering, Management & Application*, Kathmandu, Nepal, 2010.

[2] B. W. Ford and K, Kaikhah. “Honing File Format Extension Through Steganography,” presented at the I*nternational Conference on Infocomm Technologies in Competitive Strategies*, Singapore, 2010.

[3] G. Cantrell and D. D. Dampier. “Experiments in hiding data inside the file structure of common office documents: a steganography application.” *In Proceedings of the International Symposium on information and Communication Technologies*, 2004, pp. 146-151.

[4] G.A. Francia and T. S. Gomez. “Steganography obliterator: an attack on the least significant bits.” *In Proceedings of the 3rd Annual Conference on Information Security Curriculum Development*, 2006, pp. 85-91.

[5] J. Fridrich. “Minimizing the embedding impact in steganography.” In Proceedings of the 8th Workshop on Multimedia and Security, 2006, pp. 2-10.

[6] J. Fridrich, T. Pevný, and J. Kodovský. “Statistically undetectable jpeg steganography: dead ends challenges, and opportunities.” *In Proceedings of the 9th Workshop on Multimedia and Security*, 2007, pp. 3-14.

[7] C. M.C. Chen, S. S. Agaian, and C. L. P. Chen. “Generalized collage steganography on images." *In* Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, 2008, pp. 1043-1047.

[8] Z. Oplatkova, J. Holoska, I. Zelinka, and R. Senkerik. “Detection of Steganography Inserted by OutGuess and Steghide by Means of Neural Networks.” *In Proceedings of the Third Asia International Conference on Modeling and Simulation*, 2009, pp. 25-29.

Design and Implementation of EZW & SPIHT Image Coder for Virtual Images

Priyanka Singh *priyanka10ec@gmail.com*
Research Scholar, ECE Department
Amity University
Gurgaon (Haryana, India)

Priti Singh *pritip@rediffmail.com*
Professor, ECE Department
Amity University
Gurgaon (Haryana, India)

Abstract

The main objective of this paper is to designed and implemented a EZW & SPIHT Encoding Coder for Lossy virtual Images. Embedded Zero Tree Wavelet algorithm (EZW) used here is simple, specially designed for wavelet transform and effective image compression algorithm. This algorithm is devised by Shapiro and it has property that the bits in the bit stream are generated in order of importance, yielding a fully embedded code. SPIHT stands for Set Partitioning in Hierarchical Trees. The SPIHT coder is a highly refined version of the EZW algorithm and is a powerful image compression algorithm that produces an embedded bit stream from which the best reconstructed images. The SPIHT algorithm was powerful, efficient and simple image compression algorithm. By using these algorithms, the highest PSNR values for given compression ratios for a variety of images can be obtained. SPIHT was designed for optimal progressive transmission, as well as for compression. The important SPIHT feature is its use of embedded coding. The pixels of the original image can be transformed to wavelet coefficients by using wavelet filters. We have anaysized our results using MATLAB software and wavelet toolbox and calculated various parameters such as CR (Compression Ratio), PSNR (Peak Signal to Noise Ratio), MSE (Mean Square Error), and BPP (Bits per Pixel). We have used here different Wavelet Filters such as Biorthogonal, Coiflets, Daubechies, Symlets and Reverse Biorthogonal Filters .In this paper we have used one virtual Human Spine image (256X256).

Keywords: Image Compression, Embedded Zerotree Wavelet, Set Partitioning in Hierarchical Trees, CR, PSNR, MSE, BPP.

1. INTRODUCTION

With the growth of technology and the entrance into the Digital Age, the world has found itself a vast amount of information. Dealing with such enormous amount of information can often present difficulties. Digital information must be stored, retrieved, analyzed and processed in an efficient manner, in order for it to be put to practical use. Image compression is technique under image processing having wide variety of applications. Image data is perhaps the greatest single threat to the capacity of data networks [1,2].As image analysis systems become available on lower and lower cost machines, the capability to produce volume of data becomes available to more and more users. New data storage technologies have been developed to try to keep pace with the potential for data creation.

1.1 Image Compression

The fundamental components of compression are redundancy and irrelevancy reduction. Redundancy means duplication and Irrelevancy means the parts of signal that will not be noticed by the signal receiver, which is the Human Visual System (HVS).
There are three types of redundancy can be identified:

- **Spatial Redundancy** i.e. correlation between neighboring pixel values.
- **Spectral Redundancy** i.e. correlation between different color planes or spectral bands.

- **Temporal Redundancy** i.e. correlation between adjacent frames in a sequence of images.

Image compression focuses on reducing the number of bits needed to represent an image by removing the spatial and spectral redundancies. The removal of spatial and spectral redundancy is often accomplished by the predictive coding or transforms coding. Quantization is the most important means of irrelevancy reduction [2].

1.2 Basic Types of Image Compression

Basic types of image compression are lossless and lossy. Both compression types remove data from an image that isn't obvious to the viewer, but they remove that data in different ways. Lossless compression works by compressing the overall image without removing any of the image's detail. As a result the overall file size will be compressed. Lossy compression works by removing image detail, but not in such a way that it is apparent to the viewer. In fact, lossy compression can reduce an image to one tenth of its original size with no visible changes to image quality [2, 3].

One of the most successful applications of wavelet methods is transform-based image compression (also called coding). The overlapping nature of the wavelet transform alleviates blocking artifacts, while the multiresolution character of the wavelet decomposition leads to superior energy compaction and perceptual quality of the decompressed image. Furthermore, the multiresolution transform domain means that wavelet compression methods degrade much more gracefully than block-DCT methods as the compression ratio increases. Since a wavelet basis consists of functions with both short support (for high frequencies) and long support (for low frequencies), large smooth areas of an image may be represented with very few bits, and detail added where it is needed. Wavelet-based coding provides substantial improvements in picture quality at higher compression ratios. Over the past few years, a variety of powerful and sophisticated wavelet-based schemes for image compression, as discussed later, have been developed and implemented. Because of the many advantages, wavelet based compression algorithms are the suitable candidates for the new JPEG-2000 standard. Such a coder operates by transforming the data to remove redundancy, then quantizing the transform coefficients (a lossy step), and finally entropy coding the quantizer output. The loss of information is introduced by the quantization stage which intentionally rejects less relevant parts of the image information. Because of their superior energy compaction properties and correspondence with the human visual system, wavelet compression methods have produced superior objective and subjective results [4]. With wavelets, a compression rate of up to 1:300 is achievable [5]. Wavelet compression allows the integration of various compression techniques into one algorithm. With lossless compression, the original image is recovered exactly after decompression. Unfortunately, with images of natural scenes, it is rarely possible to obtain error-free compression at a rate beyond 2:1 [5-6]. Much higher compression ratios can be obtained if some error, which is usually difficult to perceive, is allowed between the decompressed image and the original image.

2. EMBEDDED ZERO TREE WAVELET (EZW)

The EZW algorithm was introduced in the paper of Shapiro [2]. The core of the EZW compression is the exploitation of self-similarity across different scales of an image wavelet transform. The Embedded Zero-tree Wavelet (EZW) algorithm is considered the first really efficient wavelet coder. Its performance is based on the similarity between sub-bands and a successive-approximations scheme. Coefficients in different sub-bands of the same type represent the same spatial location, in the sense that one coefficient in a scale corresponds with four in the prior level. This connection can be settled recursively with these four coefficients and its corresponding ones from the lower levels, so coefficient trees can be defined. In natural images most energy tends to concentrate at coarser scales (higher levels of decomposition), then it can be expected that the

nearer to the root node a coefficient is, the larger magnitudes it has. So if a node of a coefficient tree is lower than a threshold, it is likely that its descendent coefficients will be lower too.
We can take profit from this fact, coding the sub-band coefficients by means of trees and successive-approximation, so that when a node and all its descendent coefficients are lower than a threshold, just a symbol is used to code that branch The successive-approximation can be implemented as a bit-plane encoder. The EZW algorithm is performed in several steps, with two fixed stages per step: the dominant pass and the subordinate pass. In Shapiro's paper the description of the original EZW algorithm can be found. However, the algorithm specification is given with a mathematical outlook. We present how to implement it, showing some implementation details and their impact on the overall codec performance. Consider we need n bits to code the highest coefficient of the image (in absolute value). The first step will be focused on all the coefficients that need exactly n bits to be coded. In the dominant pass, the coefficients which falls (in absolute value) in this range are labeled as a significant positive/negative (sp/sn), according to its sign. These coefficients will no longer be processed in further dominant passes, but in subordinate passes. On the other hand, the rest of coefficients are labeled as zero-tree root (zr), if all its descendants also belong to this range, or as isolated zero (iz), if any descendant can be labeled as sp/sn. Notice that none descendant of a zero-tree root need to be labeled in this step, so we can code entire zero-trees with just one symbol. In the subordinate pass, the bit n of those coefficients labeled as sp/sn in any prior step is coded. In the next step, the n value is decreased in one so we focus now on the following least significant bit [7]. Compression process finishes when a desired bit rate is reached. That is why this coder is so called embedded. In the dominant pass four types of symbols need to be code (sp, sn, zr, iz), whereas in the subordinate pass only two are needed (bit zero and bit one). Finally, an adaptive arithmetic encoder is used to get higher entropy compression. EZW approximates higher frequency coefficients of a wavelet transformed image. Because the wavelet transform coefficients contain information about both spatial and frequency content of an image, discarding a high-frequency coefficient leads to some image degradation in a particular location of the restored image rather than across the whole image. Here, the threshold is used to calculate a significance map of significant and insignificant wavelet coefficients. Zerotrees are used to represent the significance map in an efficient way. Figure 1 shows the Embedded Zerotree Scanning process.

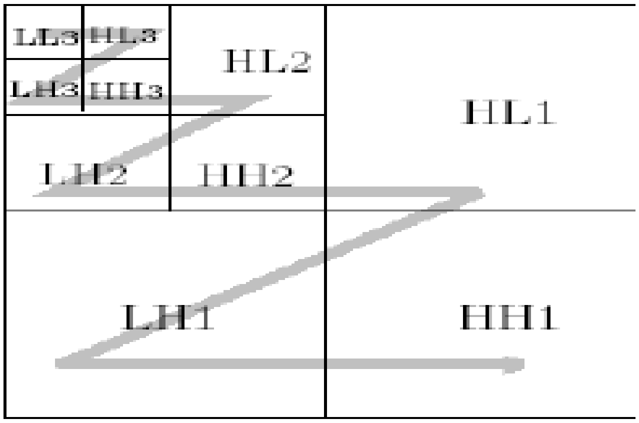


FIGURE 1 Scanning of a Zerotree

3. EZW ENCODING ALGORITHM

1. Initialization: Set the threshold T to the smallest power of that is greater than max (i,j) |c i,j|/2, where Ci,j are the wavelet coefficients.

2. Significance map coding: Scan all the coefficients in a predefined way and output a symbol when |Ci,j | > T. When the decoder inputs this symbol, it sets ci,j = ±1.5T.

3. Refinement: Refine each significant coefficient by sending one more bit of its binary representation. When the decoder receives this, it increments the current coefficient value by ±0.25T.

4. Set Tk = Tk-1/2, and go to step 2 if more iterations are needed [7-8].

The next scheme, called SPIHT, is an improved form of EZW which achieves better compression and performance than EZW.

4. SET PARTITIONING IN HIERARICHAL TREES (SPIHT)

SPIHT sorts the coefficients and transmits their most significant bits first. A wavelet transform has already been applied to the image and that the transformed coefficients are sorted.The next step of the encoder is the refinement pass. The encoder performs a sorting step and a refinement step in each iteration. SPIHT uses the fact that sorting is done by comparing two elements at a time, and each comparison results in a simple yes/no result. The encoder and decoder use the same sorting algorithm, the encoder can simply send the decoder the sequence of yes/no results, and the decoder can use those to duplicate the operations of the encoder. The main task of the sorting pass in each iteration is to select those coefficients that satisfy 2n<=|ci,j |<2n+1. This task is divided into two parts. For a given value of n, if a coefficient ci,j satisfies |ci,j|>=2n, then that it is said as significant; otherwise, it is called insignificant. The encoder partitions all the coefficients into a number of sets Tk and performs the significance test.

$$S_n(T) = \begin{cases} 1, \max_{(i,j)\in T} |C_{I,J}| \geq 2^N \\ 0, Otherwise. \end{cases}$$

…………………………..eq. (1)

On each set Tk. The result may be either “no” This result is transmitted to the decoder. If the result is “yes,” then Tk is partitioned by both encoder and decoder, using the same rule, into subsets and the same significance test is performed on all the subsets. This partitioning is repeated until all the significant sets are reduced to size 1. The result, Sn(T), is a single bit that is transmitted to the decoder.

The sets Tk are created and partitioned using a spatial orientation tree. This set partitioning sorting algorithm uses the following four sets of coordinates:

1. The set contain the coordinates of the four offspring of node is Off [i,j]. If node is a leaf of a spatial orientation tree, then Off [i, j] is empty.

2. The set contain the set of coordinates of the descendants of node is called Des[i,j].

3. The set contain the set of coordinates of the roots of all the spatial orientation trees called R.

4. Next the set is a difference set Des[i,j]- Off[i,j]. This set contains all the descendants of tree node except its four offspring as Diff [i,j].

The spatial orientation trees are used to create and partition the sets Tk. The partitioning rules are given below:

1. Each spatial orientation tree need initial set.

2. If set Des[i, j] is significant, then it is partitioned into Diff[i, j] plus the four single element sets with the four offspring of the node.

3. If Diff[i, j] is significant, then it is partitioned into the four sets Des[k, l], where k=1..4 of node .

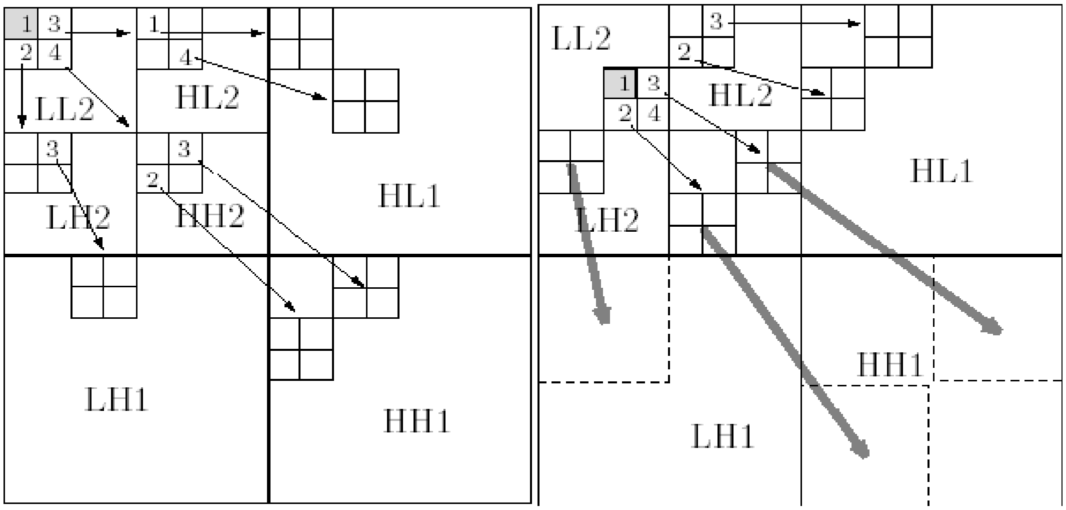


FIGURE 2 Shows the Spatial Orientation Trees in SPIHT.

5. SPIHT Algorithm

It is important to have the encoder and decoder test sets for significance .So the coding algorithm uses three lists called SP for list of significant pixels, initialized as empty, IP is list of insignificant pixels for the coordinates of all the root node belongs to root set R, and IS is list of insignificant sets to the coordinates of all the root node in R that have descendants and treated as special type entries [5].

Procedure:

```
Step 1: Initialization: Set n to target bit rate.
               for each node in IP do:
                  if Sn [ i, j] = 1,(according to eq 4.1)
                   move pixel coordinates to the SP and
                       keep the sign of ci,j ;
Step 2: for each entry in the IS do the following steps:
              if the entry is root node with descendants
                    if Sn(Des[i, j]) = 1, then
                     for each offspring (k, i ) in Off[i, j] do:
                      if ( Sn(k, i) = 1) then
                               { add to the SP,
                                      output the sign of ck,l;}
else
  attach (k, l) to the IP;
         if (Diff[i, j] <> 0)
          {move (i, j) to the end of the IS,
                        go to X;}
else
   remove entry from the IS;
        If the entry is root node without descendants then
            output Sn(Diff[i, j]);
                if Sn(Diff[i, j]) = 1, then
                    append each (k, l) in Off(i, j) to the IS as a special
                           entry and remove node from the IS:
```

Step 3: Refinement pass: for each entry in the SP, except those included in the last process for sorting, output the nth most significant bit of |i,j|;

Step 4: Loop: reduced n by 1 and go to X if needed.

6. Experimental Results & Analysis

In this paper we have implemented our result on a grayscale virtual image named Humane Spine having size (256X256) using various Wavelet filter families. Here we used MATLAB 2011(a) software and wavelet toolbox for analysing our results. The results of experiments are used to find the CR (Compression Ratio), BPP (Bits per Pixel), PSNR (Peak Signal to Noise Ratio) values and MSE (Mean Square Error) values for the reconstructed images. Fig 3(a) shows the Original Image and fig 3(b) shows the compressed image by EZW. Similarly fig 4(a) shows original image and fig 4(b) shows the compressed image by SPIHT image compression algorithms. The result got by EZW & SPIHT is shown in the Following Tables 1-2.Table 1 shows the results of CR, BPP, MSE & PSNR by using EZW algorithm. Table 2 shows the values for SPIHT Algorithms.

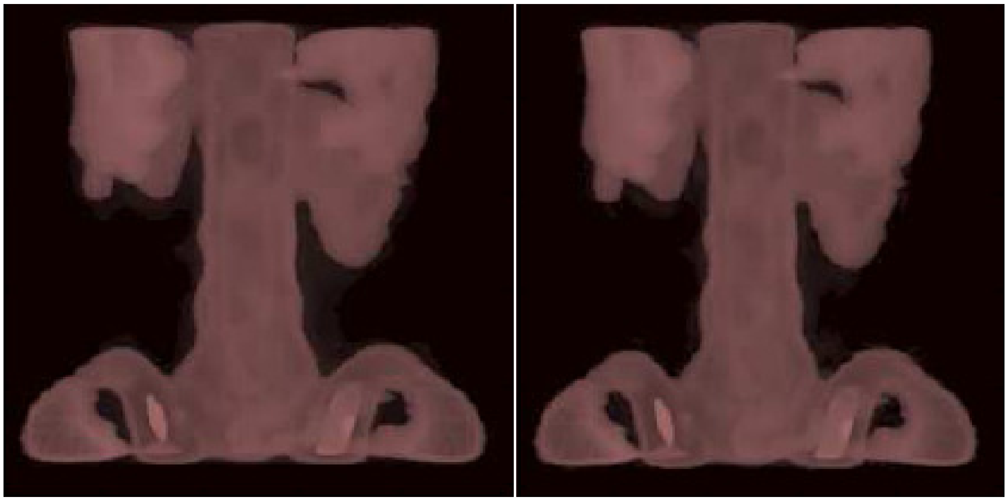


FIGURE 3(a) Original Image 3(b) Compressed Image by EZW

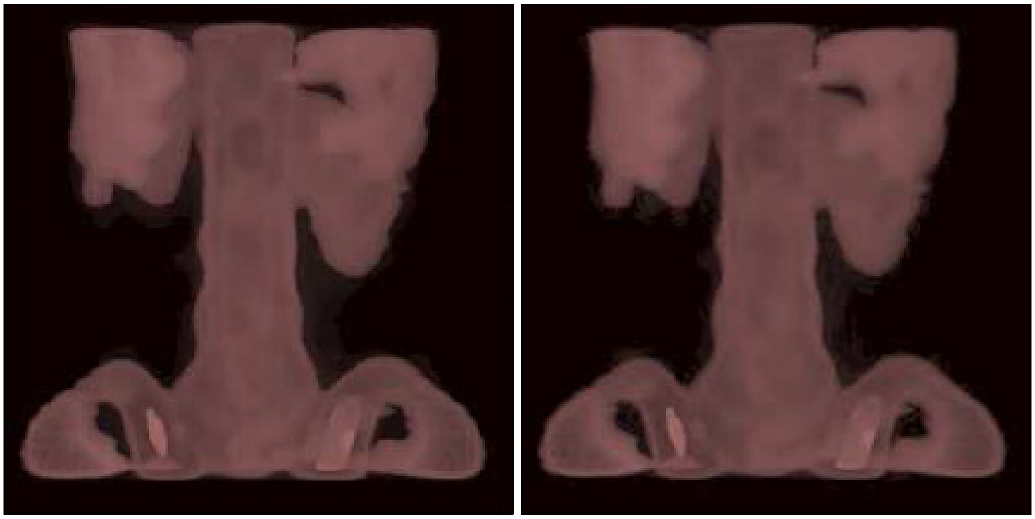


FIGURE 4(a) Original Image 4(b) Compressed Image by SPIHT

Image ---Spine, Size---2.37 KB, Entropy--4.102				
Wavelet	**CR**	**BPP**	**mse**	**psnr db**
	%		**Db**	
bior3.1	13.98	1.12	0.57	50.55
dmey	19.06	1.52	0.26	54.04
db8	19.13	1.53	0.26	53.98
sym5	17.91	1.43	0.27	53.89
coif2	17.91	1.43	0.47	51.46
rbio4.4	20.29	1.62	0.25	54.09

TABLE 1: Various Parameters Values of Different Wavelet for EZW Algorithm

Image ---Spine , Size---2.37 KB, Entropy--4.102				
Wavelet	**CR %**	**BPP**	**mse Db**	**psnr db**
bior3.1	8.73	0.7	0.61	50.29
dmey	7.85	0.63	0.42	51.94
db8	8.23	0.66	0.41	51.99
sym5	7.63	0.61	0.4	52.1
coif2	7.7	0.62	0.6	50.35
rbio4.4	8.77	0.7	0.38	52.32

TABLE 2: Various Parameters Values of Different Wavelet for SPIHT Algorithm

The graphical representation of CR, BPP, PSNR and MSE values are expressed as a bar graph are shown in Fig. 5, 6, 7 and Fig. 8. The main features of EZW include compact multiresolution representation of images by discrete wavelet transformation, zerotree coding of the significant wavelet coefficients providing compact binary maps, successive approximation quantization of the wavelet coefficients, adaptive multilevel arithmetic coding, and capability of meeting an exact target bit rate with corresponding rate distortion function (RDF) [7]. The SPIHT method provides highest image quality, progressive image transmission, fully embedded coded file, Simple quantization algorithm, fast coding/decoding, completely adaptive, lossless compression.

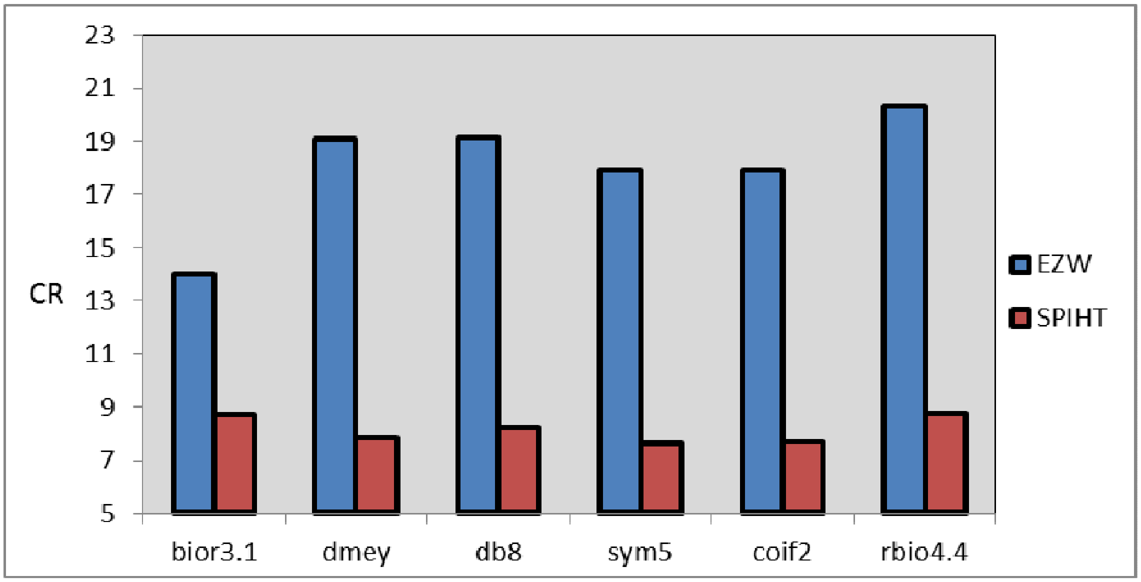


FIGURE 5 Comparison Chart of CR using EZW & SPIHT Algorithms.

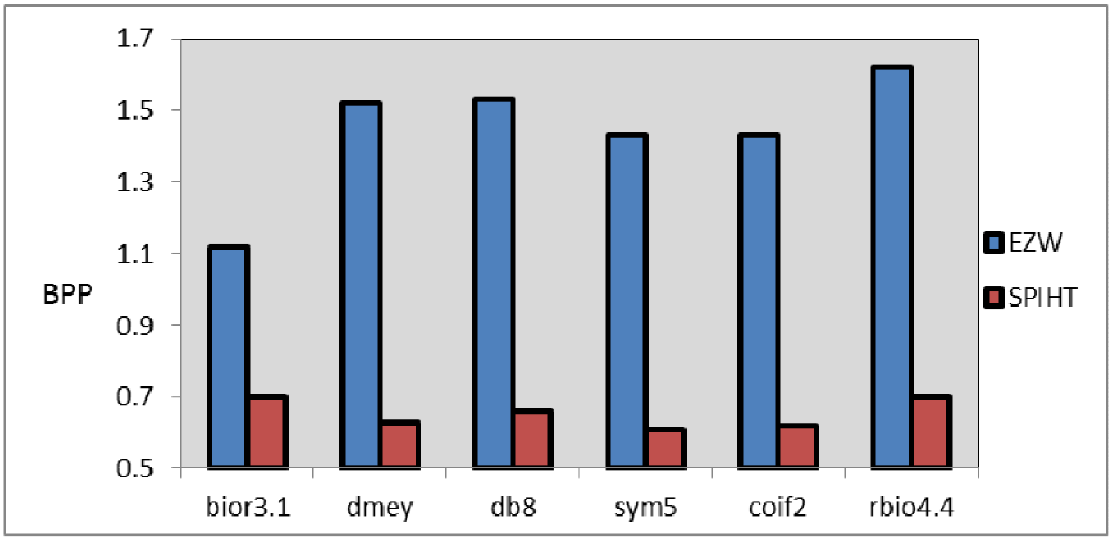


FIGURE 6 Comparison Chart of BPP using EZW & SPIHT Algorithms.

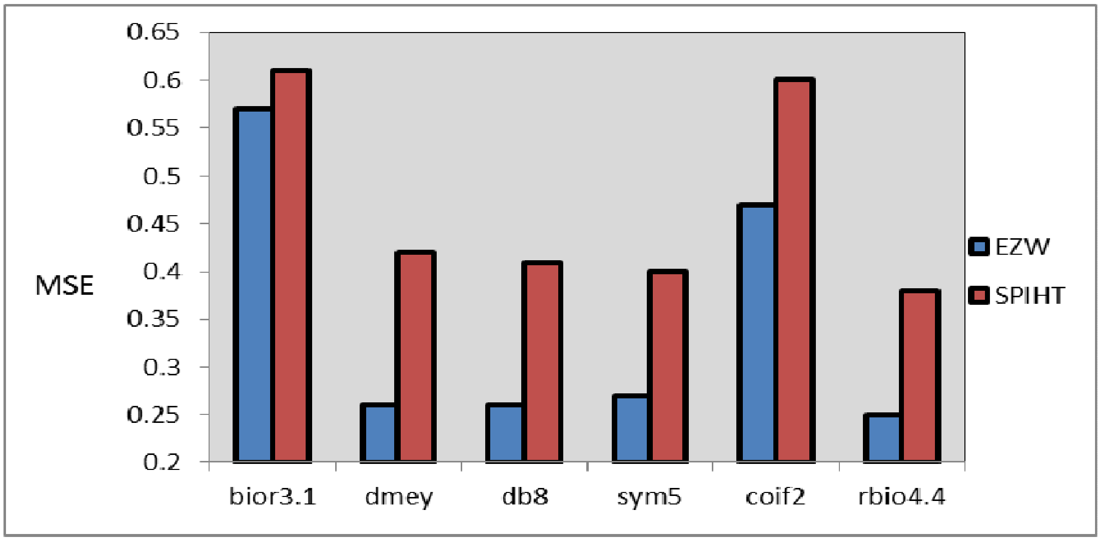


FIGURE 7 Comparison Chart of MSE using EZW & SPIHT Algorithms.

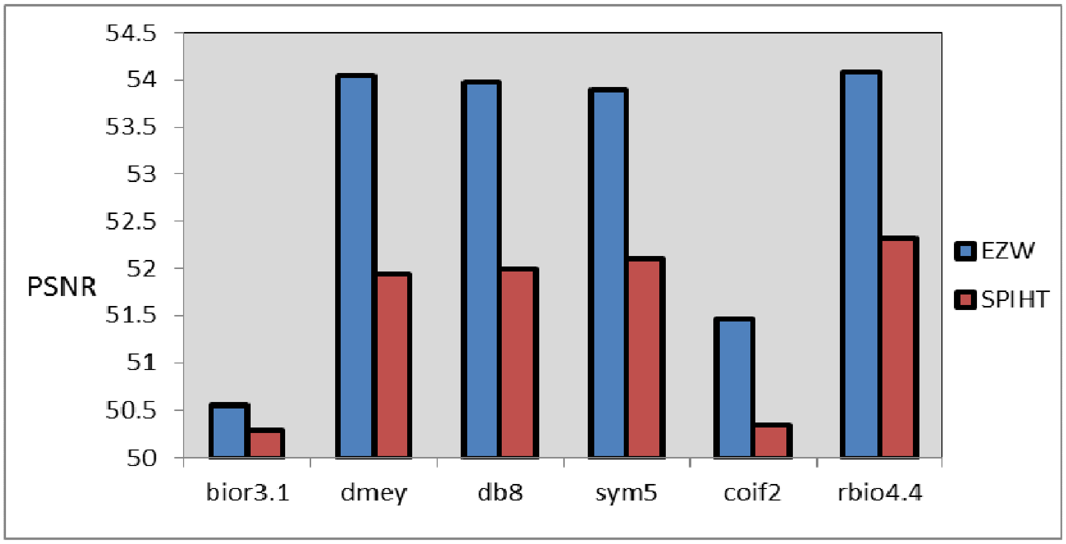


FIGURE 8 Comparison Chart of PSNR using EZW & SPIHT Algorithms.

7. CONCLUSION & FUTURE WORK

In this paper, the results of two different wavelet-based image compression techniques are compared. The effects of different wavelet functions filter orders, number of decompositions, image contents and compression ratios are examined. The results of the above techniques EZW and SPIHT are compared by using four parameters such as CR, BPP, PSNR and MSE values from the reconstructed image. These compression algorithms provide a better performance in picture quality at low bit rates. We found from our experimental results that SPIHT is better algorithm than EZW. Its results are 60-70% better than EZW as we can see from Table 1-2.We have analyzed that CR is reduced. BPP is also low comparative to EZW.MSE is low and PSNR is increased by a factor of 13-15%.as we can verify from our experiments. The above algorithms can be used to compress the image that is used in the web applications. Furthermore in future we can analyze different Image coding algorithms for improvement of different parameters.

8. REFERENCES

[1] R.Sudhakar, Ms R Karthiga, S.Jayaraman, “Image Compression using Coding of Wavelet Coefficients – A Survey”, ICGST-GVIP Journal, Volume (5), Issue (6), June 2005.

[2] J. M. Shapiro, “Embedded image coding using zero trees of wavelet Coefficients”, IEEE Trans. Signal Processing, vol. 41, pp. 3445- 3462, 1993.

[3] Basics of image compression - from DCT to Wavelets: a review.

[4] Bopardikar, Rao “Wavelet Transforms: Introduction to Theory and Applications.”

[5] T.Ramaprabha M Sc M Phil ,Dr M.Mohamed Sathik, “A Comparative Study of Improved Region Selection Process in Image Compression using SPIHT and WDR” International Journal of Latest Trends in Computing (E-ISSN: 2045-5364) Volume 1, Issue 2, December 2010

[6] Shamika M. Jog, and S. D. Lokhande, “Embedded Zero-Tree Wavelet (EZW) Image CODEC” ICAC3’09, January 23–24, 2009, Mumbai, Maharashtra, India.

[7] S.P.Raja, A. Suruliandi “Performance Evaluation on EZW & WDR Image Compression Techniques”, IEEE Trans on ICCCCT, 2010.

[8] Loujian yong, Linjiang, Du xuewen “Application of Multilevel 2- D wavelet Transform in Image Compression”. IEEE Trans on 978-1-4244-3291-2, 2008.

[9] Javed Akhtar, Dr Muhammad Younus Javed “Image Compression With Different Types of Wavelets” IEEE Trans on Emerging Technologies, Pakistan, Nov 2006.

[10] Rafael C. Gonzalez and Richard E. Woods, “Digital Image Processing”, 2nd Edition, Prentice Hall Inc, 2002.

[11] Khalid Sayood, “Introduction to Data Compression”, 3rd Edition 2009

[12] G. Sadashivappa, K.V.S. Ananda Babu, “WAVELET FILTERS FOR IMAGE COMPRESSION, AN ANALYTICAL STUDY” ICGST-GVIP journal, volume (9), Issue (5), September 2009, ISSN: 1687-398X

[13] Lou jian yong,Lin jiang and Du xuewen “Application of Multilevel 2-D wavelet Transform in Image Compression”IEEE Trans on Signal Processing 978-1-4244-3291-2, 2008.

Hierarchical Coordination for Data Gathering (HCDG) in Wireless Sensor Networks

Manal AL-Bzoor manal.al-bzoor@uconn.edu
Department of Computer Science & Engineering
University of Connecticut
Storrs, CT, 06269 USA

Laiali Almazaydeh lalmazay@bridgeport.edu
Department of Computer Science
University of Bridgeport
Bridgeport, CT 06604 USA.

Syed Rizvi srizvi@ecpi.edu
Electronics Engineering Technology Department
ECPI University
Virginia Beach, VA 23462, USA.

Abstract

A wireless sensor network (WSN) consists of large number of sensor nodes where each node operates by a finite battery for sensing, computing, and performing wireless communication tasks. Energy aware routing and MAC protocols were proposed to prolong the lifetime of WSNs. MAC protocols reduce energy consumption by putting the nodes into sleep mode for a relatively longer period of time; thereby minimizing collisions and idle listening time. On the other hand, efficient energy aware routing is achieved by finding the best path from the sensor nodes to the Base Station (BS) where energy consumption is minimal. In almost all solutions there is always a tradeoff between power consumption and delay reduction. This paper presents an improved hierarchical coordination for data gathering (HCDG) routing schema for WSNs based on multi-level chains formation with data aggregation. Also, this paper provides an analytical model for energy consumption in WSN to compare the performance of our proposed HCDG schema with the near optimal energy reduction methodology, PEGASIS. Our results demonstrate that the proposed routing schema provides relatively lower energy consumption with minimum delay for large scale WSNs.

Keywords: Energy Consumption, MAC Routing Protocols, Sensor Nodes, Wireless Sensor Network.

1. INTRODUCTION

There is a tremendous increase in the usage of wireless sensor networks (WSNs) for sensing and monitoring applications in the natural environment, industry, and military domains [1]. These networks usually consist of many low-power, low-energy, and low-cost sensor nodes with wireless communication links. The sensor nodes sense data from the nearby environment, receive data from other nodes , process the data, and send necessary data to other nodes or to the base station (BS) [2][3]. These networks are typically deployed in an Ad hoc manner where the participating nodes in a network share the same communication medium.

The sensor nodes are usually operated by batteries and left unattended after their deployment. This makes power saving scheme as one of the critical issues in WSNs as network should be considered to have a certain lifetime during which nodes should have sufficient energy for gathering, processing, and transmitting the information. Therefore, any protocol developed for sensor nodes communication should be designed to be extremely energy-efficient. The design of an energy-efficient protocol is an imminent problem to solve in WSNs [4].

WSNs usually consist of hundreds or even thousands of sensor nodes which may be sparsely distributed in non predefined remote locations. Thus, it becomes extremely difficult and computationally infeasible to recharge or replace the dead batteries of the network nodes. When sensor nodes in a WSN run out of energy they stop functioning as either data originators or data routers, causing a progressive deconstruction of the network. Therefore, one of the most stringent limitations that the development of a WSN faces today is the power consumption issues. In reality, a sensor node typically consumes the most of its energy during communication with the other nodes. However, lower energy expenditure takes place while performing sensing and data processing [5]. As a result, there is a great development of techniques recently requiring the elimination of energy inefficiencies at all layers of the protocol stack of sensor nodes.

More precisely, research on physical and data link layers of the protocol stack has been focused on system level energy awareness such as dynamic voltage scaling, radio communication hardware, low duty cycle issues, system partitioning, and energy aware MAC protocols [6]. At the network layer of protocol stack, the main objective is to setup the best energy-aware route from the sensor nodes to the BS to prolong the overall network lifetime. For these reasons, while routing protocols in traditional networks aim to accomplish a high quality of service, routing protocols in WSN are more concerned towards power consumption issues.

The routing protocols developed for WSNs are classified mainly as flat routing and hierarchical or cluster- based routing protocols [7] [8]. In the former, each node plays the same role (i.e., all active sensor nodes collaborate with each other to perform the sensing task). In the latter approach, however, sensor nodes are divided based on their geographical location and programmed to perform a different role with respect to their energy consumption. In this paper, we propose a hierarchical chain-based schema that introduces a new method for reducing the energy consumption. Our proposed HCDG scheme reduces the total energy consumption and provides relatively lower delay than the other hierarchical-based routing schemas such as LEACH [9] and PEGASIS [10].

The remainder of the paper is organized as follows: Section 2 provides an overview of the existing energy aware routing and MAC protocols for WSNs. In Section 3, we present our proposed HCDG routing schema. Section 4 provides analytical and simulation models for the proposed method to compare the performance with the PEGASIS and LEACH schemas. Finally, Section 5 concludes the paper with future work.

2. RELATED WORK

Energy aware routing is one of the hot research areas in WSNs. In general, routing protocols for WSNs can be classified according to their network structure as flat and hierarchical or location-based routing protocols. Specifically, routing protocols are classified into multipath-based, query-based, negotiation-based, quality of service (QoS)-based, and coherent-based routing protocols [2]. In flat networks, all nodes play the same role (i.e., each participating node aggregates data). In hierarchical protocols, nodes are divided into clusters where each cluster has one head node who is responsible to perform data aggregation. Since only head nodes can perform data aggregation, this reduces the energy consumption. Location-based protocols utilize position information to relay the data to the desired regions rather than the whole network [11]. For our proposed work, we use both hierarchical routing and location-based categories as a network structure.

Heinzelman et.al [9] introduced a hierarchical clustering algorithm for sensor networks, called Low Energy Adaptive Cluster – based protocol (LEACH). In LEACH the operation is divided into rounds. During each round, a set of nodes are selected as cluster–head nodes. Once selected, these cluster-head nodes cannot become cluster heads again for the next *P* rounds. Thereafter, each node has a 1/*p* probability of becoming a cluster head in each round. At the end of each round, each node which is not a cluster head selects the closest cluster head and joins that cluster to transmit data. In addition, cluster heads aggregate and compress the data and forward it to the BS. In this algorithm, the energy consumption distributes uniformly among all nodes whereas non–

head nodes turn off as much as possible. LEACH assumes that all nodes are in wireless transmission range of the BS which is not the case in many sensor nodes deployment algorithms. In each round, cluster heads comprise 5% of total nodes and use TDMA as a scheduling mechanism that makes it prone to long delays when applied to a large sensor network.

In [10] an enhancement over LEACH protocol was proposed. The protocol, called Power – Efficient Gathering in Sensor Information Systems (PEGASIS) a near optimal chain-based protocol for extending the lifetime of network. In PEGASIS, each node communicates with one of the closest neighbors by adjusting its signal power such that it can only be heard by the closest neighbor. Each node uses signal strength to measure the distance between its current location and the neighboring nodes to determine the node which is at the shortest possible distance. After chain formation, PEGASIS elects one of the nodes as a leader from the chain with respect to residual energy usage. Unlike LEACH [9], PEGASIS [10] avoids cluster formation and uses only one node in a chain to transmit the data to the BS rather than multiple nodes. This results in relatively lower overhead and the bandwidth requirements from the BS.

In COSEN [12], a chain oriented sensor network for collecting information was introduced where multiple lower chains are formulated exactly in the same manner as described in PEGASIS [10]. Each chain starts from the furthest node that includes a certain percentage of total nodes where the number of leaders equal to the number of formulated chains. Each leader from each chain collects and aggregates the data from its chain level and transmits this aggregated data to the higher level leader until it reaches to the BS. Introducing this hierarchical chain model in COSEN alleviated parallel data aggregation and hence achieved higher reduction in both energy and delay compared to PEGASIS and LEACH.

In [13], a new routing algorithm based on chaining structure was proposed. It was based on the same idea of chain formation as suggested by PEGASIS. However, it uses different criteria for selecting the next node in the chain formation process. PEGASIS adds the next node to the chain as the node closer to the last node in the chain. However, this method uses the distance between the next node and rest of the nodes that are currently part of the chain as criteria for selecting the next node. This new method of selecting the next node ensures that the total distance from any selected leader to other nodes in the chain is minimal and therefore offers relatively lower energy consumption than the original PEGASIS. Simulation results [13] show that this proposed method can reduce the total energy consumption more than the best traditional algorithms such as PEGASIS and LEACH with a factor of 34%.

Our proposed routing scheme differs from the existing solutions since we combine hierarchical chaining method for chain formation and selecting the next node based on the total distance to all other chain members. Our proposed method lowers the burden on the chain-leader by introducing a coordinator node who is responsible for collecting the data from the lower level chains and forwarding it to the leader node. Our proposed scheme makes parallel data gathering more feasible and thus provides relatively lower end-to-end delay than the other routing schemas that use the same hierarchical structures.

3. HIERARCHICAL COORDINATION AND DATA GATHERING SCHEME

One of the main objectives of the proposed scheme is to minimize both energy consumption and end-to-end delay which is required for data gathering in WSNs. Our proposed scheme is based on the same assumptions as described in [9] [10] [12]. Before we present the proposed scheme, it is worth mentioning some of our key assumptions.

1. We assume that the BS is located in a fixed place with a field of nodes deployed randomly where all nodes are considered to be stationary.

2. We assume that all sensor nodes encapsulate complete information about the network and each of them is able to adjust its transmission power such that it can only be heard by its closest neighbor.
3. We also assume that each node is capable to perform data aggregation from other nodes with its own data into a single packet.

4. Finally, we assume that sensor nodes and BS are homogeneous and have limited energy.

Our proposed HCDG scheme differs from [12] in both chain formation strategy and in proposing two role based coordination for each chain in the hierarchy. Our proposed schema, therefore, consists of three main phases: chain hierarchy formation, coordinators and leaders groups' selection phase, and data transmission phase.

3.1. Chain Hierarchy Formation

In this first phase of our proposed scheme, we use the next node selection criteria proposed in [13] and combined with [12] for hierarchical chain formation. In order to form the hierarchical chain, we start from the furthest node from the BS as illustrated in Algorithm 1. Next, we select the node which has the closest distance to the rest of nodes that are already exist in the chain. The chain formation reaches to its end once a certain percentage of total number of nodes in the field becomes members of that chain. We refer to this condition as chain saturation which indicates that a maximum number of nodes have associated with the chain and there is no need for extending the chain formation process. In other words, this percentage limits the number of

```
Start from furthest node to the BS

/*initialization phase for chain and member IDs*/

S1      CID= 0;      /*chain id*/
        MID=0;    /*member id initialization*/
        P=Percentage; /*node percentage for each chain*/
S2      Currentlocation= CID.member[MID].location

S3      while  (RemainingNodeCount>0) do:
S4          while (MID < P*TotalNodeCount) do:
S5              for (i=0 to RemainingNodesCount)
S6                  for     (j=0 to MID);
S7                          Totaldistance=Node[i].loc -CID.Member[j].loc
                    end for;
S8                              tmpDistance = Totaldistance/MID+1;
S9                      If      (tmpdistance < mindistanec) then
S10                             mindistance = tmpdistance;
S11                             CID.member[MID+1]= node[i];
                        end if
S12                 MID++;
S13                 RemainingNodesCount--;
S14                 Mindistance=Maxdistance;
                end for
            end while
S15             CurrentLocation=CID.member[MID].location
S16             CID++; MID=0;
        end while
```

Algorithm 1: Chain Hierarchy Formation

chains in the hierarchy. For instance, 20 percent will produce 5 chains. The percentage of nodes in each chain should be proportional to the number of nodes in concentric distances from the BS. Very long chains result in more delays and more total energy consumption and the network operation will resemble that of PEGASIS. Shorter chains also result in longer delays but only for nodes farthest from the BS. The effect of the number of chains can be clearly seen in the equation presented in Section 4 of this paper. The chain hierarchy formation is done once setting up the network or when a certain percentage of nodes die.

3.2. Leader and Coordinator Candidates Selection Phase

In this second phase of our proposed scheme, members in each chain are divided evenly into two groups: leader candidates and coordinator candidates.

Leader Candidates: Leaders candidates are refer to those nodes from which a chain leader will be elected and is responsible for the following two tasks:

- Chain leader's first responsibility is to gather data from neighboring nodes that exist in the same chain.
- Chain leader's second responsibility is to transmit the aggregated data to the higher level chain coordinator or to the BS.

This group (i.e., the leader) will be selected by the members of the chain such that the selected leader should be at the closest distance to the BS or to the coordinator of the higher level chain.

Coordinator Candidates: In our proposed scheme, coordinator candidates refer to a group of nodes where a coordinator node will be elected and is responsible for collecting the aggregated data from each leader of the lower chain. Moreover, coordinators for the first chain are elected from those nodes that are at the furthest distance from the BS. Similarly, the coordinators for the lower level chains are elected from nodes that are at the furthest distance from the leader of the higher level chain.

Fig. 1 is an illustration of group selection after chain formation phase. The black color nodes indicate group of leader's candidates whereas gray color nodes represent coordinator candidates group. In addition, white color nodes indicate a selected coordinator in a certain round for each chain. Starting from Chain 0, black nodes are selected as leaders since they have minimal distance to the BS. The white node in chain 0 is the elected coordinator. Chain 1 calculates the distance from the coordinator of Chain 0 and selects the coordinators candidate group and the lead-

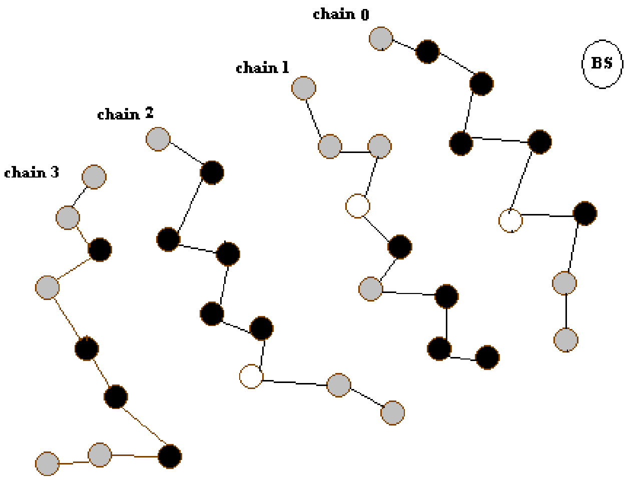


FIGURE1: Four chains sensor network deployment.

er's candidate group. Once the group selection is made, each chain coordinator keeps acting as a point of reference for lower chains to select candidate members for each group.

3.3. Data Transmission Phase

In this second phase of our proposed scheme, each node is assumed to have data available to be sent to the BS in a timely basis. In addition, each chain selects one leader and one coordinator based on the residual energy. Each sensor node will be informed by the location of the leader node using a token which is passed by the leader of the chain to all of its neighboring nodes. Nodes start receiving and sending the aggregated data packets in the direction of leader. Leader of each chain collects the data and send it to the coordinator of the higher chain.

3.4. Fairness, Energy Consumption, and Delay Reduction in HCDG

Groups of coordinators and leaders nodes are selected starting from the highest level chain. For each round, one leader and one coordinator node is selected from those groups according to the residual energy. For the lower level chains, groups are selected after every round whenever a new coordinator is selected in the hierarchy. As mentioned earlier, the higher level hierarchy changes typically after every round and imposes more processing for the nodes in lower level chains. However, this additional processing at lower level chains results in more fairness for the higher level chain nodes which performs more processing for data aggregation and direct communication with the BS.

The next node selection criteria for each chain will ensure total minimum distance between nodes. In the second phase, if the leader is comparatively at larger distance from the BS, it requires the leader to adjust its transmission to maximum power in order to reach the BS and transmit the aggregated data. The transmission at maximum power makes this node deplete energy faster than a closer leader even if it starts transmitting with comparatively higher energy. The above reason leads us to choose only those nodes as leader(s) that are closest to the BS in first chain. Similarly, in higher level chains, we choose leaders that are closest to the coordinator node. Another additional source of energy reduction in our work comes from the fact that the data gathering processing will be divided between the two nodes (i.e., the leader and the coordinator). The combination of leader and coordinator in our proposed scheme brings a degree of parallelism since both perform data gathering together at different levels of chain. For instance, a leader will start gathering its data from one side of its neighbors while the coordinator in the other side is collecting the data from the lower level. Our proposed scheme, therefore, yields comparatively lower delays than the other hierarchical routing schema such as PEGASIS [10].

4 ANALYTICAL MODEL FOR HCDG SCHEME

Firstly, in this section we present an analytical model to approximate the energy consumption for WSN. Secondly, we provide our critical analysis to analyze the performance of the proposed scheme with the other well known schemes. To support our analytical model, several numerical results will be presented in this section.

4.1. Energy Model

For the sake of analytical model, we use the same radio model as described in [10] [12] to compare the performance of proposed schema with the PEGASIS [10]. This model corresponds to the first order energy model where the parameters and values are used to evaluate the performance of all hierarchical routing schemas in WSNs. Table 1 shows the energy parameters and their corresponding values use for analytical model and performance evaluation. We use E_{Elec} as an energy consumption coefficient for the wireless transmission of a single bit whereas the parameter *k* represents the number of data bits to be transferred or received (i.e., the aggregated data packet bits). ε_{Amp} denotes the total energy required to amplify a single bit of a transmitted signal over the wireless medium. Finally, E_{Agg} indicates the combined amount of energy consumed for aggregating a nodes data packet with the received data packets.

Type	Parameter	Value
Transmitter Electronics	E_{Elec}	50nJ
Transmitt Amplifier	ε_{Amp}	100pJ/bit/ m^2
Aggregated Data Packet	K	2000 bit
Aggregation Energy	E_{Agg}	5nJ

TABLE 1: System Parameters Definition and Standard Values

Taking the above parameters into consideration, the transmission and reception energy consumption for each sensor node can be approximated as.

$$\begin{aligned} E_{T_x(k,d)} &= E_{T_x}(k) + E_{T_x\_amp}(k,d) \\ E_{T_x(k,d)} &= (E_{Elec}\times k) + (\varepsilon_{Amp}\times k\times d^2) \end{aligned} \tag{1}$$

$$E_{R_x}(k,d) = E_{R_{x\_Elec}}(k) \cong E_{Elec}\times k \tag{2}$$

In both (1) and (2), E_{T_x} represents the total amount of energy used by a node to transmit the data where the subscript *d* represents the distance between the source and the target nodes. Moreover, E_{R_x} in (1) and (2) represents the total energy consumed by a single node to receive *k* bits of a data packet.

4.2. Energy Consumption Comparison

In PEGASIS [10], all nodes are arranged in one chain and only one node is selected as a head of the chain. The head node is responsible for aggregating the data from all neighboring nodes and transmitting it to the BS. We compare energy consumption for the three modes of operations with *N* nodes in both PEGASIS and our proposed HCDG Schema.

Energy for Transmission: In PEGASIS, total energy consumption for all nodes can be approximated as follows:

$$E = (N\times E_{Elec}\times k) + (\varepsilon_{Amp}\times k\times)\left[\sum_{m=1}^{N}\langle d_{m-1,m}\rangle^{2}\right] \tag{3}$$

In our proposed HCDG schema for *N* nodes with CN chains, we have $n = N/CN$ nodes per chain. All nodes except the leader in each chain transmits the data to its closest neighboring node with total energy equal to the total energy per chain multiplied by the number of chains. This can be formularized as

$$E = CN\times E_{CH} \tag{4}$$

Further elaborating (4) results

$$E = CN\left[\left(n \times E_{Elec} \times k\right) + \left(\varepsilon_{Amp} \times k\right)\sum_{m=1}^{n}\left\langle d_{(m-1,m)}\right\rangle^{2}\right] \tag{5}$$

Comparing (3) with (5), we can observe that they are equal if and only if $d_{(i,j)}$ is minimal in both. However, the selection criteria taken in our method is proved in [13] to produce smaller distances between nodes.

Energy Consumption for Receiving Data: In PEGASIS, each node receives data if it is an intermediate node. Based on that, the energy consumed by each receiving node can be approximated as follows:

$$E = \left(N - 1\right) \times E_{Elec} \times k \tag{6}$$

In our proposed HCDG Schema, worst scenario is the same as in PEGASIS equation (6) where the last node in each chain is the leader for that chain and the first node in the next chain is the coordinator of that chain which makes our schema looks like a one chain schema.

For best case scenario, when the leader and the coordinator nodes are not the last or first nodes in the chain, the total energy consumed by each chain for receiving the data can be approximated as follows:

$$E_{CH} = \left(\frac{N}{CN}\right)\left(E_{Elec} \times k\right) \tag{7}$$

The last chain will have only $\frac{N}{CN} - 1$ *number of* received packets since there is no data to be received from lower chains. Taking this into consideration, the total energy for all chains can be approximated as:

$$\begin{aligned} E &= \left(CN - 1\right)\left(\frac{N}{CN}\right)\left(E_{Elec}k\right) + \left(\frac{N}{CN} - 1\right)\left(E_{Elec} \times k\right) \\ E &= \left(E_{Elec} \times k\right)\left\langle N\left(\frac{CN-1}{CN}\right) + \left(\frac{N-CN}{CN}\right)\right\rangle \\ E &= \frac{N}{CN}\left(E_{Elec} \times k\right)\left\langle N - 1\right\rangle \end{aligned} \tag{8}$$

Equation (8) is identically approximated as (6). From the above approximations, one can conclude that the energy consumed for receiving aggregated packets is the same as it is consumed in PEGASIS scheme.

Energy Consumption for Data Aggregation: In PEGASIS, for the best case scenario, all nodes perform data aggregation except the leaf nodes. Based on this, the total energy consumption can be approximated as:

$$E = \left(N - 2\right) \times E_{Agg} \tag{9}$$

On the other hand, in our proposed HCDG schema, all nodes in each chain perform data aggregation except the leaf nodes. Taking this into consideration, one can approximate the total energy consumption for each chain as follows:

$$E_{CH} = \left({}^{N}\!/_{CN} - 2 \right) \times E_{Agg} \tag{10}$$

Based on (10), total energy consumed by the proposed HCDG schema for data aggregation can be approximated as follows:

$$\begin{aligned} E &= CN \times E_{CH} \\ E &= CN \times \left[\left({}^{N}\!/_{CN} - 2 \right) \times E_{Agg} \right] \\ E &= \left(N - 2 \times CN \right) E_{Agg} \end{aligned} \tag{11}$$

Comparing (9) and (11), one can observe that the proposed HCDG schema yields the lower total consumption in data aggregation operation compared to what is consumed by the PEGASIS scheme. Table 2 and Fig. 2 show a comparison of time consumed between the PEGASIS and the proposed HCDG scheme in data aggregating for one round of transmission in all nodes. The power consumption is measured in nano-joules (*nJ*) for both PEGASIS and the proposed HCDG routing schema.

Energy Consumption When Transmitting to BS: In PEGASIS, all nodes in the chain takes turn to transmit the aggregated data to the BS. Based on that, one can approximate the energy consumption as follows:

$$E = \left(E_{Elec} \times k \right) + \left\langle \varepsilon_{Amp} \times k \times d^2 \right\rangle_{(i,\,BS)} \tag{12}$$

In each round of transmission, the distance between the BS and the head node varies substantially. Consequently, the total energy consumption for multiple rounds increases by increasing the distance and the elected furthest head will consume its energy faster than the other nodes.

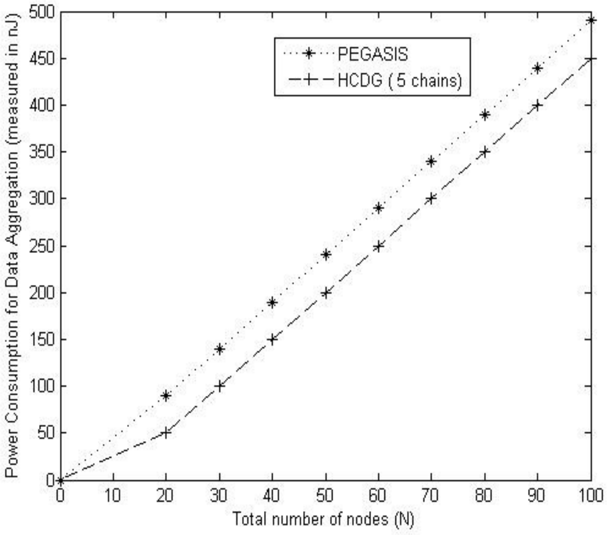


FIGURE 2: An illustration of total power consumption for data aggregation versus the total number of nodes (*N*).

	Power Consumption for Data Aggregation	
Total Nodes (N)	PEGASIS	HCDG (5 chains)
20	90nJ	50nJ
40	190nJ	150nJ
60	290nJ	250nJ
80	390nJ	350nJ
100	490nJ	450nJ

TABLE 2: Total Power Consumption for Data Aggregation versus Total Number of Nodes

On the other hand, in our proposed HCDG schema, only half of the nodes that exist in the closest chain to the BS are allowed to transmit the data. This hypothesis can be used to approximate the limits of both proposed HCDG and PEGASIS schemas for multiple rounds.

$$\begin{aligned} &Avg\left(d^2{}_{(i,\,BS)}\right)HCDG < \\ &Avg\left(d^2{}_{(i,\,BS)}\right)PEGASIS \end{aligned} \tag{13}$$

From all the above equations, we showed that our schema outperform PEGASIS in energy reduction for data transmission between nodes, data aggregation, and data transmission to the BS.

4.3. Delay Reduction

This section analyzes the best and worst cases delays performance of the proposed schema. For the sake of the performance evaluation and experimental verifications, we use the TDMA based scheduling with the proposed HCDG scheme.

Let *t* is the time unit required to transmit the data from one node to its immediate neighboring node. In PEGASIS, the worst case scenario is to have the head node as the last or the first node of the chain where the data will be sent to all *N* number of active nodes in order to reach the BS. Based on this argument, the total delay can be approximated as:

$$Delay = N \times t \tag{14}$$

On the other hand, the best case scenario is when we have the head node positioned exactly in the middle of the chain so that the data from both sides could be gathered in a parallel manner which results in a best case delay. If *N* is odd, the aggregated data from both sides arrives at the same time to the head. This implies that the head node needs to defer receiving the data from one side by a factor of one time unit. On the other hand, if *N* is even, the data from the longer side arrives in one time unit later than the shorter side. The head node adds another time unit to send to the BS. Based on the above argument, one can approximate the best case delay as follows:

$$Delay = \begin{cases} \left(\dfrac{(N-1)}{2} + 2\right) \times t \rightarrow if\ \ N\ \ is\ \ odd \\ \left(\dfrac{N}{2} + 1\right) \times t \rightarrow if\ \ N\ \ is\ \ even \end{cases} \tag{15}$$

In proposed HCDG schema, we use multiple chains that can be formalized as: *n =N/CN*. The worst case delay scenario is when the first node of the chain acts as the coordinator where as the last node acts as the leader. This configuration makes our worst case delay scenario similar to what is described for the PEGASIS. However, the probability of having this worst case delay is extremely small due to the group selection criteria we have used with the proposed HCDG scheme.

For the best case scenario, the leader and the coordinator nodes are located in the middle of the chain where both of them are one node apart from each other. This configuration is true for each chain. Based on the above specification, delay for the lowest level chain which will have only one leader and coordinator node can be approximated as follows:

$$Delay=\begin{cases}\left(\dfrac{(n-1)}{2}+2\right)\times t\rightarrow if\ \ n\ is\ odd\\ \left(\dfrac{n}{2}+1\right)\times t\rightarrow if\ \ n\ is\ even\end{cases}\tag{16}$$

In the higher level chain, leader will keep busy in gathering the data from one side and will be waiting to receive the data from the coordinator side. Coordinator, on the other hand, waits to receive the data from the lower level chains. Once Coordinator node receives the data from the lower level chain, it needs one extra time unit to send it to the leader node. The leader node also needs one additional time unit to send it to the upper level chain. In this manner, each chain adds two time units to the delay incurred from the lowest level chain. The above arguments can be used to derive an approximation for the best case delay for both even and odd number of nodes.

$$Delay=\begin{pmatrix}2\times(CN-1)\\ +\left(\dfrac{n-1}{2}+2\right)\end{pmatrix}*t\rightarrow if\ \ n\ is\ odd\tag{17}$$

$$Delay=\begin{pmatrix}2\times(CN-1)\\ +\left(\dfrac{n}{2}+1\right)\end{pmatrix}*t\rightarrow if\ \ n\ is\ even\tag{18}$$

Both Table 3 and Fig. 3 demonstrate a comparison between the PEGASIS and the proposed HCDG schema for the best case delay scenario. For the sake of experimental verifications, different sizes of networks are used with number of chains (i.e., CN=5, *n* = N/CN). A significant de-

Total Nodes (N)	Lowest Delay PEGASIS	Lowest Delay HCDG (5 chains)
50	26 t	14t
100	51t	19t
200	101t	29t
300	151t	39t
400	201t	49t

TABLE 3: Delay Analysis versus Total Number of Nodes for PEGASIS and HCDG Schemas

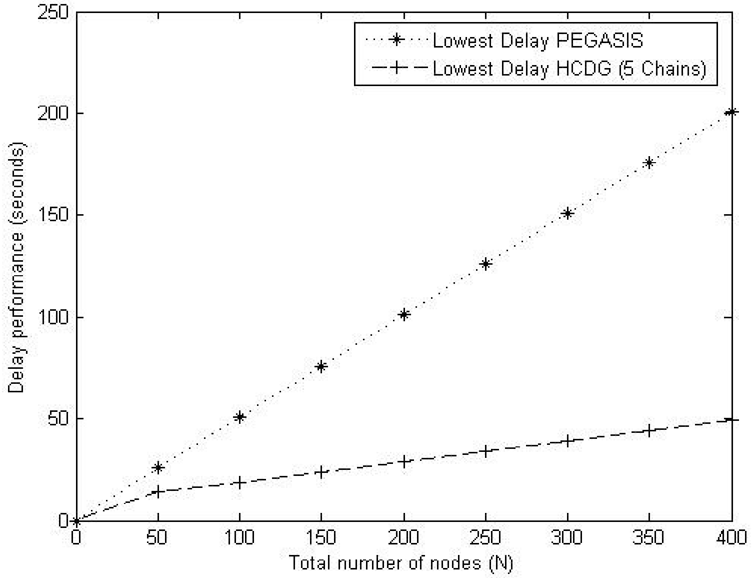


FIGURE 3: An illustration of delay performance versus the total number of nodes (*N*). The delay is measured in seconds for both PEGASIS and the proposed HCDG routing schema. A significant performance gain can be evidenced for the proposed HCDG scheme

lay reduction was obtained by using our proposed HCDG schema when compare to the PEGASIS for denser networks.

5. CONCLUSION

In this paper, we presented a new routing schema, hierarchical Coordination for Chain Based Data Gathering (HCDG) for WSN. The proposed HCDG schema introduced a new concept of leaders and coordinators nodes in a multichain hierarchical sensor network. In order to support the proposed HCDG schema, this paper provided a complete analytical model to approximate the energy consumption for wireless sensor nodes. Our numerical results demonstrate that the proposed HCDG schema can reduce energy consumption by a large magnitude when compared to the PEGASIS which was originally designed to outperform the well known LEACH method. Also, the analytical model and the results showed that the proposed HCDG schema substantially reduces the delay when compared to delay incurred in PEGASIS for denser WSN.

However, the numerical data which we have collected based on the proposed analytical model gives only a clue of the actual performance of the proposed HCDG schema. This is mainly due to the fact that the random generation of the wireless sensor nodes is hard to model using the mathematical equations presented in this paper. In future, we plan to design and conduct larger-scale experiments and simulation for better understanding of the energy consumption of the proposed HCDG schema and their correlations with different chain formation criteria and other design alternatives for selecting leaders and coordinators nodes.

6. REFERENCES

[1] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, “Next century challenges: scalable coordination in sensor networks,” MobiCOM, pp. 263-270, August 1999.

[2] A. Rai, S. Ale, S. Rizvi, and A. Riasat, “A new methodology for self localization in wireless sensor networks,” Proc of the 12th IEEE International Multitopic Conference, pp. 260 – 265, December 2008.

[3] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson,"Wireless sensor networks for habitat monitoring," Proc of WSNA, pp. 122-131, September 2002.

[4] C. Li, "Overview of wireless Sensor Networks," Journal of Computer Research and Development, vol. 42, no. 1, pp. 163-174, 2005.

[5] S. Rizvi and A. Riasat, "Use of self-adaptive methodology in wireless sensor networks for reducing energy consumption," Proc of IEEE International Conference on Information and Emerging Technologies, pp. 1 - 7, July 2007.

[6] K. Akkaya and M. Younis, "A Survey of Routing Protocols in Wireless Sensor Networks," Elsevier Ad Hoc Network Journal, vol. 3, no. 3, pp. 325-349, 2005.

[7] J. Al-Karaki and A. Kamal, "Routing techniques in wireless sensor networks: a survey," IEEE Wireless Communications, vol. 11, no. 6, pp.6-28, December 2004.

[8] M. Younis, M. Youssef, and K. Arisha, "Energy-aware routing in cluster-based sensor networks", Proc of the 10th IEEE/ACM(MASCOTS2002), Fort Worth, TX, October 2002.

[9] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocols for wireless microsensor networks", Proc of the 33rd Hawaii International Conference on System Sciences, January 2000.

[10] S. Lindsay and C. Raghavendra, "PEGASIS: Power-efficient gathering in sensor information systems", Proc. Of the IEEE, vol. 32, no. 4, pp. 66 - 71, 2003.

[11] B. Sarazin and S. Rizvi, "A Self-Deployment Obstacle Avoidance (SOA) Algorithm for Mobile Sensor Networks," International Journal of Computer Science and Security (IJCSS), Vol. 4, Issue. 3, pp. 316 - 330, 2010.

[12] N. Tabassum, Q. Mamun, and Y. Urano, "COSEN: A chain oriented sensor network for efficient data collection," Proc of the 3rd International Conference on Information Technology: New Generations (ITNG'06), pp. 7695-2497, April 2006.

[13] K. Khamforoosh and H. Khamforoush, "A new rounting algorithm for energy reduction in wireless sensor networks," Proc of 2nd IEEE International Conference on Computer Science and Information Technology, pp.505-509, 2009.

A Spatial Domain Image Steganography Technique Based on Matrix Embedding and Huffman Encoding

P.Nithyanandam nithyanandamp@ssn.edu.in
Department of Computer Application
SSN College of Engineering,
Anna University of Technology, Chennai
Kanchipuram Dt, Tamilnadu , 603110,India

T.Ravichandran *dr.t.ravichandran@gmail.com*
Principal
Hindustan Institute of Technology,
Anna University of Technology, Coimbatore
Coimbatore Dt,Tamilnadu, 641032,India

N.M.Santron nmsantron@gmail.com
III Year M.C.A.
Department of Computer Application
SSN College of Engineering,
Anna University of Technology, Chennai
Kanchipuram Dt, Tamilnadu , 603110,India

E.Priyadharshini indrapriyadharshini.e@gmail.com
III Year M.C.A.
Department of Computer Application
SSN College of Engineering,
Anna University of Technology, Chennai
Kanchipuram Dt, Tamilnadu , 603110,India

Abstract

This paper presents an algorithm in spatial domain which gives less distortion to the cover image during embedding process. Minimizing embedding impact and maximizing embedding capacity are the key factors of any steganography algorithm. Peak Signal to Noise Ratio (PSNR) is the familiar metric used in discriminating the distorted image (stego image) and cover image. Here matrix embedding technique is chosen to embed the secret image which is initially Huffman encoded. The Huffman encoded image is overlaid on the selected bits of all the channels of pixels of cover image through matrix embedding. As a result, the stego image is constructed with very less distortion when compared to the cover image ends up with higher PSNR value. A secret image which cannot be embedded in a normal LSB embedding technique can be overlaid in this proposed technique since the secret image is Huffman encoded. Experimental results for standard cover images, which obtained higher PSNR value during the operation is shown in this paper.

Keywords: Steganography, Imperceptibility, Payload, Stego Image, Least Significant Bit (LSB), Huffman Encoding, Matrix Embedding, Peak Signal to Noise Ratio (PSNR), Mean Square Error (MSE) and Discrete Wavelet Transformation (DWT).

1. INTRODUCTION

Steganography is the art of secret communication. It has apparent difference with cryptography; because cryptography hides information content whereas steganography hides information existence. Steganography is broadly classified in to spatial and frequency domain technique. Least Significant Bit (LSB) replacement, LSB matching, Matrix embedding and Pixel value differencing are some of the spatial domain techniques. Frequency domain techniques include Outguess, F5, JP Hide and Seek. Fundamentally, a steganography algorithm or embedding function can influence the cover work in three different ways, namely cover lookup, cover synthesis and cover modification. Naturally, changes of larger scale will be more obvious than changes of smaller scale. As a result, most steganographic schemes try to minimize the distortion on cover work. The location of changes is controlled by the selection rule [1]. There are three types of rule namely sequential, random and adaptive.

The primary goal of steganography is to design embedding function that should be statistically undetectable and capable of communicating large payloads. There exists a tradeoff between embedding capacity and proportion of distortion. There are many algorithms evolving to accomplish steganography goal in both spatial and frequency domain. Minimizing the embedding impact while constructing a stego image could be one of the ways; this may thwart in applying statistical analysis over a stego image. The notion of this paper is to apply one such embedding technique and to produce a less distorted cover image. Supporting a higher payload on a cover image depends upon embedding technique; but it also can be viewed in another direction of compressing the payload before overlaying. A lossless Huffman [2] [3] [4] [5] compression prior to overlaying results in fewer distortion in the cover image.

Cachin's [1] description of steganography security calls for the Kullback-Leibler distance which says, the probability distance between the cover and stego work to be as little as possible. In our technique it is achieved by minimizing the distortion between the cover and stego work. This will make it harder for the warden to detect embedding. The embedding procedure can encode the message bits in many ways. For example in LSB embedding the LSB is replaced to match the secret message bits. On average, one can embed, 2 bits per embedding change. It can be substantially improved if we adopt a clever embedding scheme. In particular, if the payload is shorter than the embedding capacity, one can influence the location of changes to encode more bits per change. Let us take a look at the following simple example. Say, we have a group of three pixels with gray scale values $x1$, $x2$ and $x3$. We wish to embed 2 message bits, $b1$ and $b2$. It seems that a practical approach might be to simply replace $b1$ with x1 and $b2$ with x2 (i.e.) replacing the LSB of the pixels to match the corresponding message bits. Assuming the 2 bits are 0 or 1 with equal probability, the expected number of changes to the whole group of pixels to embed both bits is 1. Therefore, we embed at embedding efficiency of 2 or 2 bits per change. However, it can be improved. Let us encode $b1$ = *LSB* ($x1$) XOR *LSB* ($x2$) and $b2$ = *LSB* ($x2$) XOR *LSB* ($x3$). If the values of the cover work satisfy both equations with equality, no embedding changes are required. If the first one is satisfied but not the second one, simply flip the LSB of $x3$. If the second one is satisfied but not the first one, flip the LSB of $x1$. If neither one is satisfied, flip LSB of $x2$. Because all four cases are equally likely with probability 1/4, the expected number of changes is 3/4, which is less than what we had earlier. This embedding technique is called *matrix embedding* [1] which is further extended and used in the proposed method.

Huffman compression is a variable length coding whose performance depends on the input image bit stream. The compression is directly proportional to smoothness of the image. Higher the smoothness and higher the redundancy will give good compression. Subjective and objective measures [6] are the two techniques existing to test the distortion of the processed image. Subjective measure is not reliable because human vision is a metric in assessing the distortion of the stego objects. Human vision may vary from person to person; hence this approach is not suitable. In objective measure, the mean square error (MSE) represents the cumulative squared error between the stego image and cover image. A lower figure of MSE conveys lower error/ distortion between the cover and stego image.

The equation of MSE to assess the stego and cover object is given by:

$$MSE=\frac{1}{m*n}\sum_{i-1}^{m}\sum_{j-1}^{n}(A_{ij}-B_{ij})^2 \qquad \text{...........[1]}$$

Whereas A_{ij} represents pixel in the cover image and B_{ij} represents pixel in the stego image; m, n represents the height and width of the image respectively. It is measured in constant and the unit is decibel (dB).

Peak Signal to Noise Ratio (PSNR) is a metric which calculate the distortion in decibels, between two images. Higher the PSNR indicates a better reconstructed or stego image. The PSNR is represented by the following equation:

$$PSNR=10*\log_{10}\frac{(Max)^2}{MSE} \qquad \text{..............[2]}$$

Where max denote maximum intensity of grayscale (255).PSNR is measured in decibels (dB).

2. RELATED WORK

Chang, C.C et al., [7] has proposed an image steganography technique which offer high embedding capacity and bring less distortion to the stego image. The embedding process embed bits of secret bit stream on the stego image pixels. Instead of replacing the LSB of every pixel, this method replaces the pixel intensity with similar value. The range of modifiable pixel value is higher in edge areas than smooth areas to maintain good perceptual excellence. Various bit embedding methods are followed; which are decided by the correlation between the actual pixel and the neighboring pixels. The neighboring pixels may be a pixel left, right, top or bottom to the actual pixels. The different schemes are two sided, three sided and four sided one. Two sided scheme take upper and left pixels, three side scheme take upper, left and right whereas four sided take upper, left, and right and bottom pixels. The embedding capacity and PSNR are inversely proportional to the sides taken into account.

Po-Yueh Chen et al., [8] proposed an image steganography scheme which fixes the limitation of steganography technique proposed in [7]. The limitation of [7] is falling of boundary problem which means the pixel which is located for embedding will become unused; since it exceeds the maximum intensity level which is greater than 255 (maximum gray scale intensity). Fewer bits are added even on such pixels which improve the embedding capacity without compromising PSNR in this technique.

A. Nag et al., [9] proposed a stenographic technique which is based on wavelet transformation on the images. Discrete Wavelet Transformation (DWT) converts the spatial domain of cover image into frequency domain. Huffman compression is applied for the stream of secret bits before overlaying them on the cover image. A high PSNR and very high embedding capacity is achieved.

R.Amirtharajan et al., [10] proposed a stenographic technique which is based on LSB replacement technique. Varying lengths of secret bits get embedded in every pixel. In method1 green and blue are embedding channels keeping red channel as indicator channel. In method2 an option is provided for choosing the indicator channel among the three channels. Once chosen, the remaining two channel act as embedding channel. In method3 the indicator channel is chosen by rotation scheme across all the pixels. In the first pixel red channel is indicator; green channel is the indicator in second pixel and in third channel blue act as indicator. Once indicator is finalized the remaining two channels will be used for embedding. This scheme is repeated for the consecutive pixels. The MSE and PSNR is calculated for all channel and the average number of bits get embedded in every pixel is shown in their results.

The rest of the paper is organized as follows. Section III discusses the proposed steganography technique. In Section IV experimental results are exhibited and discussed. Finally the conclusion and future direction are provided for the proposed work.

3. PROPOSED METHOD

3.1. System Architecture

Fig.1 shows the overall system architecture on which the proposed study stands on. The secret image pixel values are Huffman compressed which comprises of Huffman encodings and Huffman table. The size of Huffman table and Huffman encodings are measured in a 32 bit quantity each. These 64 bits are recorded across the last 64 byte's LSB of the stego image. Both the Huffman encodings and Huffman table binary content are embedded in the LSB of every byte using LSB replacement or Matrix embedding technique. The binary content of Huffman table is followed by Huffman encodings. The starting and the ending point of the corresponding binary component i.e. Huffman encodings or Huffman table is identified through the processed individual 32 bits entry stored at the end of the stego image. In the case of the secret image being sufficiently large, the stego image LSB may be fully utilized. Always, the last 64 byte is reserved for storing the size of Huffman table and Huffman encodings.

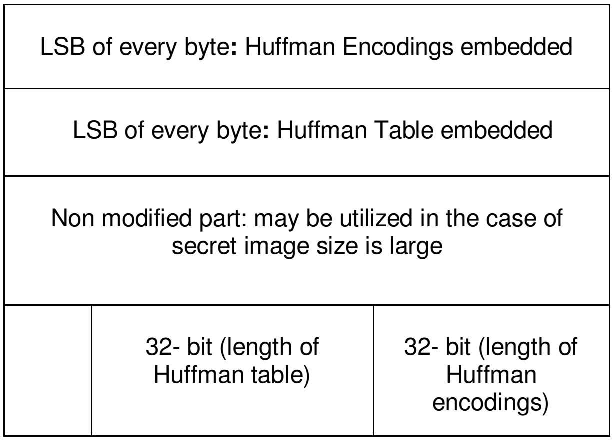

LSB of every byte: Huffman Encodings embedded
LSB of every byte: Huffman Table embedded
Non modified part: may be utilized in the case of secret image size is large
32- bit (length of Huffman table)
32- bit (length of Huffman encodings)

FIGURE 1: Stego Image Architecture

3.2. Huffman Compression on Image

The intensity of the pixels across the image is spatially correlated [3]. Information is pointlessly repeated in the representation of the correlated pixels. These repetitive pixels should also be represented by fixed number of bits in unencoded Huffman format. Actually these values are the best source for exploiting compression. A very frequent occurrence intensity value can be represented by variable numbers of bits (i.e. shorter bits) in contrast to the fixed number of bits for representing the pixel intensity used in unencoded Huffman technique. This is the core concept of Huffman encoding technique. The secret image is Huffman encoded prior to embedding process.

3.3. Extended Matrix Embedding

An extended matrix embedding technique is used in proposed method. Generally (1, n, k) matrix embedding [11] mechanism is used; which denotes k secret bits are embedded in n cover bits with at most 1 change. Here using three Least Significant Bits of RGB channel 2 bits of secret bits might be embedded with at most one change, which is typically (1,3,2) in the above case. Here n is 2^k-1.

It can be further expanded by considering; more secret bits can be embedded in a single go with at most 1 change. For example if k is 3, then n is 2^k -1. K secret bit should be embedded in 2^k -1 cover bit with at most 1 change. It is denoted by (1,7,3), where 1 represent number of changes allowed,7 represent number of cover bit involved in the operation and 3 represent number of secret bit to be embedded. Now the cover bit selection and embedding mechanism to be designed in such a way that, k secret bits should be embedded in n cover bits with at most 1 change.

1) Cover bit Selection: Two types of cover bit selection are attempted in the above proposed technique and the results are shown for both the types.

Method1: In this method the LSB of every byte is chosen as cover bit. 7 bits of data are required to embed a 3 bit secret data. Those 7 bits are collected from seven consecutive bytes of the image. All 7 bytes' LSB is serving as cover bit.

Method2: In this method to collect 7 cover bit for the operation, on every pixel last two bits of red channel, last three bits of green channel and last two bits of blue channel are taken.

2) Secret bit Embedding: In order to embed and extract the 3 secret bit in the 7 cover bit with atmost 1 change, a reversible embedding and extraction algorithm should be designed. Equation 3 shown below will be used to meet the above goal. Assume b_1,b_2,b_3 are the secret bits, x_1,x_2,x_3,x_4,x_5,x_6,x_7 are cover bits. The cover bits are adjusted according to the secret bits b_1, b_2 and b_3 with atmost 1 change i.e. only one change is permitted out of all the 7 cover bits. At the same time the secret bit should be mapped inside the cover bit. The following equation is used in both embedding and extraction process.

$$\begin{aligned} b_1 &= x_1 \oplus x_4 \oplus x_6 \oplus x_7 \\ b_2 &= x_2 \oplus x_4 \oplus x_5 \oplus x_7 \\ b_3 &= x_3 \oplus x_5 \oplus x_6 \oplus x_7 \end{aligned} \quad . \quad \text{....................[3]}$$

The above 3 expression in equation 3 is operated to check the coincidence of secret bit against cover bit. An exclusive OR operation is performed on the cover bit; if all the three expression is satisfied no adjustment is required on the cover bit. Sometimes the cover bit by itself, is suitable to fit the secret data. If any or more than one of the expressions in equation 3 is not satisfied then modification on the cover bit is followed according to Table 1. This slight modification on the cover bit enable the secret bit to be mapped on the cover bit with at most only one change. Since the cover bit are adjusted according to the secret bit; during extraction the same equation can be used in recovering the secret bit from the cover bit.

Secret bit Positions not matched (b_1,b_2,b_3)	1	2	3	1,2	2,3	1,3	1,2,3
Cover bit to be inverted	x_1	x_2	x_3	x_4	x_5	x_6	x_7

TABLE 1: Embedding/ Extraction Reference

Huffman coding technique is used in the proposed method to securely and imperceptibly hide the secret image in the cover image. The Huffman encoded bit stream and Huffman table bit stream is embedded in the cover image pixel either by method1 or method2 through Matrix embedding technique. Cover bit selection will differ in method1 and method2 whereas embedding process remain same.

3.4. Hashing

Finally to attain the integrity of the stego image; the secret image is hashed prior to embedding. This hash code should be send as a supplementary component in addition to stego image. In the receiving end, the retrieved secret image is hashed to cross check against the hash code received. If both the hash codes are same, it conveys no intruder has modified the stego image.

3.5. Embedding Process

Fig. 2a shows the embedding process carried on the sender side. The Hash code of secret image and stego images are sent to receiver.

The steps carried on the sender side are given below:

Step 1: Hash the secret image.

Step 2: The Secret image is converted into a Numerical matrix which contains the RGB value or intensity of each pixel.

Step 3: Apply Huffman encoding for the output obtained from Step 2 which results in Huffman table and Huffman encoded secret image bit streams.

Step 4: Group the above obtained binary bit stream (Huffman table and Huffman encoded) in chunk of three bits.

Step 5: M1: Method1:- Each color image pixel is represented by 3 bytes (RGB). Collect 7 consecutive bytes from the image. All 7 bytes' LSB is serving as cover bit.

Step 6: M1: Method1:- Using equation 3 adjusts the 7 bytes LSB to match the three secret bit chunk obtained in Step 4. (OR)

Step 5: M2: Method2:- Each color image pixel is represented by 3 bytes (RGB). In this method to collect 7 cover bit for the operation, on every pixel LSB and LSB -1 from Red channel, LSB, LSB -1 and LSB -2 from Green channel, LSB and LSB -1 from Blue channel; a total of 7 bits are chosen as cover bit.

Step 6: M2: Method2:- Using equation 3 adjusts the above 7 bits to match the three bit chunk obtained in Step 4.

Step 7: Repeat Step5 and Step6 until all the 3 secret bit chunks are mapped over the cover image pixels moving from left to right and top to bottom of the cover image.

Step 8: Send the Hash Code and stego image obtained from Step 7 to the receiver.

3.6. Extraction Process

Fig. 2b shows the extraction process carried on the receiver side. Upon receiving the stego image, and the Hash code, receiver should extract the Huffman table, Huffman encoded bit streams, and secret image dimension from the stego image.

The steps carried on the receiver side are given below:

Step 1: Apply the relevant bit collection on the stego image pixel depends on the method (method1/method2); the secret bit is embedded in the cover image as explained in embedding process.

Step 2: Size of secret image, Huffman Table and Huffman symbols are retrieved.

Step 3: The Binary Huffman table is then converted to the actual format that can be accepted by the Huffman decoding.

Step 4: The Huffman table and Huffman encodings obtained in Step 2 are used in Huffman decoding process. As a result RGB/intensity value, for every pixel of secret image is obtained.

Step 5: Finally, the image is constructed using all the pixels which is computed in Step 4 will reveal the secret image.

Step 6: To ensure the stego image integrity, the received hash code is compared against the Hash code of constructed secret image. If both are equal, cover image is free from intruder attack.

The intermediate results obtained in every stage of embedding and extraction process are redirected to a text file may be assumed for better understanding of the proposed method wherever required.

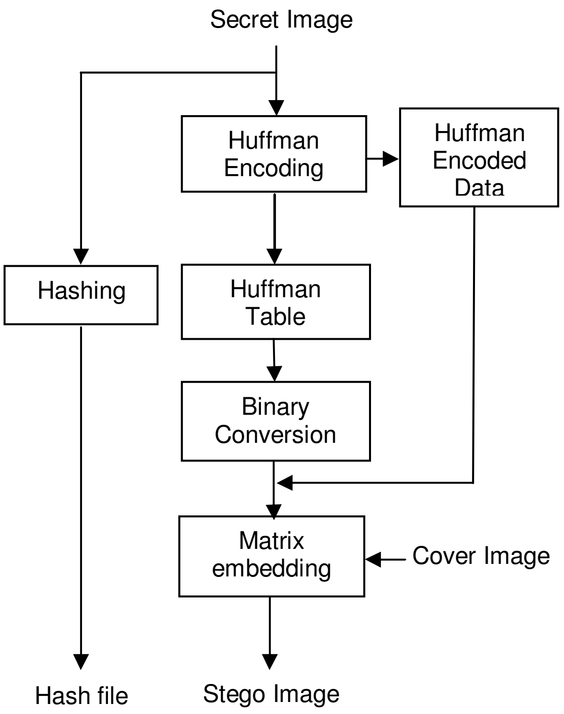


FIGURE 2a: Embedding Process

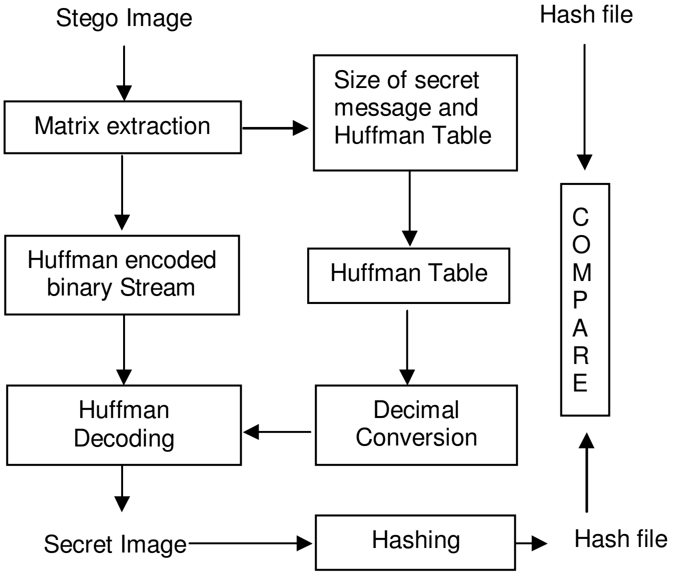


FIGURE 2b: Extraction Process

4. EXPERIMENTAL RESULTS

Java 2.0 and MATLAB 7.6 are the programming tools used to implement the proposed method. PSNR, Embedding Capacity and Mean Square Error are the three metrics taken here to consolidate the strength of proposed method. PSNR result is shown separately for all the channels. Two tables are used to present the performance of both the methods. The same cover image of size 256 X 256 is used in both the methods. The cover image and secret image taken here for experimentation is 24 bit color depth bmp (Bit Map Format) image.

A secret image Cameraman Fig. 3 of various sizes is embedded in the RGB cover images like Lena, Airplane, Baboon and Boat each of size 256 x 256. Fig. 4-7 shows the cover images, obtained stego images and histogram arrived in method1 and method2 of matrix embedding technique. Table2 and Table3 show the experimental results of method1 and method2 respectively. The PSNR and MSE arrived using the proposed method shows that the distortion occurred in stego image are very less. In method1 secret image of different sizes such as 85x85, 90x90 and 95x95 with 24 bit depth are embedded. The maximum capacity that the cover image can hold is 216,600 bits which is 26.5KB. The embedding capacity is 14% of the cover image using method1. The average PSNR and mean in method1 for 95x95 secret image is 58 and 0.12 respectively.

In method2, since the 7 cover bits are collected on a single pixel, the embedding capacity of the same cover image is better than method1. In method2, the same secret image Cameraman Fig. 3 of different size such as 85x85, 90x90, 95x95, 140x140, 150x150, and 155x155. In method2 a higher capacity is achieved but PSNR and mean is compromised. The maximum capacity that the cover image can hold is 576,600 bits which is 70.38KB. The embedding capacity is 37% of the cover image using method2. The average PSNR and mean in method2 for 155x155 secret image is 50 and 0.6 respectively. The PSNR and mean has declined with an enhanced capacity; but still PSNR value with more than 40 is acceptable.

Cover Image of size 256 X 256		Red Channel		Green Channel		Blue Channel	
		PSNR	MSE	PSNR	MSE	PSNR	MSE
Lena	85 x 85	57.94	0.1044	57.90	0.1052	57.89	0.1057
	90 x 90	57.63	0.1120	57.45	0.1169	57.51	0.1151
	95x 95	57.18	0.1243	57.22	0.1232	57.11	0.1263
Airplane	85 x 85	57.94	0.1044	57.89	0.1057	57.82	0.1072
	90 x 90	57.51	0.1151	57.61	0.1125	57.46	0.1164
	95x 95	57.23	0.1227	57.12	0.1259	57.19	0.1242
Baboon	85 x 85	57.87	0.1061	57.93	0.1046	57.87	0.1060
	90 x 90	57.54	0.1145	57.55	0.1141	57.49	0.1156
	95x 95	57.15	0.1252	57.22	0.1232	57.17	0.1246
Boat	85 x 85	57.95	0.1040	57.88	0.1059	57.82	0.1073
	90 x 90	57.56	0.1139	57.55	0.1141	57.46	0.1167
	95x 95	57.15	0.1251	57.14	0.1254	57.14	0.1255

TABLE 2: 7 COVER BIT ON 7 BYTE (METHOD 1)

Cover Image of size 256 X 256		Red Channel		Green Channel		Blue Channel	
		PSNR	MSE	PSNR	MSE	PSNR	MSE
Lena	85 x 85	54.55	0.2277	48.40	0.9393	54.55	0.2278
	90 x 90	54.15	0.2497	47.96	1.0397	54.30	0.2411
	95x 95	53.81	0.2700	47.72	1.0969	53.87	0.2665
	140x140	51.02	0.5131	44.77	2.1646	50.97	0.5200
	150x150	50.50	0.5783	44.17	2.4874	50.37	0.5959
	155x155	50.25	0.6131	43.96	2.6100	50.22	0.6169
Airplane	85 x 85	54.56	0.2275	48.40	0.9379	54.57	0.2266
	90 x 90	54.28	0.2423	47.94	1.0445	54.20	0.2469
	95x 95	53.81	0.2700	47.57	1.1366	53.89	0.2652
	140x140	50.98	0.5180	44.80	2.1505	51.02	0.5134
	150x150	50.45	0.5856	44.16	2.4921	50.51	0.5781
	155x155	50.23	0.6155	44.04	2.5631	50.15	0.6275
Baboon	85 x 85	54.57	0.2267	48.34	0.9527	54.64	0.2229
	90 x 90	54.33	0.2397	47.97	1.0374	54.12	0.2517
	95x 95	53.89	0.2652	47.70	1.1028	53.75	0.2740
	140x140	51.01	0.5145	44.73	2.1881	50.97	0.5190
	150x150	50.47	0.5824	44.24	2.4460	50.43	0.5882
	155x155	50.21	0.6191	43.98	2.6004	50.16	0.6266
Boat	85 x 85	54.61	0.2248	48.27	0.9673	54.56	0.2272
	90 x 90	54.24	0.2448	47.89	1.0569	54.27	0.2427
	95x 95	53.86	0.2673	47.72	1.0984	53.81	0.2699
	140x140	51.04	0.5110	44.76	2.1710	51.01	0.5147
	150x150	50.43	0.5882	44.25	2.4384	50.48	0.5815
	155x155	50.30	0.6064	43.93	2.6298	50.23	0.6160

TABLE 3: 7 COVER BIT ON 1 PIXEL – 2, 3,2 (METHOD 2)

The proposed method's hiding capacity depends upon the Huffman encoding output. The Huffman encoded result of a secret image (Huffman encoded bit stream and Huffman Table) size should be lesser than the total number of LSB spot available in the cover image. The last 64 pixel in cover image is reserved for storing the technical details which will be used in the receiver side to extract the secret image from the stego image. This 64 pixel (64x3=192 bytes) should be excluded while computing the hiding capacity of cover image. Image of any size/richness can be hidden through our proposed method, provided it meets the above said condition. Integrity of the stego image is verified by crosschecking the hash code received against the constructed secret image hash code. If both hash code are same, it conveys no intruder modified the stego image.

4.1. Discussion

In method2 the PSNR of green channel is less, compared to the other two channels. It is due to the reason that the cover bits are selected in the same pixel in this order (2, 3, and 2). Two bits from red channel, three bits from green channel and two bits from blue channel are taken. Out of 7 bits, 3 bits are taken from green channel; hence this channel is highly vulnerable to distortion. So, as a result the PSNR of green channel has declined in method2.

We quite often found that a secret image which is richer and whose dimension is lesser than Cameraman,(shown in Fig. 3) say 100 X 100 cannot be embedded in this 256 X 256 cover image shown in figure 4. In contrast, a secret image which is not richer and whose dimension is higher than 100 X 100 can be embedded in the cover image. This makes us to finalize that the embedding capacity of our proposed technique depends on Huffman encoding. Any image, whose Huffman compression is less, fits in the cover image irrespective of its size and richness.

The embedding capacity of the cover image can be improved further, if a pixel adjustment process technique is adapted. The number of bits get embedded in the proposed technique is just 3 bit per pixel in method1 or 3 bit using LSB's of seven consecutive bytes in method2. Pixel adjustment process technique is just substituting the intensity of the every cover pixel with an equivalent resembling pixels. This could exploit the cover pixels in embedding greater than 3 bits (9 bits/pixel). But, it will be on the cost of compromising a little bit distortion gets introduced on the cover image.

To discuss on security side, the proposed technique is robust enough; because extracting a data without knowing the architecture of the proposed technique is difficult, moreover data is Huffman encoded. Stego image integrity is validated through hashing which give confidence to the receiver. Thus, the privacy and security issues are addressed in this proposed technique to a reasonable extent.

CONCLUSION

We had proposed an image steganography algorithm which brings a better PSNR and MSE. The experimental results show that distortion between cover and stego image is minimum. Capacity improvement and distortion reduction has been addressed in this proposed technique. In the proposed method, the embedding capacity of the cover image is increased which results in slight decline in both PSNR and MSE parameters. The veracity of the stego image is verified and then progressed for their usage on receiver side. The proposed technique is not robust against any geometrical distortion such as rotation, translation, scaling, cropping etc., induced on the stego image. Improving this parameter is still under research and not matured yet.

FUTURE WORK

The proposed algorithm should be customized to support embedding in the frequency domain. It should be enhanced to withstand geometrical distortion induced on the image.

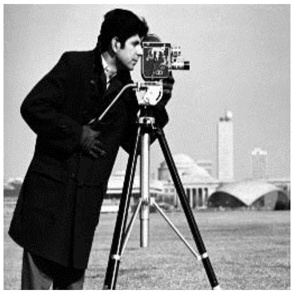
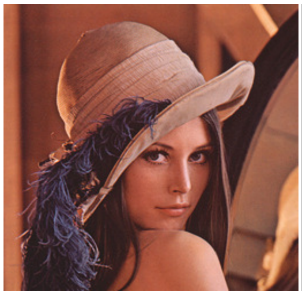
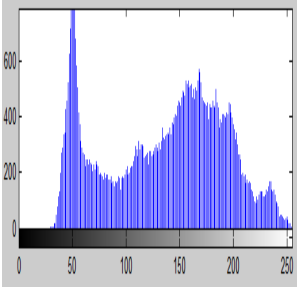
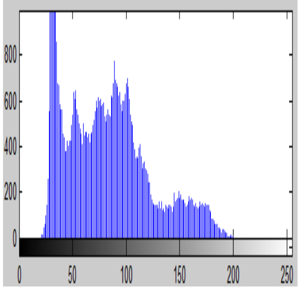


FIGURE 3: Cameraman







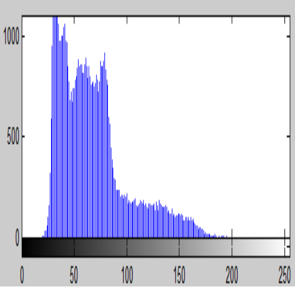
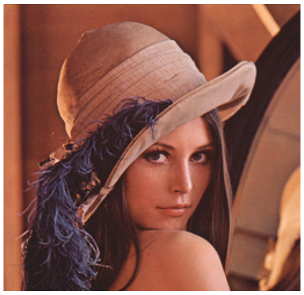
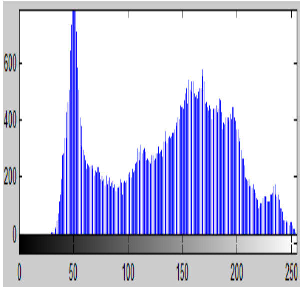
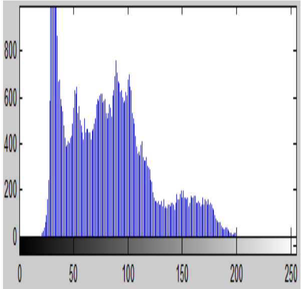


FIGURE 4a: Lena Cover **FIGURE 4b:** Red Channel **FIGURE 4c:** Green Channel **FIGURE 4d:** Blue Channel







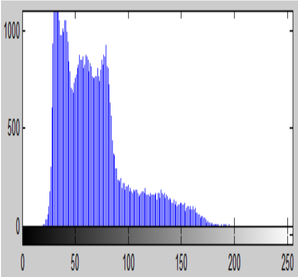
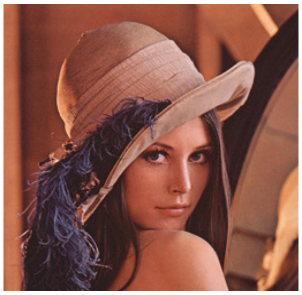
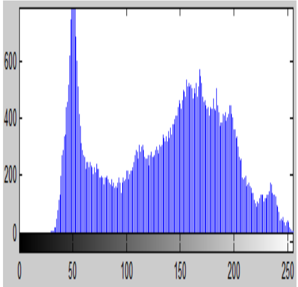
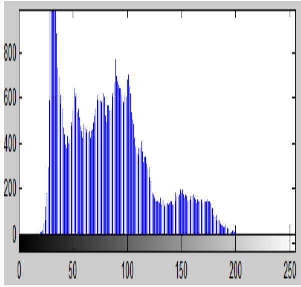


FIGURE 4e: Lena Stego M1 **FIGURE 4f:** Red Channel **FIGURE 4g:** Green Channel **FIGURE 4h:** Blue Channel







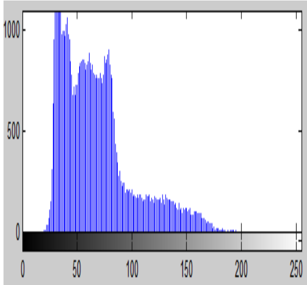


FIGURE 4i: Lena Stego M2 **FIGURE 4j:** Red Channel **FIGURE 4k:** Green Channel **FIGURE 4l:** Blue Channel

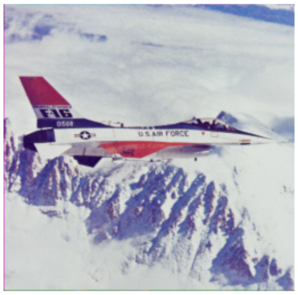


FIGURE 5a: Airplane Cover

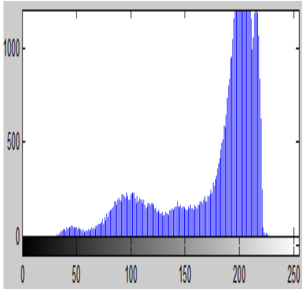


FIGURE 5b: Red Channel

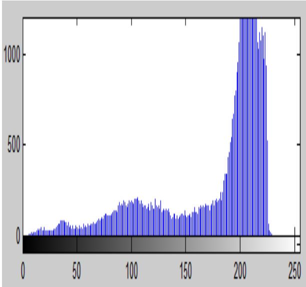


FIGURE 5c: Green Channel

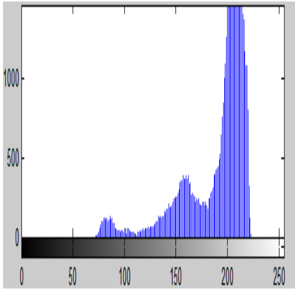


FIGURE 5d: Blue Channel

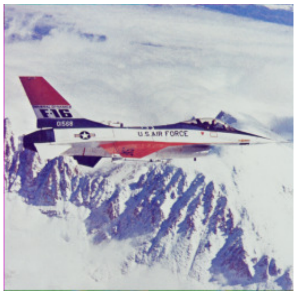


FIGURE 5e: Airplane Stego M1

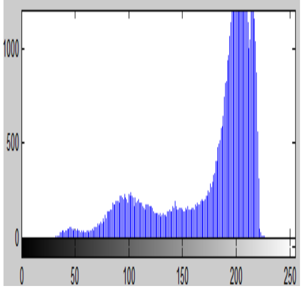


FIGURE 5f: Red Channel

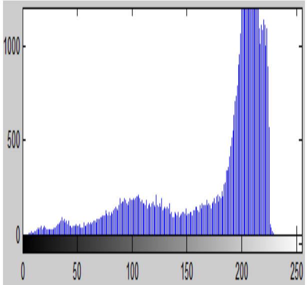


FIGURE 5g: Green Channel

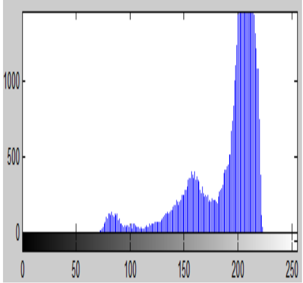


FIGURE 5h: Blue Channel

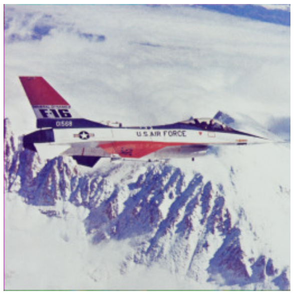


FIGURE 5i: Airplane Stego M2

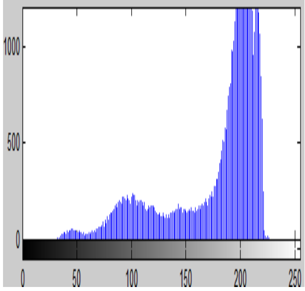


FIGURE 5j: Red Channel

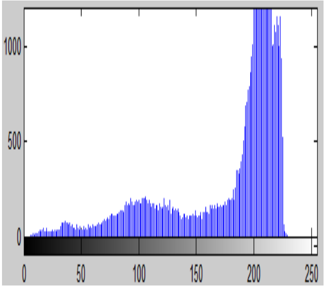


FIGURE 5k: Green Channel

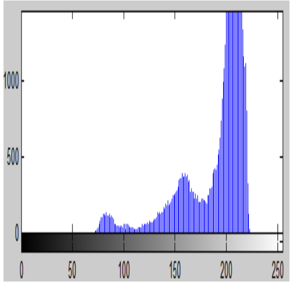


FIGURE 5l: Blue Channel

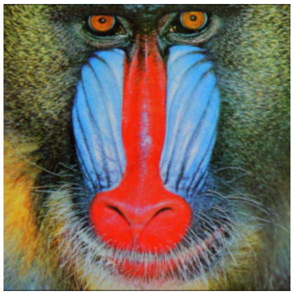


FIGURE 6a: Baboon Cover

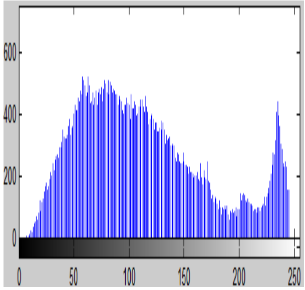


FIGURE 6b: Red Channel

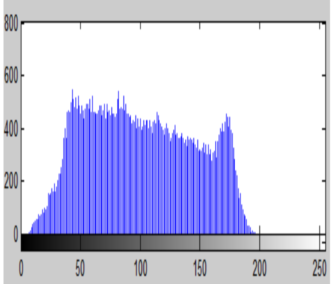


FIGURE 6c: Green Channel

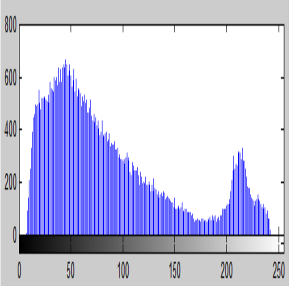


FIGURE 6d: Blue Channel

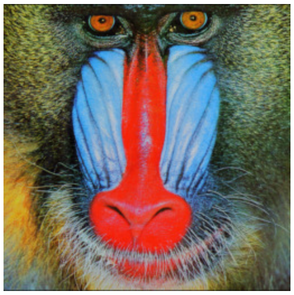


FIGURE 6e: Baboon Stego M1

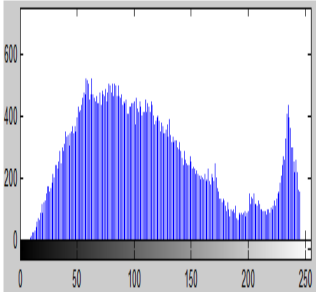


FIGURE 6f: Red Channel

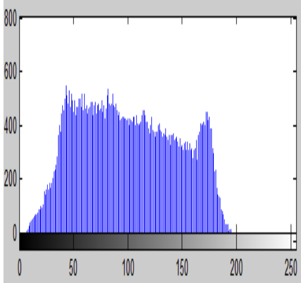


FIGURE 6g: Green Channel

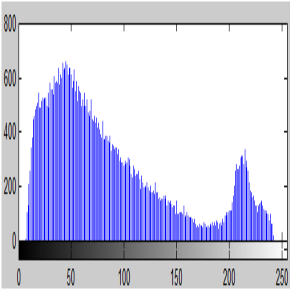


FIGURE 6h: Blue Channel

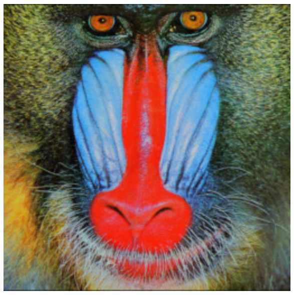


FIGURE 6i: Baboon Stego M2

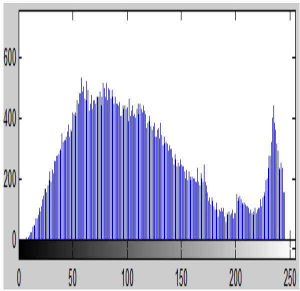


FIGURE 6j: Red Channel

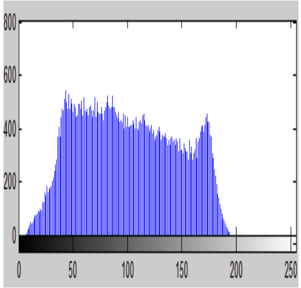


FIGURE 6k: Green Channel

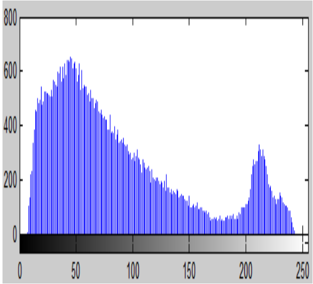


FIGURE 6l: Blue Channel

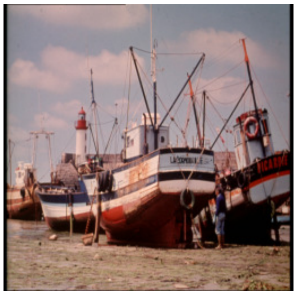


FIGURE 7a: Boat Cover

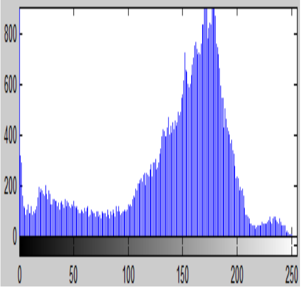


FIGURE 7b: Red Channel

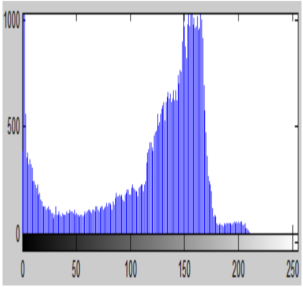


FIGURE 7c: Green Channel

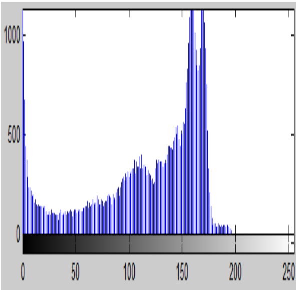


FIGURE 7d: Blue Channel

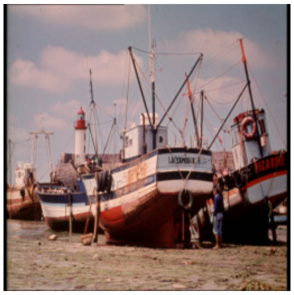


FIGURE 7e: Boat Stego M1

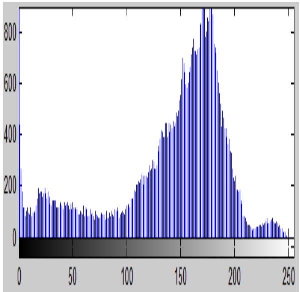


FIGURE 7f: Red Channel

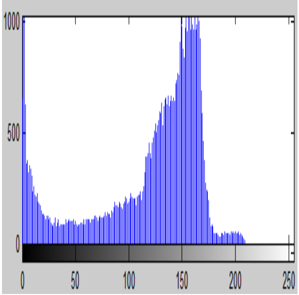


FIGURE 7g: Green Channel

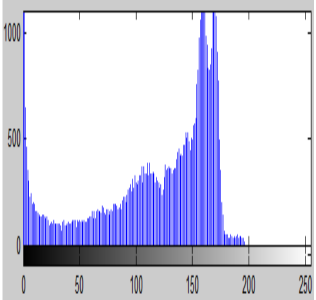


FIGURE 7h: Blue Channel

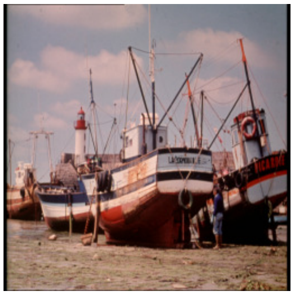


FIGURE 7i: Boat Stego M2

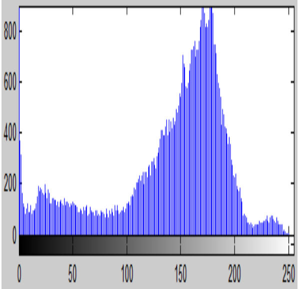


FIGURE 7j: Red Channel

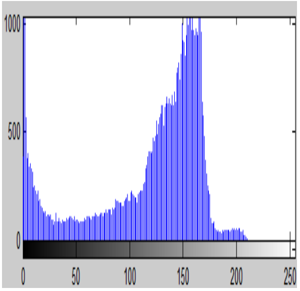


FIGURE 7k: Green Channel

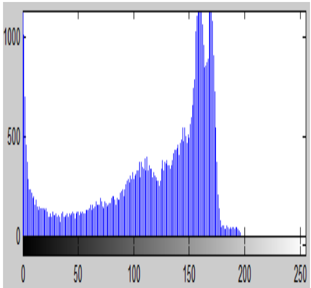


FIGURE 7l: Blue Channel

REFERENCES

[1] Injemar J. Cox, Matthew L. Miller, Jeffrey A. Bloom, Jessica Fridrich, Ton Kalker. *Digital Watermarking and Steganography.* Morgan Kaufmann, Second Edition,2008.

[2] Professor Luca Trevisan, 2001, "*Lecture notes on Intro. To CS Theory.*" Online. Available: http:// ww.cs.berkeley.edu/~luca/cs170/notes/lecture15.ps

[3] Rafael C. Gonzalez Richard E. Woods. *Digital Image Processing.* ,PHI, Third Edition, 2008.

[4] Alasdair McAndrew. Introduction *to Digital Image Processing with MATLAB,* CENGAGE Learning, 2004

[5] [Online].Available : http://www.binaryessence.com

[6] Ali K. Hmood, Z. M. Kasirun, Hamid A. Jalab,Gaz Mahabubul Alam, A. A. Zaidan, B. B. Zaidan. “On the accuracy of hiding information metrics: Counterfeit protection for education and important certificates.” *International Journal of the Physical Sciences*, Volume. 5, Issue 7, pp. 1054-1062, August,2010.

[7] Chang, C.C and Tseng, H.W. 2004. “A Steganographic method for digital images using side match*.” Pattern Recognition Letters*, 25: 1431 – 1437, June 2004.

[8] Po-Yueh Chen,Wei-En Wu. “A Modified Side Match Scheme for Image Steganography” , *International Journal of Applied Science and Engineering,* Volume 7, Issue 1, pp. 53-60, October 2009.

[9] A. Nag, S. Biwa’s, D. Sarkar, P.P. Sarkar. “A Novel Technique for Image Steganography Based on DWT and Huffman Encoding” , *International Journal of Computer Science and Security,* Volume 4, Issue 6, pp. 561-570, Feb. 2011.

[10] R.Amirtharajan, Sandeep Kumar Beher, Motamarri Abhilash Swarup, Mohamed Ashfaaq K and John Bosco Balaguru Rayappan. “Colour Guided Colour Image Steganography”, *Universal Journal of Computer Science and Engineering Technology*, Volume 1, pp.16 – 23, Oct . 2010.

[11] Santosh Arjun, Atul Negi, Chaitanya Kranti, and Divya Keerthi. “An Approach to Adaptive Steganography Based on Matrix Embedding” *TENCON 2007 - 2007 IEEE Region 10* , Volume 1,pp.1-4, Oct . 2007.

A Novel Steganography Technique That Embeds Security Along With Compression

Anuradha anu107sharma@gmail.com
Student/CSE DCRUST,Murthal
Sonepat, 131039, India

Nidhi nidhi.sharma.1012@gmail.com
Student/IT Banasthali Vidyapeeth
Bansathali, 304022, India

Rimple rimple.gilhotra@hotmail.com
Student/CSE Banasthali Vidyapeeth
Bansathali, 304022, India

Abstract

Problem faced by today's communicators is not only security but also the speed of communication. This paper provides a mechanism that increases the speed of communication by reducing the size of content; for this data compression method is used and security factor is added by using Steganography. Firstly, the focus has been made on Data Compression and Steganography. Finally, proposed technique has been discussed. In Proposed technique first data is compressed to reduce the size of the data and increase the data transfer rate. Thereafter on compressed data state table operation is applied to improve the security. Then, this is used as the input to the LSB technique of Steganography. At receiver end, the LSB extraction technique is used, thereafter the state table operation in reverse form is applied and finally the original data is obtained. Hence our proposed technique is effective that can reduce data size, increases data transfer rate and provides the security during communication.

Keywords: Compression, Arithmetic Coding, Steganography, Hexadecimal, One Time Pad, Least Significant bit (LSB).

1. INTRODUCTION

The present network scenario demands exchange of information with more security and reduction in both the space requirement for data storage and the time for data transmission [9]. This can be accomplished by compression and data hiding. Now a day's people use the network as the basic transport for transmitting their personal, secure and day to day information. Hence they need some form of security from the third person during transmission which is provided by Steganography where Steganography refers to the science of invisible communication [10, 11]. Along with that sometimes the data that is to be sent is in huge amount that requires lots of space and bandwidth, so we must have a mechanism with the help of which we can reduce the size of data as well as time and bandwidth is saved; this is done by using Arithmetic Coding.

Proposed mechanism embeds the compressed data behind the cover data; this mechanism is used to achieve the present network scenario for exchange of information with more security and compression.

2. DATA COMPRESSION

Compression is used just about everywhere. Compression algorithms reduce the redundancy in data representation to decrease the storage required for that data. The task of compression consists of two components, an encoding algorithm that takes a message and generates a "compressed" representation and a decoding algorithm that reconstructs the original message or some approximation of it from the compressed representation. We distinguish between lossless algorithms, which can reconstruct the original message exactly from the compressed message, and lossy algorithms, which can only reconstruct an approximation of the original message. Lossless algorithms are typically used for text, and lossy for images and sound where a little bit of loss in resolution is often undetectable, or at least acceptable [4].

2.1 Arithmetic Coding

In information theory an entropy encoding is a lossless data compression scheme that is independent of the specific characteristics of the medium. One of the main types of entropy coding assigns codes to symbols so as to match code lengths with the probabilities of the symbols. Typically, these entropy encoders are used to compress data by replacing symbols represented by equal-length codes with symbols represented by codes where the length of each codeword is proportional to the negative logarithm of the probability. Therefore, the most common symbols use the shortest codes. The most popular entropy encoding is Arithmetic Encoding [5].

In arithmetic coding, a message is represented by an interval of real numbers between 0 and 1. As the message becomes longer, the interval needed to represent it becomes smaller, and the number of bits needed to specify that interval grows. Successive symbols of the message reduce the size of the interval in accordance with the symbol probabilities generated by the model. The more likely symbols reduce the range by less than the unlikely symbols and hence add fewer bits to the message.[6] Before anything is transmitted, the range for the message is the entire interval [0, l), denoting the half-open interval 0 <= x < 1 [12]. Thus, the algorithm successively deals with smaller intervals, and the code string, viewed as a magnitude, lies in each of the nested intervals. The data string is recovered by using magnitude comparisons on the code string to recreate how the encoder must have successively partitioned and retained each nested subinterval. [13]

3. STEGANOGRAPHY

Steganography is the art and science of communicating in such a way that the presence of a message cannot be detected [1]. Steganography involved a Greek fellow named Histiaeus. As a prisoner of a rival king, he needed a way to get a secret message to his own army. He shaves the head of a willing slave and tattoos his message on to the bald head. When hairs grew back, off he went to deliver the hidden writing in person [3].

Steganography derives from the Greek word “steganos” means covered or secret and “graphy” means writing. On the simplest level Steganography is hidden writing, whether it consists of invisible ink on paper or copyright information hidden within an audio file. Today, Steganography, “stego” for short, is most often associated with in other data in an electronic file. This is done by replacement the least important or most redundant bits of data in the original file i.e. bits that the human eye or ear hardly misses with hidden data bits [2].

3.1 Where it Comes From [3,8]

One of the earliest uses of Steganography was documented in histories. Herodotus tells how around 400 B.C. Hisitieaus shaved the head of his most trusted slave and tattooed it with the message which disappeared after the hair has regrown. The purpose of this message was to investigate the revolt against the Persians. Another slave could used to send the reply.

During the American Revolution, invisible ink which would glow over a flame was used by both the British and the American’s to communicate secretly. German hides text by using invisible ink to print small dots above or below letters and by changing the heights of letter-strokes in cover texts.

In world war 1prisoners of war would hide Morse code messages in letters home by using the dots and dashing on I, j, t and f. censors intercepting the messages were often altered by the phrasing and could change them in order to alter the message.

During world war 2nd the Germans would hide data in microdots. This involved photographing the message to be hidden and reducing the size so that it can be used as a period within another document.

3.2 Types of Steganography[3]

Steganography can be split into two types, these are Fragile and Robust. The following section describes the definition of these two different types of Steganography.

3.2.1 Fragile

Fragile Steganography involves embedding information into a file which is destroyed if the file is modified. This method is unsuitable for recording the copyright holder of the file since it can be so easily removed, but is useful in situations where it is important to prove that the file has not been tampered with, such as using a file as evidence in a court of law, since any tampering would have removed the watermark. Fragile Steganography techniques tend to be easier to implement than robust methods.

3.2.2 Robust

Robust marking aims to embed information into a file which cannot easily be destroyed. Although no mark is truly indestructible, a system can be considered robust if the amount of changes required to remove the mark would render the file useless. Therefore the mark should be hidden in a part of the file where its removal would be easily perceived.

3.3 Key Features

There are several key features regarding Stegangraphy and its usage are as follows:

- The main goal of Steganography is to hide a message m in some audio or video (cover) data d, to obtain new data d', practically indistinguishable from d, in such a way that an eavesdropper cannot detect the presence of m in d'.
- The goal of Steganography is to hide the message in one-to-one communication
- We can hide as much data as possible.
- Ease of detection level should be Difficult.
- We can hide as much data as possible.
- Goal of detector is to detect the hidden data.

3.4 Applications

There are various areas where Steganography can be applied:

- Confidential communication and secret data storing.
- Protection of data alteration
- Access control system for digital content distribution.
- Media Database systems
- Corporate espionage, Cover Communication by Executives, Drug dealers, Terrorists.

4. PROPOSED TECHNIQUE

The proposed technique is based on the concept of arithmetic coding and Steganography in which a word of text is converted into floating point number that lie in range between 0 and 1. This floating point number is converted into hexadecimal number and after that one time pad and a state table is used to hide the compressed hexadecimal data.

At Receiver end, data is extracted by using the Steganography method that will be explained later; after that decompression is done to obtain the original word.

4.1 Compression and Hiding

Firstly input symbol is compressed using arithmetic coding after that one time pad and the state table is used on the result of arithmetic coding.

ALGORITHM

To compress and encrypt the message Algorithm includes following steps:

Step 1: Using table encode the input symbol.

```
    a) Initialize lower_ bound=0, upper_ bound=1

    b) While there are still symbols to encode

        Current _range = upper _bound - lower _bound
        Upper_ bound = lower _bound + (current _range * upper _bound of new symbol)
        Lower_ bound = lower_ bound + (current _range * upper_ bound of new symbol)

      End while
```

Step 2: The string may be encoded by any value within the probability range and after that convert the output decimal number into hexadecimal data.

Step 3: Choose 2nd MSB of the selected cover image. This is the one time pad.

Step 4: Now, the state table operation is applied on the hexadecimal equivalent and the one time pad. The information about this state table is exchanged between sender and receiver earlier. This state table will help in confusing the intruder because the intruder does not know anything about the state table. Hence, the security level is increased further. The state table is given in Table 1. [7]

Input		**Output**	
0	0	0	0
1	0	0	1
0	1	1	0
1	1	1	1

TABLE 1: State Table

Step 5: The output obtained from step 4 is used in LSB substitution method of Steganography.

Step 6: The final embedded cover image is send to the receiver side.

4.2 Decompression and Extraction

Algorithm

Step 1: Extract the LSB’s from the cover image.
Step 2: Choose 2nd MSB’s of the cover Image, this is the onetime pad.

Step 3: Apply the state table (Table 1) to the LSB’s and the 2nd MSB’s of the cover image.

Step 4: The output obtained from step 3 is the original hidden data in hexadecimal format.

Step 5: Convert the hexadecimal format into decimal equivalent.

Step 6: Apply arithmetic decoding procedure.

```
Encoded_ value=Encoded input

While string is not fully decoded

Identify the symbol containing encoded value within its range
current_ range = upper _bound of new symbol - lower _bound of new symbol
encoded value = (encoded _value - lower_ bound of new symbol) ÷ current_ range

End while
```

Output: The output is the original symbol.

4.3 Example

Suppose Input Data is: “ganga”

Step 1: Create Probability Table

For character g:

Occurrence of character ‘g’ in Input data is “2”.
Probability is 2/5=0.4

For character a:
Occurrence of character ‘a’ in Input data is “2”.
Probability is 2/5=0.4

For character n:
Occurrence of character ‘n’ in Input data is “1”.
Probability is 1/5=0.2

The probability table is prepared according to the occurrences of the letters. This is explained in table 2.

Symbol	**Probability**	**Range(lower_ bound, upper_ bound)**
A	40%	[0.00,0.40)
G	40%	[0.40,0.8)
N	20%	[0.8,0.1)

TABLE2: Symbols along with probability of occurrence

4.3.1 Compression and Hiding

Data to be encoded is “ganga”

Step1:
Encode 'g'
current_ range = 1 - 0 = 1
upper bound = 0 + (1 × 0.4) = 0.4
lower bound = 0 + (1 × 0.8) = 0.8

Encode 'a'
current range = 0.8 - 0.4 = 0.4
upper bound = 0.4 + (0.4 × 0.0) = 0.4
lower bound = 0.4 + (0.4 × 0.8) = 0.56

Encode 'n'
current range = 0.56-0.4 = 0.16
upper bound = 0.4 + (0.16 × 0.8) = 0.528
lower bound = 0.4 + (0.16 × 1) = 0.56

Encode 'g'
current_ range = 0.56-0.528 = 0.032
upper bound = 0.528 + (0.032 × 0.4) = 0.5408
lower bound = 0.528 + (0.032 × 0.8) = 0.5536

Encode 'a'
current range = 0.5536 - 0.5408 = 0.0128
upper bound = 0.5408 + (0.0128 × 0.0) = 0.5408
lower bound = 0.5408 + (0.0128 × 0.4) = 0.54592

Step2:

The string "ganga" may be encoded by any value within the range [0.5408, 0.54592).
Now output is 0.54260 and its hexadecimal equivalent= 01010100001001100000

Step3: Select an Image which is considered as a cover image.

11001010	10101010	11100010	10100001	11100011
11100010	10100001	10101101	10001001	10101010
10101101	10101010	10100001	11100011	10100001
11100011	11001010	10101010	11001010	11100010
10101010	10101101	10100001	10101010	11100010
10101010	11100011	11001010	11100010	10101010
10101010	11001010	10101010	10101010	11100011
10101010	11100010	11100011	10101010	11001010

TABLE 3: Cover image

Step4: Choose 2nd MSB's of cover Image as a one time pad key.

Step5: Our One time pad is – 10101100000001011011
Data- 01010100001001100000 from step 2.
Apply operation on bits according to the given state table [1].

Step6: Final Output is: 0110011001110000000010000011100101000101

Step7: Now Apply LSB substitution method of stenography to hide data in cover image.

11001010	10101011	11100011	10100000	11100010
11100011	10100001	10101100	10001000	10101011
10101101	10101011	10100000	11100010	10100000
11100010	11001010	10101010	11001010	11100010
10101011	10101100	10100000	10101010	11100010
10101010	11100011	11001011	11100011	10101010
10101010	11001011	10101010	10101011	11100010
10101010	11100010	11100011	10101010	11001011

TABLE 4: Cover image with data hidden inside.

4.3.2 Decompression and Extraction

Step 1: Extract the LSB's of cover image which gives us hidden data.
Hidden Data: 0110011001110000000010000011100101000101

Step2: Reverse the operation on bits by taking combination of 2 bits, which gives the combination of one time pad key and actually compressed data i.e.
1001100110110000000010000110110100010100

Step3: Separate the one time pad key and compress data i.e.

One time pad key: 10101100000001011011
Data- 01010100001001100000

Step4: Convert hexadecimal format into decimal format i.e. 0.54260

Step5: Using the probability ranges from table decodes the three character string encoded as 0.54260.

Decode first symbol
0.54260 is within [0.4, 0.8)
0.54260 encodes 'g'

Remove effects of 'g' from encode value
Current _range = 0.8 - 0.4 = 0.4
Encoded _value = (0.54260-0.4) ÷ 0.4 = 0.3565

Decode second symbol
0.3565 is within [0.0, 0.4)
0.3565 encodes 'a'

Remove effects of 'a' from encode value
current range = 0.0 - 0.4 = 0.4
encoded value = (0.3565 - 0.0) ÷ 0.4 = 0.89125

Decode third symbol
0.89125 is within [0.8, 1)
0.89125 encodes 'n'

Remove effects of 'n' from encode value
Current _range = 1 - 0.8 = 0.2
Encoded _value = (0.89125-0.8) ÷ 0.2 = 0.45625

Decode second symbol
0.45625 is within [0.4, 0.8)
0.45625 encodes 'g'

Remove effects of 'g' from encode value
Current range = 0.0 - 0.4 = 0.4
Encoded value = (0.45625 - 0.4) ÷ 0.4 = 0.14063

Decode third symbol
0.14063 is within [0.8, 1)
0.14063 encodes 'a'
Now we are with our secret data i.e. "ganga"

5. BENEFITS

- In proposed system generated cipher text takes very less bandwidth of secure channel.
- Highly Secure.

6. CONCLUSION AND FUTURE SCOPE

The Present network scenario demands exchange of information with reduction in both space requirement for data storage and time for data transmission along with security. Our proposed technique fulfils all such requirements as this technique uses the concept of data compression and Steganography. Along with that the state table that increases the security further because the intruder does not have any idea about this state table. By using this technique we can reduce the size of data and after that compressed data can be hidden to provide the security. Hence this technique increased the data transfer rate and security during data communication. There exists some enhancement in the compression method used as future work. We can use any other compression method that will provide better compression ratio than the existing one.

7. REFERENCES

[1]. Christian Cachin, “An Information-Theoretic Model for Steganography”, A preliminary version of this work was presented at the 2nd Workshop on Information Hiding, Portland, USA, 1998, and appears in the proceedings (D. Aucsmith, ed., Lecture Notes in Computer Science, vol. 1525, Springer).Original work done at MIT Laboratory for Computer Science, supported by the Swiss National Science Foundation (SNF).March 3, 2004, pp. 1-14.

[2]. Eric Cole, Ronald L. Krutz, James W. Conley, “Network security bible” Wiley Pub. 2005, pp. 482-520

[3]. SecondLieutentJ.caldwell,“Steganography”,UnitedStatesAirForce,http://www.stsc.hill.af.mil /crosstalk/2003/caldwell.pdf, June2003.

[4]. Guy E. Blelloch. Computer Science Department. Carnegie Mellon University blellochcs. cmu.edu.http://www.cs.cmu.edu/afs/cs/project/pscicoguyb/realworld/www/compression. pdf ,September 25, 2010.

[5]. V.Kavitha , K.S Easwarakumar. “Enhancing Privacy in Arithmetic Coding” ICGST-AIML Journal, Volume 8, Issue I, pp. 23-28, June 2008.

[6]. IAN H. WIllEN, RADFORD M. NEAL, and JOHN G. CLEARY. “Arithmetic coding for data compression.” Communications of the ACM , Volume 30 Number 6,pp.521-540, June 1987.

[7]. Ajit Singh, Nidhi Sharma. “Development of mechanism for enhancing the Data Security using Quantum Cryptography.” Advanced Computing: An International Journal (ACIJ), Vol.2, No.3, pp.22-25, May 2011.

[8]. Herodotus. The Histories. London, England: J. M. Dent & Sons Ltd, 1992.

[9]. Ajit Singh , Rimple Gilhotra. “Data security using private key encryption system based on arithmetic coding.” International Journal of Network Security & Its Applications (IJNSA), vol.3, no.3, May 2011.

[10]. Mehdi Kharrazi, Husrev T. Sencar Nasir Memon. “Performance study of common image Steganography and steganalysis techniques.” *Journal of Electronic Imaging 15(4),041104 (Oct–Dec 2006)*

[11]. M. Kharrazi, H. T. Sencar, and N. Memon, Image Steganography Concepts and Practice, Lecture Notes Series, Institute for Mathematical Sciences, National University of Singapore, Singapore _2004

[12]. J.A Storer, (1988) “Data Compression: Methods and Theory” Computer Science Press.

[13]. Glen G. Langdon, (1984) “An introduction to arithmetic coding”, IBM Journal of Research and Development Volume 28, No.2

Phishing Website Detection and Optimization Using Particle Swarm Optimization Technique

Radha Damodaram,M.C.A,M.Phil *radhabalaji10@gmail.com*
Asst. Professor,
Department of BCA, SS & IT,
CMS College of Science & Commerce,Coimbatore.

Dr.M.L.Valarmathi *ml_valarmathi@rediffmail.com*
Asst. Professor,
Dept. of Computer Science & Engg
Government College of Technology,Coimbatore.

Abstract

Fake websites is the process of attracting people to visit fraudulent websites and making them to enter confidential data like credit-card numbers, usernames and passwords. We present a novel approach to overcome the difficulty and complexity in detecting and predicting fake website. There is an efficient model which is based on using Association and classification Data Mining algorithms optimizing with PSO algorithm. These algorithms were used to characterize and identify all the factors and rules in order to classify the phishing website and the relationship that correlate them with each other. It also used MCAR classification algorithm to extract the phishing training data sets criteria to classify their legitimacy. After classification, those results have been optimized with Ant Colony Optimization (ACO) algorithm. But, this work has limitations like Sequences of random decisions (not independent) and Time to convergence uncertain in the phishing classification. So to overcome this limitation we enhance Particle Swarm Optimization (PSO) which finds a solution to an optimization problem in a search space, or model and predict social behaviour in the presence of phishing websites. This will improve the correctly classified phishing websites. The experimental results demonstrated the feasibility of using PSO technique in real applications and its better performance. This project employs the JAVA technology.

Keywords: Fake Website, Association and Classification Technique, ACO, PSO

1. INTRODUCTION

1.1 Phishing

In the computer field stealing information is a simple trick in which the hacker creates the duplicate page of a site and asks you to enter your details or credentials there. When you enter credentials such as username, password or credit card number the whole data goes to the hacker which he/she later use[4]. Phishing is an illegal process and there are many hackers who are behind the bars because of this fraudulent process called “Phishing”. Phishing is nowadays a major problem for the users of social networking sites and net banking.

There are many ways through which you can identify a phishing website. Some ways to detect phishing sites are:

- **Uniform Resource Locator (URL)**: Check always the web address or URL of the site you are visiting. Fake websites have different addresses than those of the genuine websites.

- **Install Anti-Phishing Software:** Use anti-phishing software to detect phishing sites.

- Browsers with inbuilt Software: Always work with the browsers who have anti-phishing software inbuilt in them. you may use opera, mozilla firefox, safari, google chrome, internet explorer 8 etc[5].

- **Slashes missing**: A fake website sometimes doesn't have slashes in-between its URL address. For example "http://www.scramable.com:login&mode=secured" but the original one is" http://www.scramable.com/wp_admin"[13].

- **Using fake passwords**: Always type your fake password on the websites that you are visiting first time and later you can change it.

- **Searching More:** Always use to search on good search engines like Google.

- **More information:** Always get more and more information about the site as phishing websites don't hold for too long.

- **Totally avoid Pop-Ups**: Never login with your original password in the pop-up windows as a hacker may ask you to enter the password in the pop up and after he redirects you to the exact page[17].

1.2 Phishing Nature and Detection

Phishing websites work by impersonating legitimate websites, and they have a very short life time. On average a phishing website lasts 62 hours, and users rarely visit the phishing website prior to the attack. Secondly, phishing attacks always generate mismatches between a user's perception and the truth. In successful web based phishing attacks, victims have believed they are interacting with websites which belong to legitimate and reputable organizations or individuals. Thus the crucial mismatch that phishers create is one of real *versus* apparent identity. Phishing attacks can be detected if we can detect such a mismatch. One approach is to predict a user's perception and then compare it with the actual fact understood by the system. CANTINA is an example of this approach [18].

The main issue with this approach is that the data the system relies on is under the control of attackers, and there are so many techniques that attackers can apply to manipulate the data to easily evade the detection. For CANTINA, attackers could use images instead of text in the body of the webpage, they could use iframes to hide a large amount of content from the users while computer programs can still see it; they could use Java script to change the content of the page after the detection has been done. The authentication credentials, which phishers try to elicit, ought to be shared only between users and legitimate organizations. Such (authentication credential, legitimate website) pairs are viewed as the user's binding relationships. In legitimate web authentication interactions, the authentication credentials are sent to the website they have been bound to [21]. In a phishing attack the mismatches cause the user to unintentionally break binding relationships by sending credentials to a phishing website. No matter what spoofing techniques or deception methods used, nor how phishing WebPages are implemented, the mismatch and violation of the binding relationships always exists. So, one can discover the mismatch by detecting violation of users' binding relationships. Hence phishing websites can be detected when both of the following two conditions are met:

1) The current website has rarely or never been visited before by the user;

2) The data, which the user is about to submit, is bound to website other than the current one. This detection process flow shown in Fig.1.1.

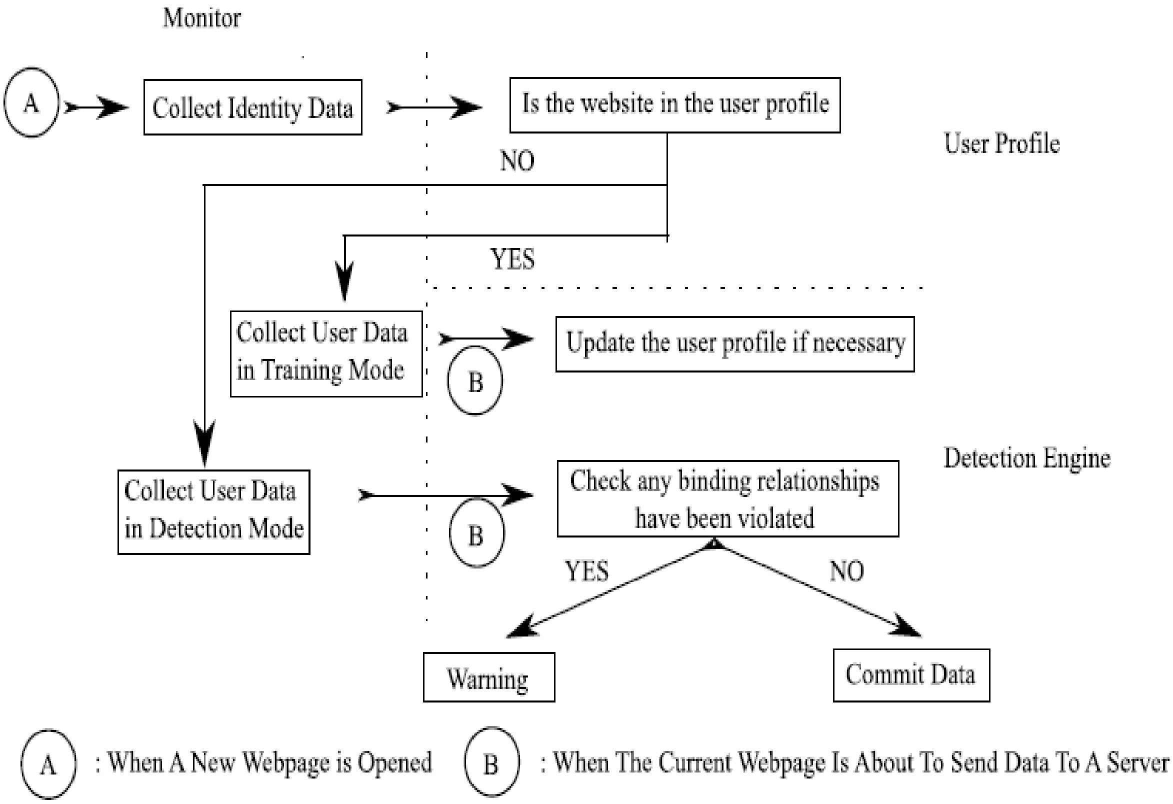

Monitor
A
Collect Identity Data
Is the website in the user profile
User Profile
NO
YES
Collect User Data in Training Mode
B
Update the user profile if necessary
Detection Engine
Collect User Data in Detection Mode
B
Check any binding relationships have been violated
YES
NO
Warning
Commit Data
A : When A New Webpage is Opened
B : When The Current Webpage Is About To Send Data To A Server

FIGURE 1.1 Detection Process Work Flow[20]

13. Ant Colony Optimization

The Ant Colony System or the basic idea of an ant food searching system is illustrated in Fig.1.2. In the left picture, the ants move in a straight line to the food. The next picture shows the situation soon after an obstacle is inserted between the nest and the food. To avoid the obstacle, first each ant chooses to turn right or left at random. Let us assume that ants move at the same speed depositing pheromone in the trail equivalently. However, the ants that, by chance, choose to turn right will reach the food sooner, whereas the ants that go around the obstacle turning right will follow a longer path, and so will take long time to circumvent the obstacle. As a result, pheromone gathered faster in the shorter path around the obstacle. Since ants prefer to follow trails with larger amounts of pheromone, ultimately all the ants congregate to the shorter path around the obstacle [1].

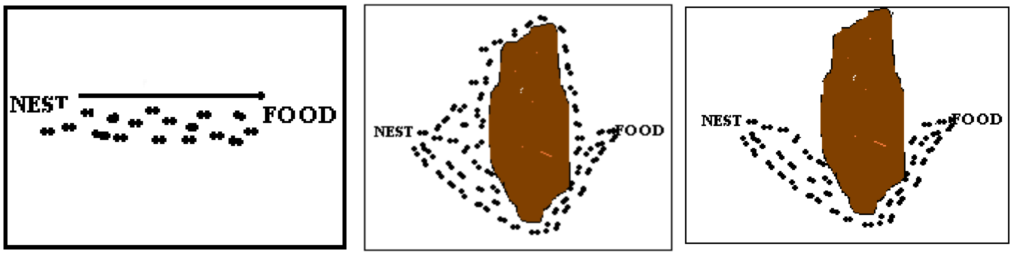


FIGURE 1.2- Illustrating the behaviour of real ant movements.

This new heuristic, called Ant Colony Optimization (ACO) has been found to be both robust and versatile in handling a wide range of combinatorial optimization problems. The main idea of ACO is to model a problem as the search for a minimum cost path in a graph. Artificial ants as if walk on this graph, looking for cheaper paths. Each ant has a rather simple behaviour capable of finding relatively costlier paths. Cheaper paths are found as the emergent result of the global cooperation among ants in the colony. The behaviour of artificial ants is inspired from real ants: they lay pheromone trails (obviously in a mathematical form) on the graph edges and choose their path with respect to probabilities that depend on pheromone trails. These pheromone trails progressively decrease by evaporation. In addition, artificial ants have some extra features not seen in their counterpart in real ants. In particular, they live in a discrete world (a graph) and their moves consist of transitions from nodes to nodes.

The ACO differs from the classical ant system in the sense that here the pheromone trails are updated in two ways. Firstly, when ants construct a tour they locally change the amount of pheromone on the visited edges by a local updating role. Secondly, after all the ants have built their individual tours, a global updating rule is applied to modify the pheromone level on the edges that belong to the best ant tour found so far [2].

1.5 Particle Swarm Optimization

Particle Swarm Optimization (PSO) technique is a modelled algorithm on Swarm intelligence, that gets a solution to an optimization problem in a search place, or model and predict social behaviour in the presence of objectives. The PSO is a stochastic, population-based computer algorithm modelled on swarm intelligence. Swarm intelligence is based on social-psychological principles and provides insights into social behaviour, as well as contributing to engineering applications.

The particle swarm optimization algorithm was first described in 1995 by James Kennedy and Russell C. Eberhart. The particle swarm simulates this kind of social optimization. A problem is given, and some way to evaluate a proposed solution to it exists in the form of a fitness function[9]. A communication structure or social network is also defined, assigning neighbors for each individual to interact with. Then a population of individuals defined as random guesses at the problem solutions is initialized. These individuals are candidate solutions. They are also known as the particles, hence the name particle swarm. An iterative process to improve these candidate solutions is set in motion. The particles iteratively evaluate the fitness of the candidate solutions and remember the location where they had their best success. The individual's best solution is called the particle best or the local best. Each particle makes this information available to their neighbors.

They are also able to see where their neighbors have had success. Movements through the search space are guided by these successes, with the population usually converging, by the end of a trial, on a problem solution better than that of non-swarm approach using the same methods. Each particle represents a candidate solution to the optimization problem. The position of a particle is influenced by the best position visited by itself i.e. its own experience and the position of the best particle in its neighborhood i.e. the experience of neighboring particles. When the neighborhood of a particle is the entire swarm, the best position in the neighborhood is referred to as the global best particle, and the resulting algorithm is referred to as the gbest PSO. When smaller neighborhoods are used, the algorithm is generally referred to as the lbest PSO. The performance of each particle is measured using a fitness function that varies depending on the optimization problem[19].

Each Particle in the swarm is represented by the following characteristics:

1. The current position of the particle
2. The current velocity of the particle

The particle swarm optimization which is one of the latest evolutionary optimization techniques conducts searches uses a population of particles.

2.OVERVIEW

2.1 Existing System

In the existing system, we implemented the Ant colony optimization technique to optimize the detected e-banking phishing websites. This will improve the correctly classified phishing websites. Ant Colony Optimization (ACO) is a paradigm for designing metaheuristic algorithms for combinatorial optimization problems. The essential trait of ACO algorithms is the combination of prior information about the structure of a promising solution with a posteriori information about the structure of previously obtained good solutions. The functioning of an ACO algorithm can be summarized as follows:

A set of computational concurrent and asynchronous agents (a colony of ants) moves through states of the problem corresponding to partial solutions of the problem to solve. They move by applying a stochastic local decision policy based on two parameters, called *trails* and *attractiveness [12]*. By moving, each ant incrementally constructs a solution to the problem. When an ant after completion or during the construction phase of a solution, it evaluates and modifies the trail value of the components used in its solution. This pheromone information will direct the search of the future ants.

Furthermore, an ACO algorithm includes two more mechanisms*:* trail evaporation and, optionally, daemon actions. Trail evaporation decreases all trail values over time, in order to avoid unlimited accumulation of trails over some component. Daemon actions can be used to implement centralized actions which cannot be performed by single ants, such as the invocation of a local optimization procedure, or the update of global information to be used to decide whether to bias the search process from a non-local perspective. More specifically, an *ant* is a simple computational agent, which iteratively constructs a solution for the instance to solve. Partial problem solutions are seen as *states*. At the core of the ACO algorithm lies a loop, where at each iteration, each ant *moves* (performs a *step*) from a state i to another one ψ, corresponding to a more complete partial solution. Trails are *updated* usually when all ants have completed their solution, increasing or decreasing the level of trails corresponding to moves that were part of "good" or "bad" solutions, respectively [3].

2.2 Objective

The objective is the motivation behind this study is to create a resilient and effective method that uses Data Mining algorithms and tools to detect e-banking phishing websites in an Artificial Intelligent technique. Associative and classification algorithms can be very useful in predicting Phishing websites. We implement the Ant colony optimization algorithm to detect e-banking phishing websites. This will improve the correctly classified phishing websites. We enhance Particle swarm optimization (PSO) which finds a solution to an optimization problem in a search space, or model and predict social behavior in the presence of phishing websites. This will improve the correctly classified phishing websites.

2.3 Proposed System

In the proposed system, we overcome the limitation of ACO like Sequences of random decisions (not independent) and Time to convergence uncertain in the phishing classification. We enhance Particle swarm optimization (PSO) which finds a solution to an optimization problem in a search space, or model and predict social behavior in the presence of phishing websites. This will improve the correctly classified phishing websites.

Particle Swarm Optimization (PSO) is an evolutionary technology (evolutionary computation). Predatory birds originated from the research PSO with similar genetic algorithm is based on iterative optimization tools. Initialize the system for a group of random solutions, through iterative search for the optimal values. However, there is no genetic algorithm with the cross- (crossover) and the variation (mutation). But particles in the solution space following the optimal particle search. The steps detailed chapter on the future of genetic algorithm, the advantages of PSO is simple and easy to achieve without many parameters need to be adjusted. It has been widely used function optimization, neural networks, fuzzy systems control and other genetic algorithm applications [20].Now we are using the same in website phishing field.

3. METHODOLOGIES

3.1. Extracting Phishing Characteristics Attribute

Two publicly available datasets were used to test our implementation: the "phishtank" from the phishtank.com which is considered one of the primary phishing report collators. The PhishTank database records the URL for the suspected website that has been reported, the time of that report, and sometimes further detail such as the screenshots of the website, and is publicly available. We use a java program to extract the above features, and store these in database for quick reference. Our goal is to gather information about the strategies that are used by attackers and to formulate hypotheses about classifying and categorizing of all different e-banking phishing attacks techniques.

3.2. Fuzzification

In this step, linguistic descriptors such as High, Low, Medium, for example, are assigned to a range of values for each key phishing characteristic indicators. Valid ranges of the inputs are considered and divided into classes, or fuzzy sets. For example, length of URL address can range from 'low' to 'high' with other values in between. We cannot specify clear boundaries between classes. The degree of belongingness of the values of the variables to any selected class is called the degree of membership; Membership function is designed for each Phishing characteristic indicator, which is a curve that defines how each point in the input space is mapped to a membership value between [0, 1]. Linguistic values are assigned for each Phishing indicator as Low, Moderate, and high while for e-banking Phishing website risk rate as Very legitimate, Legitimate, Suspicious, Phishy, and Very phishy (triangular and trapezoidal membership function). For each input their values ranges from 0 to 10 while for output, ranges from 0 to 100. The following list consists of the details of criteria and phishing indicators for each criterion [10].

URL & Domain Identity

1. Using IP address
2. Abnormal request URL
3. Abnormal URL of anchor
4. Abnormal DNS record
5. Abnormal URL

Security & Encryption

1. Using SSL Certificate
2. Certificate authority
3. Abnormal cookie
4. Distinguished names certificate

Source Code & Java script

1. Redirect pages
2. Straddling attack

3. Pharming attack
4. On Mouse over to hide the Link
5. Server Form Handler (SFH)

Page Style & Contents

1. Spelling Errors
2. Copying website
3. Using form s with Submit button
4. Using pop-ups windows
5. Disabling right-click

Web Address Bar

1. Long URL address
2. Replacing similar char for URL
3. Adding a prefix or suffix
4. Using the @ Symbols to confuse
5. Using hexadecimal char codes

Social Human Factor

1. Emphasis on security
2. Public generic salutation
3. Buying time to access accounts

3.3 Rule Generation Using Associative Classification Algorithms

To derive a set of class association rules from the training data set, it must satisfy certain user-constraints, ie support and confidence thresholds. Generally, in association rule mining, any item that passes Min-Supp is known as a frequent item. We recorded the prediction accuracy and the number of rules generated by the classification algorithms and a new associative classification MCAR algorithm. Error rate comparative having specified the risk of e-banking phishing website and its key phishing characteristic indicators, the next step is to specify how the e-banking phishing website probability varies. Experts provide fuzzy rules in the form of if…then statements that relate e-banking phishing website probability to various levels of key phishing characteristic indicators based on their knowledge and experience[13]. On that matter and instead of employing an expert system, we utilized data mining classification and association rule approaches in our new e-banking phishing website risk assessment model which automatically finds significant patterns of phishing characteristic or factors in the e-banking phishing website archive data [6,7].

3.4. Aggregation of the Rule Outputs

This is the process of unifying the outputs of all discovered rules. Combining the membership functions of all the rules consequents previously scaled into single fuzzy sets (output).

3.5. Defuzzification

This is the process of transforming a fuzzy output of a fuzzy inference system into a crisp output. Fuzziness helps to evaluate the rules, but the final output has to be a crisp number. The input for the defuzzification process is the aggregate output fuzzy set and the output is a number. This step was done using Centroid technique since it is a commonly used method. The output is e-banking phishing website risk rate and is defined in fuzzy sets like 'very phishy' to 'very legitimate'. The fuzzy output set is then defuzzified to arrive at a scalar value [7, 8].

3.6. Ant Colony Optimization

The characteristic of ACO algorithms is their explicit use of elements of previous solutions. The original idea comes from observing the exploitation of food resources among ants, in which ants' individually limited cognitive abilities have collectively been able to find the shortest path between a food source and the nest.

1. The first ant finds the food source (F), via any way (a), then returns to the nest (N), leaving behind a trail pheromone (b)

2. Ants indiscriminately follow four possible ways, but the strengthening of the runway makes it more attractive as the shortest route.

3. Ants take the shortest route; long portions of other ways lose their trail pheromones.

3.7 Particle Swarm Optimization

We enhance Particle swarm optimization (PSO) which finds a solution to an optimization problem in a search space, or model and predict social behavior in the presence of phishing websites. This will improve the correctly classified phishing websites.

A basic variant of the PSO algorithm works by having a population (called a swarm) of candidate solutions (called particles). These particles are moved around in the search-space according to a few simple formulae. The movements of the particles are guided by their own best known position in the search-space as well as the entire swarm's best known position. When improved positions are being discovered these will then come to guide the movements of the swarm. The process is repeated and by doing so it is hoped, but not guaranteed, that a satisfactory solution will eventually be discovered[14].

Formally, let $f: \mathbb{R}^n \to \mathbb{R}$ be the fitness or cost function which must be minimized. The function takes a candidate solution as argument in the form of a vector of real numbers and produces a real number as output which indicates the fitness of the given candidate solution. The gradient of f is not known. The goal is to find a solution $\mathbf{a}$ for which $f(\mathbf{a}) \leq f(\mathbf{b})$ for all $\mathbf{b}$ in the search-space, which would mean $\mathbf{a}$ is the global minimum. Maximization can be performed by considering the function $h = -f$ instead.

Let S be the number of particles in the swarm, each having a position $\mathbf{x}_i \in \mathbb{R}^n$ in the search-space and a velocity $\mathbf{v}_i \in \mathbb{R}^n$. Let $\mathbf{p}_i$ be the best known position of particle i and let $\mathbf{g}$ be the best known position of the entire swarm. A basic PSO algorithm is then:

- For each particle $i = 1, ..., S$ do:
    - Initialize the particle's position with a uniformly distributed random vector: $\mathbf{x}_i \sim U(\mathbf{b}_{lo}, \mathbf{b}_{up})$, where $\mathbf{b}_{lo}$ and $\mathbf{b}_{up}$ are the lower and upper boundaries of the search-space.
    - Initialize the particle's best known position to its initial position: $\mathbf{p}_i \leftarrow \mathbf{x}_i$
    - If $(f(\mathbf{p}_i) < f(\mathbf{g}))$ update the swarm's best known position: $\mathbf{g} \leftarrow \mathbf{p}_i$
    - Initialize the particle's velocity: $\mathbf{v}_i \sim U(-|\mathbf{b}_{up}-\mathbf{b}_{lo}|, |\mathbf{b}_{up}-\mathbf{b}_{lo}|)$
- Until a termination criterion is met (e.g. number of iterations performed, or adequate fitness reached), repeat:
    - For each particle $i = 1, ..., S$ do:

- Pick random numbers: r_p, r_g ~ $U(0,1)$
- Update the particle's velocity: $\mathbf{v}_i \leftarrow \omega\, \mathbf{v}_i + \varphi_p\, r_p\, (\mathbf{p}_i\text{-}\mathbf{x}_i) + \varphi_g\, r_g\, (\mathbf{g}\text{-}\mathbf{x}_i)$
- Update the particle's position: $\mathbf{x}_i \leftarrow \mathbf{x}_i + \mathbf{v}_i$
- If $(f(\mathbf{x}_i) < f(\mathbf{p}_i))$ do:
  - Update the particle's best known position: $\mathbf{p}_i \leftarrow \mathbf{x}_i$
  - If $(f(\mathbf{p}_i) < f(\mathbf{g}))$ update the swarm's best known position: $\mathbf{g} \leftarrow \mathbf{p}_i$
- Now **g** holds the best found solution[16].

3.8 Performance Comparison

The performance analysis of the proposed system is compared with the existing system with the performance metrics mentioned.

Error rate: The proposed algorithm will get the less error rate when compared to the existing algorithm.

Correct prediction: the proposed algorithm predicts the phishing website more accurate than the existing algorithm.

3.9 Pseudocode Web Phishing

Input: Webpage URL
Output: Phishing website identification
Step 1: Read web phishing URL
Step 2: Extract all 27 feature
Step 3: For each feature, Assign fuzzy membership degree value and Create fuzzy data set
Step 4: Apply association rule mining & generate classification rule.
Step 5: Aggregate all rule above minimum confidence.
Step 6: De-fuzzification of fuzzy values into original values [19].

4. IMPLEMENTATION

4.1 Ant Colony Optimization

The **ant colony optimization** algorithm (ACO), is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs. This algorithm is a member of **ant colony algorithms** family, in swarm intelligence methods, and it constitutes some meta-heuristic optimizations.

In a series of experiments on a colony of ants with a choice between two unequal length paths leading to a source of food, biologists have observed that ants tended to use the shortest route. A model explaining this behavior is as follows:

1. An ant (called "blitz") runs more or less at random around the colony;

2. If it discovers a food source, it returns more or less directly to the nest, leaving in its path a trail of pheromone;
3. These pheromones are attractive, nearby ants will be inclined to follow, more or less directly, the track;
4. Returning to the colony, these ants will strengthen the route;
5. If two routes are possible to reach the same food source, the shorter one will be, in the same time, traveled by more ants than the long route will
6. The short route will be increasingly enhanced, and therefore become more attractive;
7. The long route will eventually disappear, pheromones are volatile;
8. Eventually, all the ants have determined and therefore "chosen" the shortest route [11].

The design of an ACO algorithm implies the specification of the following aspects.

- An environment that represents the problem domain in such a way that it lends itself to incrementally building a solution to the problem.
- A problem dependent heuristic evaluation function, which provides a quality measurement for the different solution components.
- A pheromone updating rule, which takes into account the evaporation and reinforcement of the trails.
- A probabilistic transition rule based on the value of the heuristic function and on the strength of the pheromone trail that determines the path taken by the ants.
- A clear specification of when the algorithm converges to a solution [17].

The ant system simply iterates a main loop where *m* ants construct in parallel their solutions, thereafter updating the trail levels. The performance of the algorithm depends on the correct tuning of several parameters, namely: a, b, relative importance of trail and attractiveness, r, trail persistence, t*ij*(0), initial trail level, *m*, number of ants, and Q, used for defining to be of high quality solutions with low cost[8]. The ANTS algorithm is the following.

1**.** Compute a (linear) lower bound LB to the problem Initialize t*iy* ("i,y) with the primal variable values .

2. For k=1,m (m= number of ants) do repeat

2.1 compute h*iy* "(iy)

2.2 choose in probability the state to move into.

2.3 append the chosen move to the *k*-th ant's tabu list **until** ant *k* has completed its solution

2.4 carry the solution to its local optimum **end for**

3. For each ant move (iy), compute Dt*iy* and update trails by means of (5.6)

4. If not (end test) **go to** step 2.

4.2 Particle Swarm Optimization (pso)

The PSO algorithm has become an evolutionary computation technique and an important heuristic algorithm in recent years. The main concept of PSO originates from the study of fauna behavior. PSO learned from such a scenario and used it to solve the optimization problems. In PSO, each single solution is a "particle" in the search space. We refer to each solution as a "particle.[15]" All particles have fitness values, which are evaluated by the fitness function to be optimized. The particles also have velocities which direct the flight of the particles. Particles fly through the problem space by following the current optimum particles.

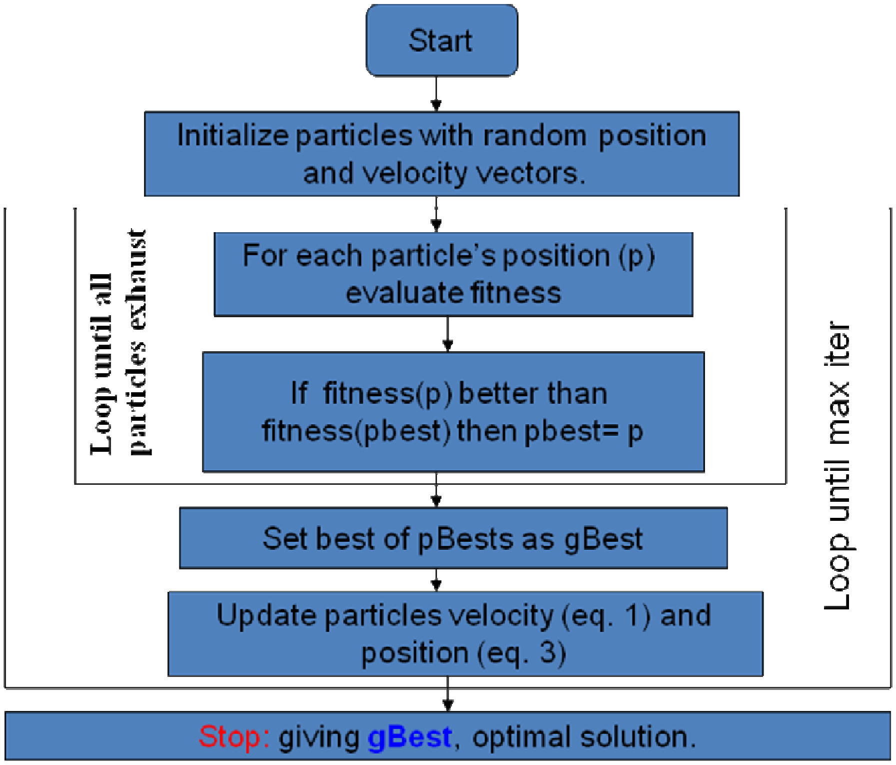

Start
Initialize particles with random position and velocity vectors.
For each particle's position (p) evaluate fitness
If fitness(p) better than fitness(pbest) then pbest= p
Loop until all particles exhaust
Set best of pBests as gBest
Update particles velocity (eq. 1) and position (eq. 3)
Loop until max iter
Stop: giving gBest, optimal solution.

FIGURE 4.1 - Data flow for finding optimal solution

As shown in the Fig.4.1, PSO is initialized with a group of random particles (solutions) and then searches for optima by updating generations. During all iterations, each particle is updated by following the two "best" values. The first one is the best solution (fitness) it has achieved so far. The fitness value is also stored. This value is called "pbest." The other "best" value that is tracked by the particle swarm optimizer is the best value obtained so far by any particle in the population. This best value is a global best and is called "gbest" [10].

5. RESULTS AND DISCUSSION

There is a significant relation between the two phishing website criteria's *(URL* & *Domain Identity)* and *(Security* & *Encryption)* for identifying e-banking phishing website. Also found insignificant trivial influence of the *(Page Style* & *Content)* criteria along with *(Social Human Factor)* criteria for identifying e-banking phishing websites. Particle Swarm Optimization produces more accurate classification models than Associative classifiers. We recorded the prediction accuracy and the number of rules generated by the classification algorithm, the Ant Colony algorithm and PSO algorithm.

Table 5.1 shows that the PSO produce more accuracy and less time taken than associative classifier and ACO. Selected 802 cases randomly used for inducing rules from 1050 cases in original data set, the remaining 300 cases are used for testing accuracy of the induced rules of the proposed method by measuring the average percentage of correct predictions.

TABLE 5.1 Prediction accuracy and time taken comparison

Method	Association Classification	
Test Mode	10 Fold Cross Validation	
No. of URLs	1052 (Both Genuine and Fake)	
Correct Classified	1006	
Incorrect Classified	46	
Optimization	**Accuracy**	**Time Taken**
Association Classification	81%	12ms
ACO Optimization	89%	11ms
PSO Optimization	91%	9ms

The Fig.5.1 shows the comparision of fuzzy associative classifiers, ant colony optimization and PSO with the error rate.

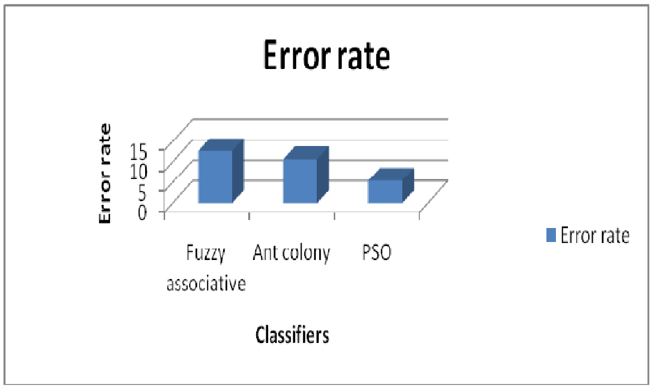


FIGURE: 5.1- Error rate comparision

The Fig.5.2 shows the comparision of ant colony algorithm and fuzzy associative algorithms in terms of prediction accuracy.

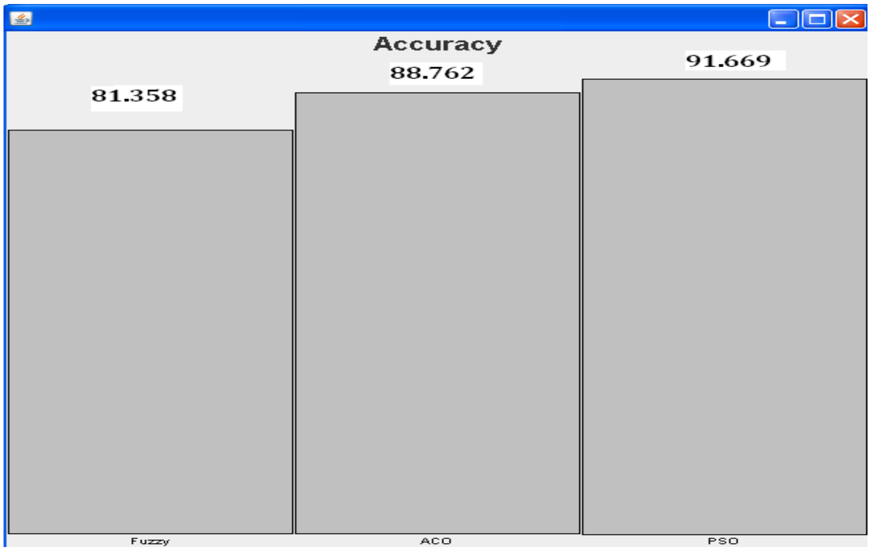


FIGURE : 5.2 - Accuracy report

From Fig.5.3 we can see the effectiveness of PSO implementation which shows difference from other methods.

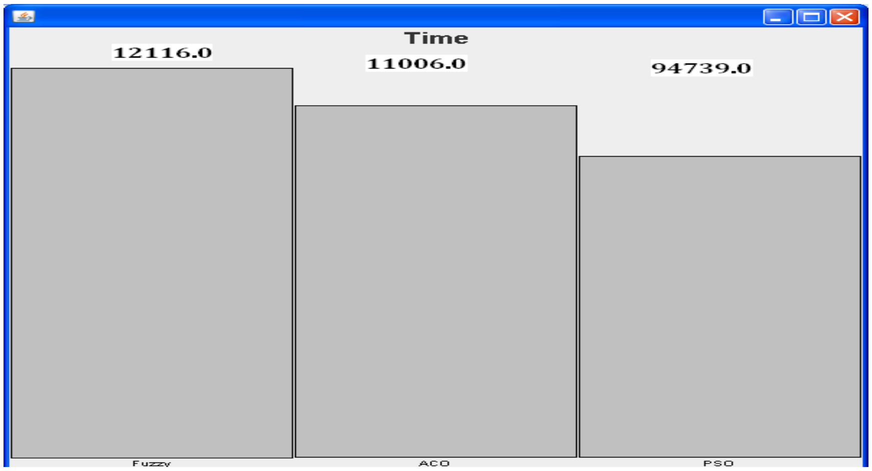


FIGURE : 5.3 – Time difference between three methods

6.CONCLUSION

The Associative Classification Algorithm with Particle Swarm Optimization Technique for e-banking phishing website detection model is outperformed when compared with existing classification algorithms in terms of prediction accuracy and error rate. Particle swarm optimization (PSO) is an algorithm modelled on swarm intelligence, that finds a solution to an optimization problem in a search space, or model and predict social behaviour in the presence of objectives. The PSO algorithm for e-banking phishing website model showed the significance importance of the phishing website in two criteria's (URL & Domain Identity) and (Security & Encryption) with insignificant trivial influence of some other criteria like 'Page Style & content' and 'Social Human Factor'. Combining these two techniques has given a fruitful result. After more than 1050 websites detection for both its application effectiveness and its theoretical groundings, PSO became one of the most successful paradigms in network security.

7. REFERENCES

[1] A. Hossain, M. Dorigo, Ant colony optimization web page, http:// iridia.ulb.ac.be / mdorigo/ACO/ACO.html N. Ascheuer, Hamiltonian path problems

[2] Ant Colony Optimization, Vittorio Maniezzo, Luca Maria Gambardella, Fabio de Luig**i.**

[3] Optimizing Large Scale Combinational Problems Using Multiple Ant Colonies Algorithm based on Pheromone Evaluation technique, Alaa Aljanaby, Ku Ruhana Ku Mahamud,

[4] Associative Classification Techniques for predicting e-Banking Phishing Websites, Maher Aburrous Dept. of Computing ,Universit y of BradfordBradford, UK.

[5] B. Adida, S. Hohenberger and R. Rivest , "Lightweight Encryption for Email," USENIX Steps to Reducing Unwanted Traffic on the Internet (SRUTI), 2005 ,

[6] Bing Liu, Wynne Hsu, Yiming Ma, "Integrating Classification and Association Rule Mining." Proceedings ofthe Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98, Plenary Presentation), New York, USA.

[7] GARTNE R, INC. Gartner Says Number of Phishing Emails Sent to U.S. Adults Nea rly Doubles in Just Two Years, http //www .gartner. com/ it/pag e.jsp3.

[8] Gartner. "UK phishing fraud losses double" STAMFORD, Conn., (April 14, 2009). "Gartner Says Number of Phishing Attacks on U.S. Consumers Increased 40 Percent in 2008".. Finextra. March 7, 2006. http:// www. finextra. com/ fullstory asp?id=15013.

[9] Jaco F. Schutte "The Particle Swarm Optimization Algorithm" EGM 6365 - Structural Optimization Fall 2005

[10] L. Bianchi, L.M. Gambardella, M.Dorigo. An ant colony optimization approach to the probabilistic traveling salesman problem. In Proceedings of PPSN-VII, Seventh InterGARTNE R, INC.

[11] M. E. Bergen, Technische Universität Berlin, Germany, 1995 Canstraint-based assembly line sequencing, Lecture Notes in Computer

[12] Miller, Rich. "Bank, Customers Spar Over Phishing Losses". *Netcraft*. http://n ews.netcraft .com/ rchives/ 2006/09.

[13] Mining Fuzzy Weighted Association Rules Proceedings of the 40th Hawaii International Conference on System Sciences – 2007.

[14] Particle Swarm Optimization , www.swaminteLligence.org.

[15] Particle Swarm Optimization, WIKI Pedia.

[16] Richardson, Tim (May 3, 2005). "Brits fall prey to phishing". The Register. http:/ /www .theregister.co.uk/2005/05/03/aol_phishing/.

[17] T.Moore and R. Clayton, "An empirical analysis of the current state of phishing attack and defence", In Proceedings of the Workshop on the Economics of Information Security (WEIS2007)

[18] WEKA - University of Waikato, New Zealand, EN,2006: "Weka -Data Mining with Open Source Machine Learning Software in Java", 2006 ,

[19] Xun Dong,"PSO Introduction" Department of Computer Science University of York, United Kingdom Email: xundong@cs.york.ac.uk,

Maintenance of Network Connectivity Across Process Migration

Mehdi Bahrbegi *m.bahribayli@gmail.com*
Department of Computer
Islamic Azad University, Shabestar Branch
Tabriz, East Azerbaijan, Iran

Hadi Bahrbegi *hadi.bahrbegi@gmail.com*
Department of Computer
Islamic Azad University, Shabestar Branch
Tabriz, East Azerbaijan, Iran

Amir Azimi Alasti Ahrabi *amir.azimi.alasti@gmail.com*
Department of Computer
Islamic Azad University, Shabestar Branch
Tabriz, East Azerbaijan, Iran

Elnaz Safarzadeh *elnaz_safarzadeh@yahoo.com*
Department of Computer
Islamic Azad University, Shabestar Branch
Tabriz, East Azerbaijan, Iran

Abstract

Most of processes running on a computer need network connections to communicate with other processes and access resources such as files, databases, and so on. Network addressing protocols are tightly coupled with physical addresses. When a process migrates, all of these connections are torn. While these connections are essential to running a process, the process may crash. And process migration becomes an inapplicable approach. We have developed a new method which maintains network connectivity across process migration. Our solution is based on API interception and removes the need for modification of operating system or applications. It also removes disadvantages of other approaches that are based on the API interception and offers a better performance.

Keywords: Network Connectivity, Process Migration, Twin.

1. INTRODUCTION

Problem of network connectivity in the systems that use process migration as solution to problems such as load sharing has existed from the early advent of process migration. Solutions typically consist of modifications to operating systems or applications. Because of complexity of adding transparent migration to systems originally designed to run stand-alone, since designing new systems with migration in mind from the beginning is not a realistic option anymore [1] at one hand and need to developing new applications because of impossibility of use of existing applications, has made process migration an unpopular approach.

Because of the increasing costs of operating system development and the lack of standard solutions for distributed systems and heterogeneity, middleware level solutions have be-come of more interest [3].

Our approach is to create a process called twin for each remote process. The twin is responsible for redirecting communications of the remote process. Every remote process has logically its own twin. But in practice there is not a one to one mapping between remote processes and twins. A twin can be responsible for more than one remote process. This approach works with existing operating systems and applications.

2.1 Terminology

A process is a key concept in operating systems [5]. It consists of data, a stack, register contents, and the state specific to the underlying Operating System (OS), such as parameters related to process, memory, and file management. A process can have one or more threads of control. Threads, also called lightweight processes, consist of their own stack and register contents, but share a process's address space and some of the operating-system-specific state, such as signals. The task concept was introduced as a generalization of the process concept, whereby a process is decoupled into a task and a number of threads. A traditional process is represented by a task with one thread of control [1].

Process migration is the act of transferring a process between two machines (the source and the destination node) during its execution [1].

The node that process is instantiated in is called home node and the node that process is migrated to, is called a host node. A process that is running away from its home is referred as remote or foreign process.[1]

For every migrated process there exists a special process running at home which is called twin. It is possible that a twin be responsible for redirecting of communications of more than one remote process.

2. RELATED WORKS

Xiaodong Fu et al. [2] and Tom Boyd et al. [4] have introduced a method to maintain network connectivity for process migration and dynamic cluster environments without need to modify operating system or applications. This work introduces another approach that minimizes disadvantages of previous works.

3. NETWORK CONNECTIVITY

Nearly all of addressing protocols are tightly coupled with physical addresses. File addresses in file systems, NetBIOS names, and IP addresses to name a few. It means that a process is tied to a specific location and if it moves to another location, it will lose all of its connections to the resources which, is connected to. This is in contrariety with migration.

Previous approaches for maintaining network connectivity fall into two broad categories: modification to the OS network protocol stack, and introduction of a new API for accessing underlying resources [2] The first approach results in modifications of OS structures to support migration [6, 7, 8, 9], and second approach results in modification or rewriting the application and requiring the use of a new application programming interface (API)) [10, 11, 12, 13] whose implementation isolates the application from the consequences of migration . Moreover, most such solutions do not address the issue of adapting to changes in resource characteristics.[2] which are not desirable solutions.

Xiaodong Fu et al. [2] and Tom Boyd et al. [4] introduced a method for transparent network connectivity .They have developed a layer that operates transparently between the application and the underlying communication layer or operating system. This layer interfaces with the application and the operating system using API interception techniques [14, 15] to create a middleware to redirect communications transparently [2]. This middleware was called 'agent' and removed the need for modification of application or operating system. Disadvantage to this approach is that the middleware possibly becomes a bottleneck. It means that a single process (the agent) is unable to undertake redirecting communication for all remote processes. It is also prone to the famous problem of distributes systems, 'single point of failure' – if agent crashes all connections will be lost.

Twins use the same technology to intercept API calls, but because there exist theoretically a twin for each remote process the problems of 'bottleneck' and 'single point of failure' is removed.

4. TWINS

When a process asks for a network connection, it is bound to a specific address (for example an IP and Socket in TCP/IP protocol). After migration the packets are still sent to the old address and will be lost. A twin is a ready made lightweight process which is instantiated when a process asks for a network connection for first time. After the process is migrated packets which are sent to the old address are redirected to the new address by its twin. The main idea is taken from Mobile IP protocol. (For more detail about Mobile IP Protocol refer to [1, 16]).

4.1 Twin Implementation

Implementation of twins is very similar to the implementation of the agents in [2]. It is illustrated in Figure 1.

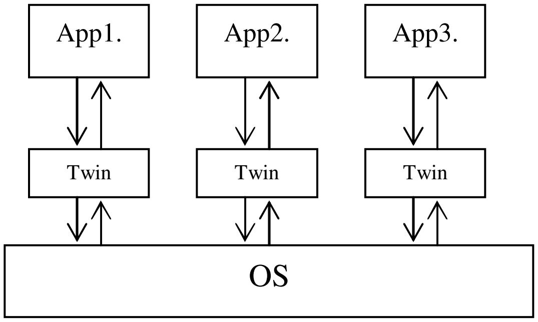


FIGURE 1: Structure of twins.

There is just one twin for each process but inverse is not necessarily true. After twin is created and connection is establish between twin and process every thing goes normally. Twin plays a dual role. It acts as the process at other end of connection for its twin and acts as its twin for the process at the other end.

Every twin listens to a specific port called control port. This port is used by migration manager.

4.2 Whole Story

Scenario of Creating a Network Connection

Process calls the API to create a network connection.

Migration Manager intercepts the API call and looks up in the Table of Twins. If there is not a twin for that process, it creates a twin for that process and inserts handle of process and control port of its twin in the Table of Twins. Then Migration Manger sends a CREATE_NEW_CONNECTION message to the twin's control port. Twin returns a network connection handle. Migration Manager hands this handle to the process.

Scenario of Migration

Migration Manager decides to send a process – which has one or more active network connections – to another node.

Migration Manager looks up in the Table of Twins for its twin then sends a MIGRATING message to its twin. While the twin has not received MIGRATED message it will buffer all incoming packets.

After migration ends Migration Manager moves corresponding record of the Table of Twins to the destination node then sends a MIGRATED message to the twin. One of parameters for MIGRATED message is new address for remote process.

5. FUTURE WORKS

Creating a separate twin for each process consumes a large amount of resources; so it is a good practice to design twins in a way that a twin can be responsible for more than one process. This idea can be combined with other techniques to gain even a better performance. For example twins can be pooled. Hence there is no need to kill and create twins over and over.

6. CONCLUSION

Figure 2 illustrate the process migration's grows tree .

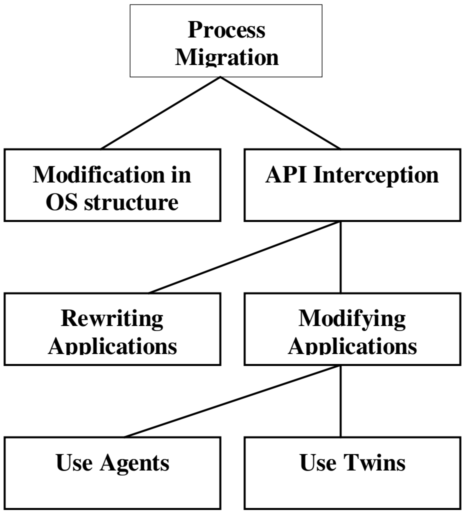


FIGURE 2: Process migration's growth tree.

Modifying stack protocol to implement process migration is not ignorable .This modification results in modification to the core of operating system and it is not desirable in process migration [2].Using API interception solves above problem . This solution operates in two distinct categories, rewriting applications and rebuilding (modifying) applications. High cost of rewriting application cause to this approach also is not desirable to use in process migration. The recent development in rebuilding applications cause to process migration using agents (this paper and [2, 4]). This approach also have problems that we explain in this paper .The suggested approach in this paper solved two major problems in process migration(bottleneck and single point of failure) and this approach cause to commonality and reduce costs of process migration.

7. SUMMARY

We have described a new technique to maintain network connectivity across migration of process. This approach has removed disadvantages of previous solutions.

8. REFERENCES

[1] Dejan S. Milojicic et al., "Process Migration", ACM Computing Surveys, Vol. 32, No. 3, September 2000.

[2] Xiaodong Fu et al., "Transparent Network Connectivity in Dynamic Cluster Environment", Proceedings of the 4th International Workshop on Network-Based Parallel Computing: Communication, Architecture, and Applications, 2000.

[3] Bernstein, P. A., "Middleware: A Model for Distributed System Services", Communications of the ACM, Vol. 39, Issue 2, pp. 86–98, 1996.

[4] Tom Boyd et al., "Process Migration: A Generalized Approach Using a Virtualizing Operating System", 22nd International Conference on Distributed Computing Systems, , 2002.

[5] Tanenbaum, "Modern Operating Systems", Prentice Hall, Englewood Cliffs, New Jersey, 1992.

[6] Douglis, F. and Ousterhout, J. "Process migration in the sprite operating system", In Proceedings of 7th International Conference on Distributed Computing Systems, pp. 18–25, 1987.

[7] Paoli, D. and Goscinski, A. "The RHODOS Migration Facility", Technical Report TR C95/36, School of Computing and Mathematics, Deakin University, 1995.

[8] Rozier, M., Abrossimov, V., Gien, M., Guillemont, M., Hermann, F., and Kaiser, C. Chorus "Overview of the Chorus distributed operating system". In Proceedings of USENIX Workshop on Micro-Kernels and Other Kernel Architectures, pp. 39–70, 1992.

[9] Milojicic, D., Zint, W., Dangel, A., and Giese, P. "Task migration on the top of the mach microkernel", In Proc. of the 3rd USENIX Mach Symp., pp. 273–289, 1993.

[10] Baratloo, A., Dasgupta, A., and Kedem, Z. "Calypso: A novel software system for fault-tolerant parallel processing on distributed platforms", In Proc. of 4th IEEE Intl. Symp. on High Performance Distributed Computing, 1995.

[11] Blumofe, R., Joerg, C., Kuszmaul, B., Leiserson, C., Randall, K., and Zhou, Y. Cilk: "An efficient multithreaded runtime system", In 5th ACM SIGPLAN Symp. on Principles and Practice of Parallel Programming, pp. 207–216, 1995.

[12] Birman, K. "Replication and fault-tolerance in the ISIS system", In Proc. of 10th ACM Symp. on Operating System Principle, pp. 79–86, 1985.

[13] Hayden, M. "The Ensemble System", Technical Report TR98-1662, Cornell University, 1998.

[14] Hunt, G. and Brubacher, D. "Detours: Binary interception of Win32 functions", Technical Report MSR-TR- 98-33, Microsoft Research, 1999.

[15] Balzer, R. "Mediating Connectors", In Proc. of ICDCS Middleware Workshop, 1999.

[16] Milojicic, D., Douglis, F., Wheeler, R. "Mobility: Processes, Computers, and Agents", Addison-Wesley Longman and ACM Press, 1999.

Detection of Botnets Using Honeypots and P2P Botnets

Rajab Challoo kfrc000@tamuk.edu
Dept. of Electrical Engineering & Computer Science
Texas A&M University Kingsville
Kingsville, 78363-8202, USA

Raghavendra Kotapalli raghavsan@gmail.com
Dept. of Electrical Engineering & Computer Science
Texas A&M University Kingsville
Kingsville, 78363-8202, USA

Abstract

A “botnet” is a group of compromised computers connected to a network, which can be used for both recognition and illicit financial gain, and it is controlled by an attacker (bot-herder). One of the counter measures proposed in recent developments is the “Honeypot”. The attacker who would be aware of the Honeypot, would take adequate steps to maintain the botnet and hence attack the Honeypot (Infected Honeypot). In this paper we propose a method to remove the infected Honeypot by constructing a peer-to-peer structured botnet which would detect the uninfected Honeypot and use it to detect botnets originally used by the attacker. Our simulation results show that our method is very effective and can detect the botnets that are intended to malign the network.

Keywords: Peer-to-peer network, Botnet, Honeypot, Hijacking.

1. INTRODUCTION

The Increase in the Internet malware in the recent attacks have attracted considerable amount of attraction towards botnets. Some of them include Email spamming, Key logging, click fraud and traffic sniffing [1]. Recently detected dangerous botnets include Mariposa (2008), officla (2009) and TDSS (2010). The scatter attacks done by the bot controllers using a program called bot which communicates with other botnets and receive the commands from Command and Control servers [3].

As the traditional botnets, which are designed to operate from a central source (bot-attackers machine) which can be shutdown if the source is pin-pointed by the security agencies, bot masters use or resort to peer to peer (P2P) botnets which do not have a centralized source and can grow at an alarming speed. For example, botnet Oficla can spam up to 3.6 billion targets per day [4].

In this paper we show how the use of a combination of Honeypots and Peer to Peer botnet to defend the attacks from other botnets. In order to improve the efficacy in defending against such malicious attacks, one needs to analyze the botnets from a bot-attackers perspective. This would require a study of basic structure of botnet and the network. The antivirus approach, of signature based detection of removing one bot or virus at a time works at host level but when bot-attackers use polymorphic methods creating new instances using the botcodes, evasion from antivirus becomes complicated. Security experts monitor Command and Control (C&C) traffic so as to detect an entire network which is infected, this is done to extenuate the botnet problem on a large scale by identifying the C&C channel[5]. Once a C&C channel is identified by the defenders, entire botnet could be captured by the defenders[3]. After botnet is captured, botmasters move to an advanced technique.

2. BACKGROUND

To mitigate the botnet problem, the command and control mechanism has been under study which determines the structure of C&C botnets that can monitor, hijack and shutdown the network. Defenders can however shutdown the entire C&C channel and prevent the attack [5]. In P2P botnets there is no central point for controlling the botnets. The servant bots act as client and servers [6], and accept both incoming and outgoing connections whereas the client bots do not accept incoming connections. Servant bots alone are added to the peer-lists. All bots including both client and server bots contact the servant bots to retrieve the commands [4].

2.1 Types of Botnets

Bots are basically classified into three types based on botnet topologies:
centralized, peer to peer (P2P) and random .

As described earlier, centralized bot has a point of control which shuts down the entire botnet if affected. In random botnet, one bot knows no more than the other [7]. This type of botnet has no guarantee of delivering what is required, hence the topology of random botnet is not discussed in this paper.

Peer-to-peer botnet has no central point of control and can be used by botmasters (if not in use already). Let us consider the P2P botnet constraints from a botmasters perspective [3].

1) Generate a botnet that is capable of controlling remaining bots in the network, even after considerable portion of botnet population has been removed by defenders.

2) Prevent significant exposure of the network topology when some bots are captured by defenders.

3) Monitor and obtain the complete information of a botnet by its botmaster with ease.

4) Prevent (or make it harder for) defenders from detecting bots via their communication traffic patterns.

2.2 Monitoring Botnets

Currently, there are two techniques to monitor botnets

(1) Allow Honeypots to be compromised by the botnet [2, 8], behave as normal 'bot" in the botnet, and these Honeypot spies provide all the required information to monitor botnet activities [2].

(2) To hijack the bot controllers to monitor Command and Control communications in Botnets.

The architecture of a P2P botnet is shown in Figure 1 [2].

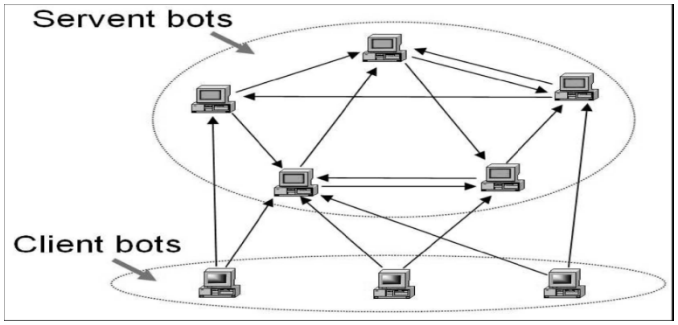


FIGURE 1: C&C Architecture of a P2P Botnet

3. PROPOSED APPROACH

Command and Control botnets, P2P botnets and Honeypots (Honeynets) are used in our approach. Consider bots A, B, C, D and E that are introduced into the network as shown in Figure 2. Honeypots are denoted as H, IH denotes the Infected Honeypot from botnets either C&C or P2P or both.

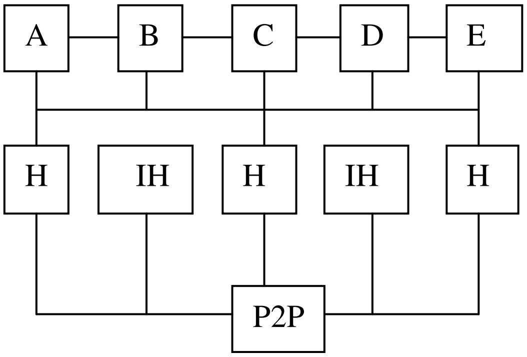


FIGURE 2: Block diagram (Using P2P botnets to detect Honeypots).

Proposed method has three steps. Explained as follows:

- Bots A, B, C, D, and E are launched into the network creating a Honeypot-aware attack,
- Bots infect the Honeypots in the network, thereby leaving infected Honeypots (IH) and uninfected Honeypots (H) in the cluster.
- Third step involves removing the Infected Honeypots (IH) using P2P botnet. Hence, uninfected Honeypots (H) can now be used in detecting bots A to E.

The P2P botnet which is constructed using peer list updating procedure, can be constructed in two parts,

The first part consists of a host which is vulnerable and later decides whether this is Honeypot or not, the second part contains the block of information and the authorization component allowing the infected host to join in the botnet.

Another honeypot-based monitoring occurrence happens during peer-list updating procedure. First, defenders could let their honeypot bots claim to be servant bots in peer-list updating. By doing this, these honeypots will be connected by many bots in the botnet, and hence, defenders are able to monitor a large fraction of the botnet. Second, during peer-list updating, each honeypot bot could get a fresh peer-list, which means the number of bots revealed to each honeypot could be doubled.

A honeypot could be configured to route all its outgoing traffic to other honeypots; at the same time, the trapped malicious code still believes that it has contacted some real machines. The P2P botnet constructed as introduced above is easy for attackers to control when facing monitoring and defense from security defenders. First, an attacker can easily learn how many zombie machines have been collected in the botnet and their IP addresses. The attacker can connect to several known infected computers, asking them to issue a command to let all bots sending a specific service request to the attacker’s sensor. On the other hand, security professionals cannot use this technique for monitoring, even if they know how to send such a command, due to their liability constraint. Second, an attacker can randomly choose any one or several bots to infill commands into the botnet—it is very hard for security defenders to cut off the control channel unless they hijack the botnet and take control of it by themselves. Such an active defense requires security professionals to issue commands to the botnet and update bot code on all (potentially hundreds or even thousands) compromised computers, which clearly puts a heavy liability burden on security professionals. Third, suppose security professionals remove many infected computers in a botnet. The attacker still has control over the remaining P2P botnet, even if the remaining botnet is broken into many separated smaller ones.

Security defenders could also try to distinguish which outgoing traffic is for honeypot detection and which outgoing traffic is for a real attack. If this could be done, then honeypots could be configured to allow the honeypot-detection traffic to be sent while blocking all other malicious traffic. For this purpose, security defenders will need to conduct more research on practically implementing automatic binary code analysis in honeypots. Internet security attack and defense is an endless war. From the attackers' perspective, there is a trade-off between detecting honeypots in their botnets and avoiding bot removal by security professionals. If an attacker conducts honeypot-aware test on a botnet frequently, honeypots in the botnet can be detected and removed quickly. But at the same time, the bots in the botnet will generate more outgoing traffic, and hence, they have more chance to be detected and removed by their users or security staff. In the end, we should emphasize that even if attackers can successfully detect and remove honeypots based on the methodology presented in the paper, there is still significant value in honeypot research and deployment for detecting the infection vector and the source of attacks. It may not be possible for honeypots to join the botnet, but the security hole used to facilitate the infection can be quickly discovered and patched.

4.0 DEFENSES AGAINST BOTNETS

Defense from Botnets can be divided to: prevention, detection and removal. In this section we provide information on how the user's system can be protected from botnet attacks. The botnet detected in the system must be removed immediately, otherwise it might cause other problems such as slow-down the performance, loss of data and leaking of information to the web.

Applying a patch if any damage occurs, is out of the context of this method. This method can recognize (fingerprint) botnets from the traffic and block the traffic, both upstream and downstream. The role of Honeypots here is to behave like servant bots to the botnets that intrude into the network. The moment the Honeypot bot is included into the botnet architecture, the monitoring of botnet activity is initiated by the defenders. The defenders can get many spying botnets into their hands so that they can monitor the commands given by the botmasters or send fake commands to the botmasters, leading to a trap since a remote code authentication (the problem of identifying a remote code program) [3] cannot distinguish the honeypot bots from the botnets from the botmaster's point of view.

4.1 Simulation and Analysis Using P2P botnets and Multiple Honeypots

Multiple Honeypots are used in synchronization with P2P botnets. It should be noted that there is a limitation of how many honeypots can be used in the cluster for efficient monitoring and detection. The number can be calculated based on an average of 50 to 100 simulation. The Simulation results are shown in Figure 3.

An analytical model can be derived by estimating the mean value of tracked bots, which is denoted by T [$B_{tracked}$]. There are h number of uninfected honeypots joining the cluster before the peer-list is updated [1]. Let the size of the peer-list be P, the final botnet has *l* number of botnets, and the number of servant bots used in peer-list updating procedure is Q.

Since the botnet has l bots, the average number of tracked bots can be calculated by

$$T\,[B_{\text{tracked}}] = l\left[1-\left(1-\frac{h}{Q}\right)^{P}\right]$$

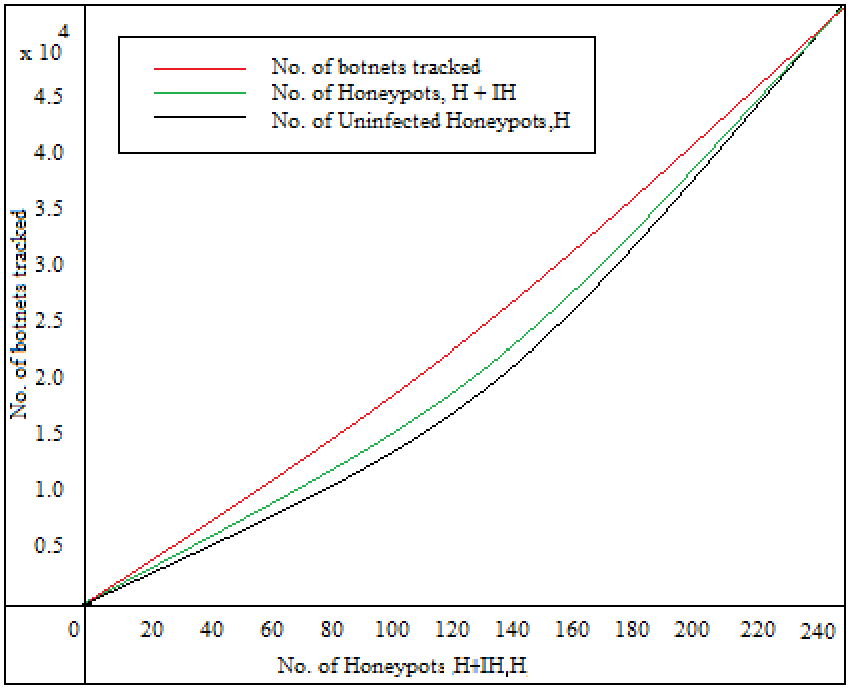


FIGURE 3: Simulation results for botnets tracked, total number of honeypots and uninfected honeypots

In Figure 3, number of Honeypots (H+IH) versus number of botnets tracked represented in the green curve and the number of uninfected honeypots denoted by H versus number of botnets tracked is represented in black curve. The number of botnets tracked versus the uninfected honeypots, H represented in red curve shows a better performance in detection after botnets are tracked.

4.2 Discussion

From the simulation and the above description on defenses against botnets, we can see that Honeypot based detection plays a vital role in the detection of botnets. The use of a robust P2P botnet to filter the infected honeypots from the network adds more to the defenders advantage as shown in Figure 3. In the future, Botmasters could design advanced ways to detect the honeypot defense system which could include fingerprinting (recognition). Attackers could also exploit the legal and ethical constraints held by the defenders [2]. The proposed method works owing to remote code authentication [3] which helps in not distinguishing the botnet from the honeypot bot.

In the future, if remote authentication method is compromised then the war between botmasters and security community would intensify. The research done until now in botnets shows “the worms” and botnets in the internet can be monitored but it gets harder to stop the attacks even after the presence of the threat is known, i.e. it should be detected without inflicting any damage to the data. Ethical and legal reasons in the prevention of botnet attacks turn out to be resource consuming [1]. The other methods which involve detection of botnets without Honeypots by using a botnet monitoring sensor is also considered which gives a clear picture on botnet activity but once the botmasters destroy the sensor, the machine or target will be infected. Our proposed method illustrates the design is practical and can be implemented by defenders with little complexities.

5 : CONCLUSION

Due to their potential for illicit financial gain, “botnets” have become popular among Internet attackers in recent years. As security defenders build more honeypot-based detection and defense systems, attackers will find ways to avoid honeypot traps in their botnets. Attackers can use software or hardware specific codes to detect the honeypot virtual environment [6, 7, 16], but they can also rely on a more general principle to detect honeypots: security professionals using honeypots have liability constraints such that their honeypots cannot be configured in a way that would allow them to send out real malicious attacks or too many malicious attacks. In this paper, we introduced a means for defending the network from botnets, when Honeypots are infected and then deploy a P2P botnet which would act as a filter to remove the infected Honeypots which remain in the Network Cluster and hence the uninfected Honeypots can be used with efficacy to defend the network. Honeypot research and deployment is important and should continue for the security community, but we hope this paper will remind honeypot researchers of the importance of studying ways to build covert honeypots, and the limitation in deploying honeypots in security defense. The current popular research focused on finding effective honeypot-based detection and defense approaches will be for naught if honeypots remain as easily detectible as they are presently.

6: REFERENCES

[1] P. Wang, S. Sparks, and Cliff C. Zou, “An Advanced Hybrid Peer-to-Peer Botnet,” IEEE; Vol. 7, No. 2, April-June 2010.

[2] Cliff C. Zou, Ryan Cunningham, “Honeypot-Aware Advanced Botnet Construction and Maintenance,” IEEE Computer society; Proceedings of the 2006 International Conference on Dependable Systems and Networks (DSN’06).

[3] Chia-Mei Chen, Ya-Hui Ou, and Yu-Chou Tsai, “Web Botnet Detection Based on Flow Information,” Department of Information Management, National Sun Yat –Sen University, Kaohsiung, Taiwan; IEEE 2010.

[4] D. Dagon, C. Zou, and W. Lee, “Modeling Botnet Propagation Using Time Zones,” Proc. 13th Ann. Network and Distributed System Security Symp. (NDSS ’06), pp. 235-249, Feb. 2006.

[5] A. Ramachandran, N. Feamster, and D. Dagon, “Revealing Botnet Membership Using DNSBL Counter-Intelligence,” Proc. USENIX Second Workshop Steps to Reducing Unwanted Traffic on the Internet (SRUTI ’06), June 2006.

[6] J.R. Binkley and S. Singh, “An Algorithm for Anomaly-Based Botnet Detection,” Proc. USENIX Second Workshop Steps to Reducing Unwanted Traffic on the Internet (SRUTI ’06), June 2006.

[7] Sinit P2P Trojan Analysis, http://www.lurhq.com/sinit.html, 2008.

[8] Phatbot Trojan Analysis, http://www.lurhq.com/phatbot.html, 2008.

[9] F. Monrose, “Longitudinal Analysis of Botnet Dynamics,”ARO/DARPA/DHS Special Workshop Botnet, 2006.

[10] Washington Post: The Botnet Trackers, http://www.washingtonpos.com/wp-dyn/ content/article/2006/02/16AR2006021601388.html, Feb. 2006.

[11] M. Rajab, J. Zarfoss, F. Monrose, and A. Terzis, “A Multifaceted Approach to Understanding the Botnet Phenomenon,” Proc. ACM SIGCOMM Internet Measurement Conf. (IMC ’06), Oct. 2006.

[12] A. Karasaridis, B. Rexroad, D. Hoeflin, “Widescale botnet detection and characterization,” Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets, 2007.

[13] A Taste of HTTP Botnets , team-cymru Inc, 2008, Available : http://www.team-cymru.org/ReadingRoom/Whitepapers/2008/http-botnets.pdf.

[14] Vogt R, Aycock J, Jacobson MJ. Army of botnets. In: Proc. of the 14th Annual Network & Distributed System Security Conf(NDSS). 2007.

[15] Zesheng Chen, Chao Chen, Qian Wang, "Delay-Tolerant Botnets," icccn, pp.1-6, 2009 Proceedings of 18th International Conference on Computer Communications and Networks, 2009.

[16] XF. Li, HX. Duan,W.Liu JP.Wu, "Understanding the Construction Mechanism of Botnets," uic-atc, pp.508-512, Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing, 2009.

[17] Chiang K, Lloyd L. A case study of the rustock rootkit and spam bot. In: Proc. of the 1st Workshop on Hot Topics in Understanding Botnets (HotBots 2007). 2007.

[18] R. Hund, M. Hamann, and T. Holz, "Towards Next-Generation Botnets," in Computer network Defense, 2008. EC22D 2008. European Conference on, 2008, pp. 13-40.

[19] C. Davis, S. Neville, J. Fernandez, J.-M. Robert, and J. McHugh, "Structured peer-to-peer overlay networks: Ideal botnets command and control infrastructures," In Proceedings of the 13th European Symposium on Research in Computer Security (ESORICS’08), October 2008.

Evaluation of Logistic Regression and Neural Network Model With Sensitivity Analysis on Medical Datasets

Raghavendra B.K. *raghavendra_bk@rediffmail.com*
Department of Computer Science & Engineering
Dr. M.G.R. Educational & Research Institute
Chennai, 600 095, India

S.K. Srivatsa *profsks@rediffmail.com*
Senior Professor
St. Joseph's College of Engineering
Chennai, 600 119, India

Abstract

Logistic Regression (LR) is a well known classification method in the field of statistical learning. It allows probabilistic classification and shows promising results on several benchmark problems. Logistic regression enables us to investigate the relationship between a categorical outcome and a set of explanatory variables. Artificial Neural Networks (ANNs) are popularly used as universal non-linear inference models and have gained extensive popularity in recent years. Research activities are considerable and literature is growing. The goal of this research work is to compare the performance of logistic regression and neural network models on publicly available medical datasets. The evaluation process of the model is as follows. The logistic regression and neural network methods with sensitivity analysis have been evaluated for the effectiveness of the classification. The classification accuracy is used to measure the performance of both the models. From the experimental results it is confirmed that the neural network model with sensitivity analysis model gives more efficient result.

Keywords: Artificial Neural Network, Classification Accuracy, Logistic Regression, Medical Dataset, Sensitivity Analysis.

1. INTRODUCTION

In the last few years, digital revolution has provided relatively inexpensive and available means to collect and store large amounts of patient data in databases, i.e., containing rich medical information and made available through the Internet for Health Services globally. Data mining techniques like logistic regression is applied on these databases to identify the patterns that are helpful in predicting or diagnosing the diseases and to take therapeutic measure of those diseases.

Nowadays statistical methods constitute a very powerful tool for supporting medical decisions. The size of medical data that any analysis or test of patients makes that doctors can be helped by statistical models to interpret correctly and to support their decisions. The models are a very powerful tool for doctors and these cannot substitute their viewpoint. On the other hand, the characteristics of medical data and the huge number of variables to be considered as fundamental point for the development of new technique as neural network for the analysis of the data [1].

Neural networks are considered as a field of artificial intelligence. The development of the models was inspired by the neural architecture of human brain. ANN have been applied in many disciplines, including biology, psychology, statistics, mathematics, medical science, and computer science. It has also been applied to a variety of business areas such as accounting and auditing, finance, management and decision making, marketing and production. Recently, artificial neural networks (ANNs) become a very popular model and have been applied to diagnose disease and predict the survival ratio of the patients. However, for medical analysis, ANNs have been shown to have some disadvantages as well as advantages. The most important advantages of ANNs are their discrimination power, detection of complex and nonlinear relationship between independent and dependent variables, and prediction of the case. The ANN model is developed empirically, they can be over-fitted for training data, and their usage is very difficult because of computational requirements. The performance of an ANN depends on the number of parameters, the network weights, the selection of an appropriate training algorithm, the type of transfer functions used, and the determination of the network size. Another disadvantage of using ANNs is that they require the initialization and adjustment of many individual parameters to optimize the classification performance. Many researchers have compared ANN versus LR. Some of them found that ANN and LR have similar classification performance. Compared to LR, neural network models are more flexible [2].

The rest of the paper is organized as follows: Section 2 reviews the prior literature, Logistic Regression technique is discussed in Section 3. Design of neural network is discussed in Section 4. Experimental validation using publicly available medical dataset is given in Section 5. Section 6 includes Experimental results and discussions followed by conclusion.

2. LITERATURE SURVEY

Logistic regression is used in power distribution fault diagnosis, while neural network, has been extensively used in power system reliability researches. Evaluation criteria of the goodness of the classifier includes: correct classification rate, true positive rate, true negative rate, and geometric mean [3].

The features of logistic regression and ANN have been compared and an experiment has been conducted on graft outcomes prediction using a kidney transplant dataset. The results shown reveal that ANN coupled with bagging is an effective data mining method for predicting kidney graft outcomes. This also confirms that different techniques can potentially be integrated to obtain a better prediction. Overall, the results reveal that in most cases, the ANN technique outperforms logistic regression [4].

The author’s presents an evaluation tool for the diagnosis of breast cancer of patients using clinical, pathological, and immunohistochemical data. The main aim was to compare the LR and NN models performances in classification. The neural network approach highlights different inputs from classical statistical model selection [5].

The research work was focuses on a machine learning approach to the classification of LR. The author’s applies a logistic regression based algorithm to three types of classification tasks: binary classification, multiple classifications and classification into a hierarchy. The results confirm that the logistic regression coupled with cross validation is an effective machine learning algorithm. It is not only a robust classification algorithm but also a very effective dimensionality reduction method. The author compared classification performance of logistic regression with several neural network algorithms: backpropagation, fuzzy artificial resonance ART, general regression, radial basis function, self-organizing map-kohonen. The best neural network: the fuzzy artificial resonance network, trained on 12 variables, and achieved 82.6% of correct predictions as compared to 90% for the logistic regression [6].

The research work from the authors was aims to identify the most and least significant factors for breast cancer survival analysis by means of feature evaluation indices derived from multilayer feedforward backpropagation neural networks (MLJFBPNN), fuzzy k-nearest neighbor classifier, and a logistic regression based backward stepwise method (LR). The results appear to suggest that SPF and NPIh appear to be the most and least important prognostic factors, respectively, for survival analysis in breast cancer patients, and should be investigated accordingly in future clinical studies in oncology. The results shown from each method identify a different set of factors as being the most important. It should therefore be suggested that it could be inappropriate to rely on one method's outcome for assessment of the factors, and thus it may be necessary to look at more than one method's outcome for a reliable prognostic factor assessment [7].

In another research work the author compares the performance of LR, NN, and CART decision tree methodologies and to identify important features for the small business credit scoring model on a Croatian bank dataset. The models obtained by all three methodologies were estimated and validated on the same hold-out sample, and their performance is compared. The results shows that the best NN model is better associated with data than LR and CART models [8].

The classification system was developed by the author and it was based on MLFFNN and LR to assess the risk of a family having HNPCC, purely on the basis of pedigree data. The proposed system can eliminate human errors associated with human fatigue and habits. Overall, MLFFNN outperformed to the LR in terms of the number of cases correctly classified and in terms of sensitivity, specificity and accuracy. Two out of 313 cases were misclassified by MLFFNN as opposed to 20 out of 313 by LR [9].

Artificial neural networks can be constructively used to improve the quality of linear models in medical statistics. ANNs are popularly used as universal non-linear inference models and they suffer from two major drawbacks. Their operation is not transparent because of the distributed nature of the representations they form, and this makes it different to interpret what they do. There is no clearly accepted model of generality, which makes it difficult to demonstrate reliability when applied to future data. In this paper neural networks generate hypotheses concerning interaction terms which are integrated into standard statistical models that are linear in the parameters, where the significance of the non-linear terms, and the generality of the model, can be assured using well established statistical tests [10].

The use of Artificial Neural Networks (ANN) is to construct distributions to carry out plausible reasoning in the field of medicine. It describes a comparison between Multivariate Logistic Regression (MLR) and the Entropy Maximization Network (EMN) in terms of explicit assessment of their predictive capabilities. The EMN and MLR have been used to determine the probability of harboring lymph node metastases at the time of initial surgery by assessment of tumor based parameters. Both predictors were trained on a set of 84 early breast cancer patient records and evaluated on a separate set of 92 patient records. Differences in performance were evaluated by comparing the areas under the receiver operating characteristic curve. The EMN model performed more accurately than the MLR model with *AZ* = 0.839, compared to the MLR model with AZ, = 0.809. The difference was statistically significant with two-tailed *P* value of less than 0.001. Accurate estimation of the prognosis would provide better stratification of patients for further treatment or investigation [11].

3. LOGISTIC REGRESSION

Regression is the analysis, or measure, of the association between a dependent variable and one or more independent variables. This association is usually formulated as an equation in which the independent variables have parametric coefficients that enable future values of the dependent variable to be predicted. Two of the main types of regression are: linear regression and logistic regression. In linear regression the dependent variable is continuous and in logistic it is either discrete or categorical. For logistic regression to be used, the discrete variable must be transformed into a continuous value that is a function of the probability of the event occurring. Regression is used for three main purposes: (1) description, (2) control and (3) prediction [12].

Logistic regression is also called as logistic model or logit model, is a type of predictive model which can be used, when the target variable is a categorical variable with two categories - for example active or inactive, healthy or unhealthy, win or loss, purchase product or doesn't purchase product etc. Logistic regression is used for the prediction of the probability of occurrence of an event by fitting the data into a logistic curve. Like many forms of regression analysis, it makes use of predictor variables; variables may be either numerical or categorical. For example, the probability that a person has a heart attack in a specified time that might be predicted from the knowledge of person's age, sex and body mass index. Logistic regression is used extensively in the medical and social sciences as well as in marketing applications such as prediction of customer's propensity to purchase a product or cease a subscription.

The response, Y, of a subject can take one of two possible values, denoted by 1 and 0 (for example, Y=1 if a disease is present; otherwise Y=0). Let X=(x_1, x_2, . . ., x_n) be the vector of explanatory variables. The logistic regression model is used to explain the effects of the explanatory variables in the form of binary response.

$$\text{Logit}\{\Pr(Y=1|x)\} = \log \{ \Pr(Y=1|x) / (1- \Pr(Y=1|x)) \} = \beta_0+\beta_1x_1+\beta_2x_2+\beta_3x_3+\ldots+\beta_kx_k \qquad (1)$$

Where β0 is called the intercept" and β_1, β_2, β_3, and so on are called the "regression coefficients" of x_1, x_2, x_3 respectively. Each of the regression coefficients describes the size of the contribution of the risk factor. A positive regression coefficient means that the risk factor increases the probability of outcome, where as a negative regression coefficient means that the risk factor decreases the probability of outcome, a large regression coefficient means that the risk factor strongly influences the probability of that outcome, a non-zero regression coefficient means that the risk factor has little influence on the probability of outcome [13].
The logistic function is given by

$$P=1/(1+e^{-\text{logit}(p)}) \qquad (2)$$

A graph of the function is shown in Figure 1. The logistic function is useful because it can take an input any value from negative infinity to positive infinity, whereas the output is confined to values between 0 and 1.

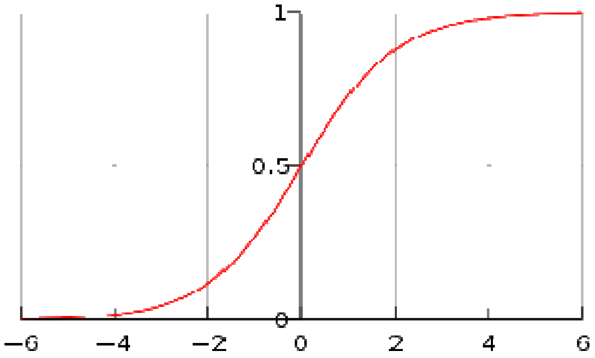


FIGURE 1: A graph of logistic regression function

4. DESIGN OF NEURAL NETWORK

A Neural network is a complex nonlinear modeling technique based on a model of a human neuron. A neural net is used to predict outputs (dependent variables) from a set of inputs (independent variables) by taking linear combinations of the inputs and then making nonlinear transformations of the linear combinations using activation function. It can be shown theoretically that such combinations and transformations can approximate virtually any type of response function. Thus, neural nets use large numbers of parameters to approximate any model. Neural nets are often applied to predict future outcome based on prior experience. For example, a neural net application could be used to predict who will respond to a direct mailing.

Neural networks are becoming very popular with data mining practitioners, particularly in medical research, finance and marketing. This is because they have proven their predictive power through comparison with other statistical techniques using real data sets. The example of a simple feed forward neural network with two layers is shown in Figure 2.

There are two main types of neural network models: supervised neural networks such as the multi-layer perceptron or radial basis functions, and unsupervised neural networks such as

Kohonen feature maps. A supervised neural network uses training and testing data to build a model. The data involves historical data sets containing input variables, or data fields, which correspond to an output. The training data is what the neural network uses to "learn" how to predict the known output, and the testing data is used for validation. The aim is for the neural networks to predict the output for any record given the input variables only [14].

One of the simplest feedforward neural networks (FFNN), such as the one in Figure 2, consists of two layers: an input layer, and output layer. In each layer there are one or more processing elements (PEs). PEs are meant to simulate the neurons in the brain and this is why they are often referred to as neurons or nodes. A PE receives inputs from either the outside world or the previous layer. There are connections between the PEs in each layer that have a weight (parameter) associated with them. This weight is adjusted during training. Information only travels in the forward direction through the network - there are no feedback loops.

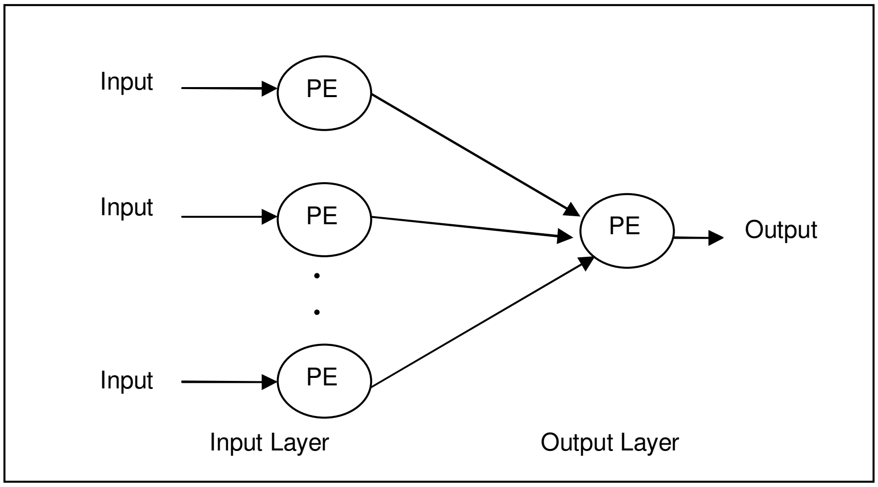


FIGURE 2: Example of a simple feed forward neural network with two layers

The simplified process for training a FFNN is as follows:
1. Input data is presented to the network and propagated through the network until it reaches the output layer. This forward process produces a predicted output.
2. The predicted output is subtracted from the actual output and an error value for the networks is calculated.
3. The neural network then uses supervised learning, which in most cases is backpropagation, to train the network. Backpropagation is a learning algorithm for adjusting the weights.
4. Once backpropagation has finished, the forward process starts again, and this cycle is continued until the error between predicted and actual outputs is minimized [13].

5. EXPERIMENTAL VALIDATION

The framework for neural network model with sensitivity analysis is shown in Figure 3. The process of evaluation is as follows. Sensitivity analysis has been done for the selected features from the dataset. The logistic function with steepness parameter (σ) is calculated using the following equation.

$$P=1/\ (1+e^{-\text{logit }(p) * \sigma}) \qquad (3)$$

where σ=2, 3

The response Y is then calculated as follows by using threshold (τ). Then the probability is calculated to develop a predictive model for classification using neural network. A tenfold cross validation has been used for evaluation on all publicly available medical dataset [15].

$$Y = \begin{cases} 1 & \text{if } P \geq \tau \\ 0 & \text{otherwise} \end{cases} \qquad (4)$$

where τ=0.2, 0,4, . . .

In 10-fold cross validation, the original sample is partitioned into 10 sub samples, of the 10 sub samples, a single sub sample is retained as the validation data for testing the model, and the remaining 9 sub samples are used as training data. The cross-validation process is then repeated 10 times (the folds), with each of the 10 sub samples used exactly once as the validation data. The 10 results from the folds then can be averaged (or otherwise combined) to produce a single estimation. The advantage of this method over repeated random sub sampling is that all observations are used for both training and validation and each observation is used for validation exactly once.

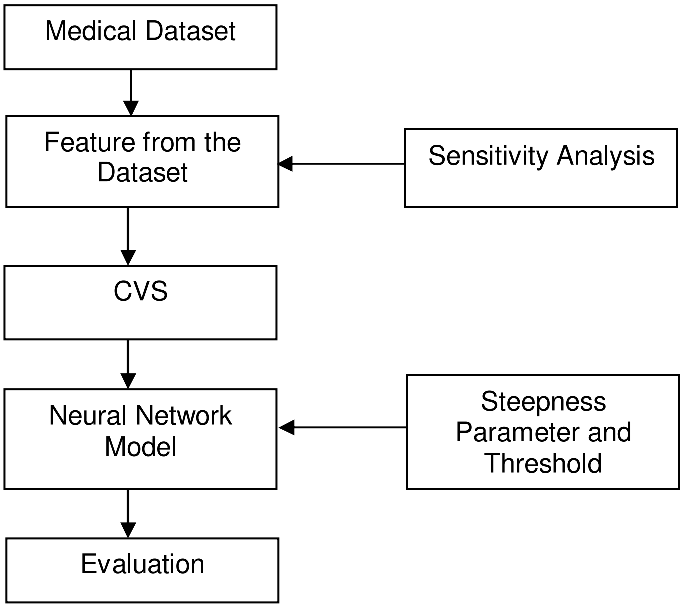


FIGURE 3: Neural network model with sensitivity analysis framework

6. RESULTS AND DISCUSSION

We have used publicly available medical datasets for our experiments whose technical specifications are as shown in Table 1. All the chosen datasets had at least one or more attributes that were continuous. The classification accuracy is used to measure the performance of logistic regression and neural network model on publicly available medical datasets. The results of the evaluation are given in Table 2. Figure 4 gives classification accuracy details after evaluation process. From the results it can be observed that the neural network model with sensitivity analysis gives more efficient result.

Sl. No	Medical Dataset	No of instances	Total no. of attributes	No of classes
1	Asthma	2464	5	2
2	Blood-transfusion	748	5	2
3	Flushot	159	4	2
4	Haberman	306	4	2
5	Liver-disorders	345	7	2
6	Spect test	187	23	2
7	Echocardiagram	132	11	2

TABLE 1: Specifications for the medical datasets

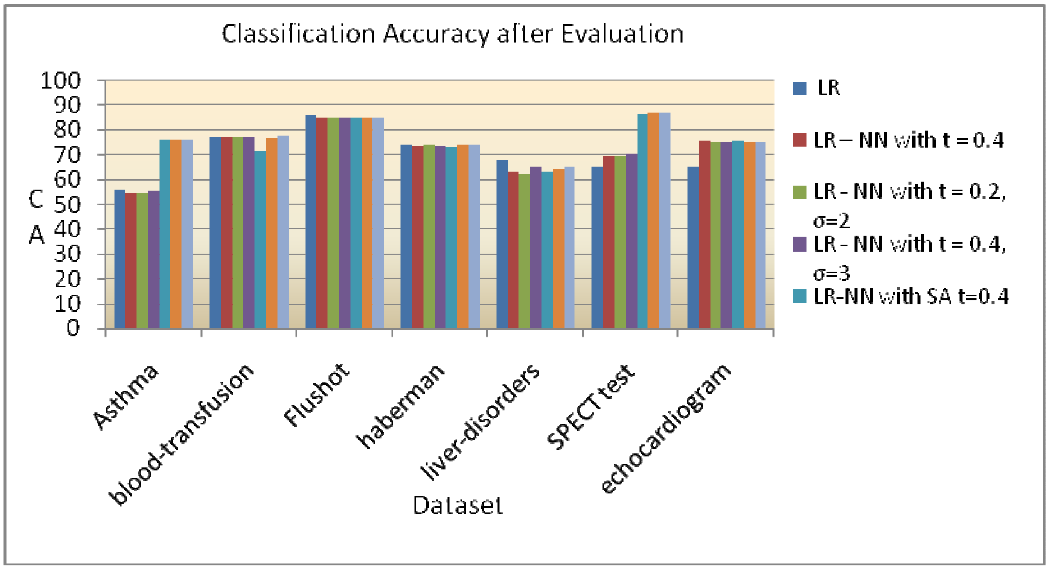


FIGURE 4: Classification accuracy after evaluation

Sl. No.	Name of the Dataset	LR	LR NN with t = 0.4	LR NN with t=0.2, σ=2	LR NN with t=0.4, σ=3	LR NN with SA, t=0.4	LR NN with SA, t=0.4, σ=2	LR NN with SA, t=0.4, σ=3
1	Asthma	56.16	54.58	54.62	55.19	76.29	76.29	76.33
2	blood-transfusion	77.13	77.4	77.27	77.27	71.65	76.6	77.94
3	Flushot	86.16	85	85	85	85	85	85
4	haberman	74.18	73.52	73.85	73.52	73.2	74.18	73.85
5	liver-disorders	68.11	63.18	62.31	65.21	63.18	64.34	65.21
6	SPECT test	65.24	69.51	69.51	70.58	86.63	87.16	87.16
7	echocardiogram	65.15	75.75	75	75	75.75	75	75

TABLE 2: Logistic regression and neural network specification with sensitivity analysis for the medical datasets

7. CONCLUSION

Finally, the conclusions of this work are based on the publicly available medical data sets. The results of this study are more promising. In this research work an attempt was made to evaluate logistic regression and neural network model with sensitivity analysis on publicly available medical data sets. The classification accuracy is used to measure the performance of both the models. From the experimental results it is confirmed that neural network model with sensitivity analysis gives more efficient result.

8. REFERENCES

[1]. Luis Mariano Esteban Escaño, Gerardo Sanz Saiz, Francisco Javier López Lorente, Ángel Borque Fernando and José Moría Vergara Ugarriza, “Logistic Regression Versus Neural Networks for Medical Data”, Monografías del Seminario Matemático García de Galdeano 33, 245-252, 2006.

[2]. Bahar Tasdelen, Sema Helvaci, Hakan Kaleagasi, Aynur Ozge, “Artificial Neural Network Analysis for Prediction of Headache Prognosis in Elderly Patients”, Turk J Med Sci 2009; 39(1); 5-12.

[3]. LeXu, Mo-Yuen Chow, and Xiao-Zhi Gao, “Comparisons of Logistic Regression and Artificial Neural Network on Power Distribution Systems Fault Cause Identification”, Proceedings of 2005 IEEE Mid-Summer Workshop on Soft Computing in Industrial Applications (SMCia/05), Helsinki, Finland, June 28-30, 2005.

[4]. Fariba Shadabi and Dharmendra Sharma, “Comparison of Artificial Neural Networks with Logistic Regression in Prediction of Kidney Transplant Outcomes”, Proceedings of the 2009 International Conference of Future Computer and Communication (ICFCC), 543-547, 2009.

[5]. V.S. Bourdes, S. Bonnevay, P.J.G. Lisbosa, M.S.H. Aung, S. Chabaud, T. Bachelot, D. Perol and S. Negrier, “Breast Cancer Predictions by Neural Networks Analysis: a Comparison with Logistic Regression”, Proceedings of the 29th International Conference of the IEEE Engineering in Medicine and Biology Society, Lyon, France, August 23-26, 2007, 5424-7.

[6]. Jack R. Brzezinski George J. Knaft, “Logistic Regression Modeling for Context-Based Classification”, DEXA Database and Expert Systems Applications Workshop, 1999.

[7]. Seker H., Odetayo M., Petrovic D., Naguib R.N.G., Bartoli C., Alasio L., Lakshmi M.S., Sherbet G.V. (2002), “An Artificial Neural Network Based Feature Evaluation Index for the Assessment of Clinical Factors in Breast Cancer Survival Analysis”, Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering.

[8]. Marijana Zekic-Susac, Natasa Sarlija, Mirta Bensic, “Small Business Credit Scoring: A Comparison of Logistic Regression, Neural Network, and Decision Tree Models”, 26th International Conference on Information Technology Interfaces (ITI 2004), Cavtat, Croatia, 265-270.

[9]. M Münevver Köküuer, Raouf N. G. Naguib, Peter Jančovič, H. Banfield Younghusband and Roger Green, “A Comparison of Multi-Layer Neural Network and Logistic Regression in Hereditary Non-Polyposis Colorectal Cancer Risk Assessment”, Proceedings of the 2005 IEEE Engineering in Medicine and Biology , 27th Annual Conference, Shanghai, China, September 2005, 2417-2420.

[10]. Lisbosa P.J.G., and H. Wong (2001), “Are neural networks best used to help logistic regression? An example from breast cancer survival analysis”, IEEE Transactions on Neural Networks, 2472-2477.

[11]. Poh Lian Choong, and Christopher J.S. DeSilva (1996), “A Comparison of Maximum Entropy Estimation and Multivariate Logistic Regression in the Prediction of Axillary Lymph Node Metastasis in Early Breast Cancer Patients”, The 1996 IEEE International Conference on Neural Networks, 1468-1473.

[12]. Neter J., Kutner M.H., Nachtsheim C.J., Wasserman W., Applied Linear Regression Models, 3rd Ed. 1996, Irwin, USA (ISBN 0-256-08601-X).

[13]. http://en.wikipedia.org/wiki/Logistic_regression

[14]. Portia A. Cerny, 2001, Datamining and Neural Networks from a Commercial Perspective, Auckland, New Zealand Student of the Department of Mathematical Sciences, University of Technology, Sydney, Australia.

[15]. C.L. Blake, C.J. Merz, "UCI repository of machine learning databases". [http://www.ics.uci.edu/~mlearn/ MLRepository.html], Department of Information and Computer Science, University of California, Irvine.

Dynamic Audio-Visual Client Recognition Modeling

Tijjani Adam *tijjaniadam@yahoo.com*
Institute of Nano Electronic Engineering
Universiti Malaysia Perlis
Kangar, 01000, Malaysia

U. Hashim *uda@unimap.edu.my*
Institute of Nano Electronic Engineering
Universiti Malaysia Perlis
Kangar, 01000, Malaysia

Abstract

This paper contains a report on an Visual Client Recognition System using Matlab software which identifies five clients and can be improved to identify as many clients as possible depending on the number of clients it is trained to identify which was successfully implemented. The implementation was accomplished first by visual recognition system implemented using The Principal Component Analysis, Linear Discriminant Analysis and Nearest Neighbor Classifier. A successful implementation of second part was achieved by audio recognition using Mel-Frequency Cepstrum Coefficient, Linear Discriminant Analysis and Nearest Neighbor Classifier the system was tested using images and sounds that have not been trained to the system to see whether it can detect an intruder which lead us to a very successful result with précised response to intruder and also explored another means implementing the visual recognition section using a Neural Network The work on visual recognition system was converted into a simulink block set which was then implemented in a Signal wave.

Keywords: Audio- visual Client Recognition, Discriminate, Multi-model Biometric System, Simulink, Neural Network.

1. INTRODUCTION

Audio-visual client recognition system is one of the multi-modal biometric systems. The system automatically recognizes or identifies the user based on voice and facial information. The aim of this study is to develop an audio-visual recognition system. The system principal component analysis (PCA) involves a mathematical procedure that transforms a number of possibly correlated variables into a smaller number of uncorrelated variables called principal components[1][2]. The first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible. It is used in signal processing for data compression. It can be shown that the best linear compression of a dataset can be achieved by projecting the data onto the Eigen vectors of the data's covariance matrix; the compressed values are thus the principal components of the data. The system is able to identify 5 clients and also detect give an intruder alert. The principal component analysis (PCA) involves a mathematical procedure that transforms a number of possibly correlated variables into a smaller number of uncorrelated variables called principal components [3]. The first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible[4]. It is used in signal processing for data compression. It can be shown that the best linear compression of a dataset can be achieved by projecting the data onto the Eigen vectors of the data's covariance matrix; the compressed values are thus the principal components of the data.

After using PCA, unwanted variations caused by the illumination, facial position and facial expression still retain. Accordingly, the features produced by PCA are not necessarily good for Discriminant among classes. Therefore, the most discriminating face features are further acquired by using the LDA method [3][4]. The purpose of LDA is to group images of the same class and separate images of different classes. The nearest neighbour algorithm calculates the distance from the unknown testing sample to every training sample by using the distance metric [5]. Audio Authentication System(AAS): An audio authentication system is a system that identifies a speaker by analyzing spectral shape of the voice signal, usually done by extracting instructions while preparing/modifying these guidelines. This guideline is used for all journals. This guideline is used for all journals. These are the manuscript preparation guidelines used as a standard template for all journal submissions. Author must follow these instructions while preparing/modifying these guidelines. This guideline is used for all journals. This guideline is used for all journals. These are the manuscript preparation guidelines used as a standard template for all journal submissions. Author must follow these instructions while preparing/modifying these guidelines. This guideline is used for all journals and matching the feature of voice signal. Ceptra are most commonly used features used in speech authentication tasks. A Cepstrum of a given signal is obtained using homomorphic filtering which converts a convolved source and filter impulse responses to linear summations. An approach to this is computing the speech linear prediction coefficients (LPCC). An alternative way is to apply a Mel-scale filter-bank function to the speech spectrum. The resulting coefficients are referred to as Mel-Frequency Cepstrum Coefficients (MFCC). There are other types of Ceptra that can be obtained through variations of, or additional processing in, the above methods. Examples of these are perceptual linear prediction coefficients (PLP) and linear filter bank cepstral coefficients (LFCC). LPCC and MFCC are the most widely used speech features. This project focuses only on MFCC because it is the best known and most popular.

The Mel-Frequency Cepstrum Coefficients are coefficients that collectively make up a Mel-Frequency Cepstrum. Mel-Frequency Cepstrum is described as the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a non linear mel scale frequency[6][7]. The mel scale is a perpetual scale of pitches judged by listeners to be equal in distance from one another. The reference point between this scale and normal frequency measurement is defined by equating a 1000Hz tone, 40dB above the listener.s threshold, with a pitch 0f 1000 mels. Above about 500Hz, larger and larger intervals are judged by listeners to produce equal pitch increments. As a result, four octaves on the hertz scale above 500Hz are judged by listeners to produce equal pitch increments. As a result, four octaves on the hertz scale above 500Hz are judged to comprise about two octaves on the Mel scale. The name Mel comes from the word melody to indicate that the scale is based on pitch comparisons. Its being used is because the human perception of the frequency contents of sound does not follow a linear scale. A neural network (NN) is a network or circuit of biological neurons. A neural network may either be a biological neural network or an artificial neural network[8][9]. Biological neural networks are made up of real biological neurons that are connected or functionally related in the peripheral nervous system or the central nervous system. In the field of neuroscience[10], they are often identified as groups of neurons that perform a specific physiological function in laboratory analysis

1.1 Audio Authentication System

An audio authentication system is a system that identifies a speaker by analyzing spectral shape of the voice signal, usually done by extracting and matching the feature of voice signal[10]. Ceptra are most commonly used features used in speech authentication tasks. A Cepstrum of a given signal is obtained using homomorphic filtering which converts a convolved source and filter impulse responses to linear summations. An approach to this is computing the speech linear prediction coefficients (LPCC). An alternative way is to apply a Mel-scale filter-bank function to the speech spectrum[11]. The resulting coefficients are referred to as Mel-Frequency Cepstrum Coefficients (MFCC). There are other types of Ceptra that can be obtained through variations of, or additional processing in, the above methods. Examples of these are perceptual linear prediction coefficients (PLP) and linear filter bank cepstral coefficients (LFCC). LPCC and MFCC are the most widely used speech features. This project focuses only on MFCC because it is the best known and most popular. The Mel-Frequency Cepstrum Coefficients are coefficients that collectively make up a Mel-Frequency Cepstrum. Mel-Frequency Cepstrum is described as the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a non linear mel scale frequency[12]. The mel scale is a perpetual scale of pitches judged by listeners to be equal in distance from one another. The reference point between this scale and normal frequency measurement is defined by equating a 1000Hz tone, 40dB above the listener□s threshold, with a pitch 0f 1000 mels. Above about 500Hz, larger and larger intervals are judged by listeners to produce equal pitch increments[10]. As a result, four octaves on the hertz scale above 500Hz are judged by listeners to produce equal pitch increments. As a result, four octaves on the hertz scale above 500Hz are judged to comprise about two octaves on the mel scale. The name mel comes from the word melody to indicate that the scale is based on pitch comparisons. Its being used is because the human perception of the frequency contents of sound does not follow a linear scale.The formula to convert frequency into mel is m=2595log10f700+1=1127logef700+1, Its inverse, f is given by: f=70010m2595-1=700em1127-1.

The need to recognise individuals is vital to human life. The most natural way to do this is by recognising people's faces or voices. However, it is impossible to personally know everyone that an individual may have to interact with. Biometric devices and technologies automate the process of recognising individuals; Audio-Visual client recognition system is one of the multi-modal biometric systems. The system automatically recognizes or identifies the user based on facial information ranging from few to large number of clients but . The aim of this study is to develop adaptable visual recognition algorithms for client recognition[15,16], the algorithm can be employed for security systems and can be compared to other biometrics such as fingerprint or eye iris recognition systems as well as access control using facial information recognition depending on condition and application is trained for[17,18,19]. The system principal component analysis (PCA) involves a mathematical procedure that transforms a number of possibly correlated variables into a smaller number of uncorrelated variables called principal components[19]. The first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible. It is used in signal processing for data compression. It can be shown that the best linear compression of a dataset can be achieved by projecting the data onto the Eigen vectors of the data's covariance matrix; the compressed values are thus the principal components of the data. The system is able to identify 5 clients and also detect give an intruder alert. The principal component analysis (PCA) involves a mathematical procedure that transforms a number of possibly correlated variables into a smaller number of uncorrelated variables called principal components [19]. The first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible[18]. It is used in signal processing for data compression. It can be shown that the best linear compression of a dataset can be achieved by projecting the data onto the Eigen vectors of the data's covariance matrix; the compressed values are thus the principal components of the data. After using PCA, unwanted variations caused by the illumination, facial position and facial expression still retain. Accordingly, the features produced by PCA are not necessarily good for Discriminant among classes. Therefore, the most discriminating face features are further acquired by using the LDA method [8,9]. The purpose of LDA is to group images of the same class and separate images of different classes. The nearest neighbour algorithm calculates the distance from the unknown testing sample to every training sample by using the distance metric [10]

1.2 Visual Recognition System Using a Neural Network

A neural network is a network or circuit of biological neurons. A neural network may either be a biological neural network or an artificial neural network. Biological neural networks are made up of real biological neurons that are connected or functionally related in the peripheral nervous system or the central nervous system. In the field of neuroscience, they are often identified as groups of neurons that perform a specific physiological function in laboratory analysis.

Artificial neural networks are made up of interconnecting artificial neurons (programming constructs that mimic the properties of biological neurons).They use mathematical or computational model for information processing based on a connectionistic approach to computations. It is an adaptive system that changes its structure based on external and internal information that flows through the network. They may hence be defined as non-linear statistical modeling or decision making tools. They can be used to model complex relationships between inputs and outputs or to find patterns in data. Artificial neural networks may either be used to gain an understanding of biological neural networks, or for solving artificial intelligence problems without necessarily creating a model of a real biological system Neural networks[13], as used in artificial intelligence have been viewed as simplified models of neural processing in the brain, even though the relation between this model and brain biological architecture is debated. It is a complex statistical processor. Neural coding is concerned with how sensory and other information is represented in the brain by neurons. The main goal of studying neural coding is to characterize the relationship between stimulus and individual or ensemble neural responses and the relationship among electrical activity of the neurons in the ensemble[14]. It is thought that neurons can encode both digital and analog information. As neural networks emulate human thinking, they are used in visual recognition systems.

2: METHODOLOGY

2.1 The process of Principal Component Analysis

2.1.1 Image data acquisition

Each acquired 2-D facial image was represented by a 1-D vector obtained by concatenating each column/row into a long thin vector in which each vector was represented as

Xi=x1i….xNi 2.1.0

2.1.2 Data Cantering

Each of the training images was mean cantered by subtracting the mean image from each of the training images. The mean image is a column vector in which each entry is the mean of all corresponding pixels of the training images in figure1.
Xi=xi-m 2.1.1

m=1pi=1pxi 2.1.2

2.1.3 Data matrix creation

The vectors were combined side by side and a data matrix of size N x P was formed, were P is the number of training images and each vector is a single image vector.

X=x1x2…xp 2.1.3

Covariance was calculated by multiplying the data matrix with its transpose.

Ù=XXT 2.1.4

Eigen values and eigenvectors of the covariance were calculated.

ΩV=▯V 2.1.5
The eigenvectors were sorted from high to low according to their corresponding eigenvalues. The eigen vector corresponding to the eigenvector with the largest value is the eigenvector that finds

the greatest variance in the training images. Else, the eigenvector that associated with the smallest Eigen value finds the least variance in the training images.
An eigenspace V was formed by this eigenvectors matrix.
V=V1V2…Vp 2.1.6

2.1.4 Projection of training images

Each of the centred training images was projected into the eigenspace. In this process, the dot product of the centred training image with each of the ordered eigenvectors was calculated.

xi= VTxi 2.1.7

The dot product of the image and the first eigenvector became the first value in the new vector. Steps 1 was carried out in a file “imagepreprocessing.m”, (2.1.1 to 2.1.6) in “pca.m” and (2.1.7) in “TestPCA.m”

2.2 The processes of Linear Discriminant Analysis (LDA)

2.2.1 Within Class scatter matrix calculation

The within class scatter matrix was calculated. It measured the amount of scatter between training images in the same class. It was calculated as the sum of the covariance matrices of the centred images in the ith class.

Si=x.Xi(x-mi)(x-mi)T 2.2.1

SW=i=1CSi 2.2.2

Where mi: is the mean of training images within the class. The within class scatter matrix (SW) is the sum of all the scatter matrices.

The class scattered matrix (SB) was calculated as the sum of the covariance matrices of the difference between the total mean and the mean value in each class.

SB=i=1Cni(mi-m)(mi-m)T 2.2.3

The generalized eigenvectors (V) and eigenvalues (L) of the within class and between class scatter matrices were computed.

SB=.SWV 2.2.4

The non-zero eigenvectors were sorted from high to low according to their corresponding eigenvalues and the first C-1 eigenvectors were kept which formed the Fisher basis vector. The training images were projected onto the fisher basis vector by calculating the dot product of the training image with each of the Fisher basis vector. The above steps were carried out in a file *“lda.m”.*

2.3 Audio Authentication System Process

The system was designed in such a way to be trained with sound clips each of which represents the identity of an individual. After it has been trained, the system receives an input signal, compare against all possible combinations of sounds it was trained with and then authenticates whether the inputted sound matches any of the sound it was trained with or not. If there is a match it welcomes a particular individual. In the absence of any match it produces an intruder alert. The training was carried out with five different sound clips representing the identity of a single individual so that it will be familiar with each person□s voice and it was done with following steps; **Data Extraction:** data stored in audio wave format is converted to a form suitable for further computer processing and analysis, **Data Pre-processing:** removing pauses, silences, weak unvoiced sound signals and valid data detection. A digital was defined to perform this task and the related programmed codes were saved as “vad.m”, **Mel-Frequency Cepstrum:** convert the speech waveform to some type of parametric representation (at a considerably lower information rate) for further analysis and processing. This is referred to as signal processing front end, **Frame Blocking:** The continuous speech signal is blocked into frames of N samples, with adjacent frames being separated by M (M<N). The first frames consist of N samples. The second frame begins on samples after the first frame, and overlaps it by N-M samples. This process continues until all the speech is accounted for one or more frames, **Windowing:** spectrum distortion minimization by using a window to taper the signal to zero at the beginning and end of each frame. If the window is w(n), 0 ≤ n ≤ N–1, result; y1n=x1nwn, 0≤n≤N-1, **Fast Fourier Transform (FFT):** Xn=K=0N-1xke-2πjknN, n = 0, 1, 2, ….., N-1, **Mel-Frequency Wrapping:** The spacing and bandwidth of such a filter is determined by a constant Mel-frequency interval. The modified spectrum of S(w) consists an output power of these filters when S(w) is the input. The number of Mel-spectrum coefficients, K was chosen to be 13, **Cepstrum:** The cepstral representation of the speech spectrum provides a good representation of the local spectral properties of the signal for the given frame analysis. Because the Mel-spectrum coefficients (and their logarithm) are real numbers, they are being converted to the time domain using Discrete Cosine Transform (DCT). In this step the Mel-Frequency Cepstrum Coefficients are finally found. The set of coefficients is called an acoustic vector. Hence each input utterance is transformed into a sequence of acoustic vectors, **Feature Matching:** Previous Nearest Neighbour algorithm was used as in the case of the Visual recognition system.

2.4 : Visual Recognition System (Neural Network) Process

The input to neural network visual recognition system is a digitized image in matrix form, **Learning process**; The neural networks function is to distinguish between different inputs fed to it and identify each one correctly. It does this by first learning the various images and their classes. In this project the main area of concern is the human visual recognition system. An individual has different moods. Images of an individual in different moods will be recognized by the system differently. This is the reasons why classes are needed and a class is just a collection of images of an individual in different moods. Each image in a class can represent the individual□s identity. Images are taught to the network. The input to the network is an input matrix, P. Any neural network may learn in a supervised or unsupervised manner by adjusting its weight. Each individual image taught to the network possesses a corresponding weight matrix. For the nth image taught to the network, the weight matrix is denoted by Wn. As the learning process continues, the weight matrix is being upgraded. Each image inputted into the neural network is stored under a class. The weight matrix is updated as shown below:

```
for all i = 1 to x
{

for all j= 1 to y

{ Wn(i, j) = Wn(i, j) + P(i ,j)

    }
}
```

x and y are the dimensions of matrix WK and P.**Candidate Score (□):** This is the product of corresponding elements of the weight matrix Wn of the nth learnt image and an input image as its candidate. It is obtained using the equation

□=i=1kj=1yWki,j*P(i,j).

Ideal Weight Model Score (μ): This gives the sum total of all the positive elements of the weight matrix of a learnt image. It can be formulated as follows;

```
For i =1 to x
{
For j =1 to y
```

```
{
If WK (i, j) > 0 then
 {
$\mu(K) = \mu(K + WK(i, j)$
}
}
}
```

Similar to the visual recognition system using PCA, LDA and nearest neighbour algorithm in parts 1 to 3, training images 1 to 5 can be taught to the network as a single class representing Mr. A. **Recognition Quotient (Q):** This statistics gives a measure of how well the recognition system identifies an input image as a matching candidate for one of its many learnt images. It is Qk=□(K)μ(K) The greater the value of Q, the more accurate the system is in identifying individuals. A low value of Q indicates a poor recognition. In this case the individual does not exist within the knowledge base or the network has to be taught until a satisfactory Q is obtained.

2.5 Implementation Using Matlab Software

This involved the use of the Neural Network Toolbox. To define an image recognition problem, there must be a set of *Q* input vectors as columns in a matrix. Another set of *R* target vectors have to be arranged so that they indicate the classes to which the input vectors are assigned. The target vector must have the following properties

1. Row headers – column vector containing names of all row headers.
2. Col headers – row vector containing the names of all column headers.
3. Textdata - matrix containing all imported test data. Empty elements are set to „□.
4. Data – matrix containing all imported numeric data. Empty elements are set to NaN

With the neural network toolbox, previously in part1 the training images were uploaded into workspace in a file "*imagepreprocessing.m*"**.** As a necessity to arrange the training images as columns in a vector, it has been already done in the file and so the generated matrix was used as the input vector. A target vector was then generated in M-file by name "*target.m*" in which images were assigned the following target values

TABLE1: Target values of training images. Images that represent the same individual were assigned the same target values.

N0.	Image	Target Value
1	1-5	001
2	6-10	010
3	11-15	011
4	16-20	100
5	21-25	101

After the development of simulink block from command ”nprtool" to the finish using Matlab command window, the Visual Recognition system was converted into simulink block set as shown in figure2. This involved calculations of eigenvectors, which makes it tedious to implement using simulink. As such the results of eigenvector calculations will be imported into simulink and will show the identity of each individual by number

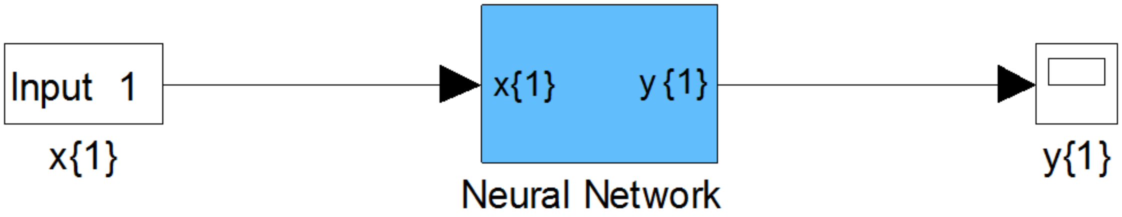
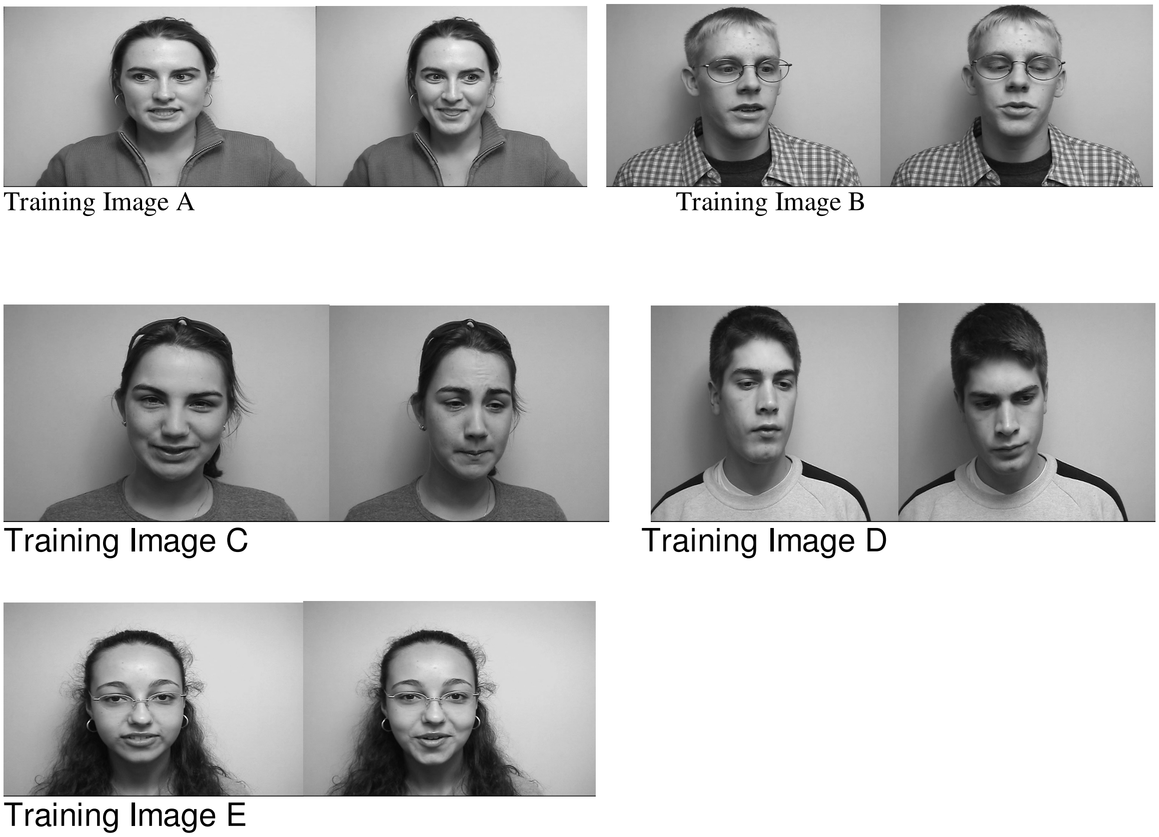


Figure2: Simulink Block set of the Neural Network, The block Neural Network is a subsystem with many components in it.



Training Image A

Training Image B

Training Image C

Training Image D

Training Image E

FIGURE1: Training images and sound used in this study

3: RESULTS AND DISCUSSION

The model feature is summarized as follow: the extracted facial signals and template representing each individual is constructed, there remains the task of classifying live samples. In the case of identity verification the live sample features must be compared with the template associated with the claimed identity. The distance between these two points in the feature space will have to be compared with a threshold of acceptability. This threshold is set empirically to produce acceptable performance. In the case of person identification the live sample will have to be compared with all the stored templates and a range of distances will be measured. The closest match with the smallest distance will then be chosen as the identity. Various architectures have been used for performing such classifications. There is usually a training phase where the classifier is given valid feature vectors and their associated identity tokens. The success of the operational phase depends on the quality of this training phase. The system can be arranged so that if the any of the modalities produce an acceptance then the user is accepted and the other layers need not be invoked. It is also possible to have a logical operation performed at the final stage to combine the decisions at the parallel layers.

In a layered approach several modalities of biometric information may be integrated The nearest neighbour algorithm calculates the distance from the unknown testing sample to every training sample by using the distance metric[9]. Euclidean distance was used in the distance computation. It is the last stage in visual recognition system. The mathematics formula of Euclidean distance is:

Dxi, yj=j=1n(xj-yj)2 (3-1)

The MATLAB codes showing the above performed task are contained in a file “PCALDAnn.m” The results below were obtained when the main program was run (PCALDAmainprog.m). They were obtained when the testing image directory was set as the testing database (imdir='E:\DSP Project Files\Testingdatabase10imgs\';).We shown a set of results also when the fraud database is set as the testing image directory in figure 1.

3.1 Data inputting

```
Input =
E:\DSP Project Files\Trainingdatabase25imgs\1.bmp
Input =
E:\DSP Project Files\Trainingdatabase25imgs\2.bmp
Input =
E:\DSP Project Files\Trainingdatabase25imgs\3.bmp
Input =
E:\DSP Project Files\Trainingdatabase25imgs\4.bmp
Input =
E:\DSP Project Files\Trainingdatabase25imgs\5.bmp
Input =
E:\DSP Project Files\Trainingdatabase25imgs\6.bmp
Input =
E:\DSP Project Files\Trainingdatabase25imgs\7.bmp
Input =
E:\DSP Project Files\Trainingdatabase25imgs\8.bmp
Input =
E:\DSP Project Files\Trainingdatabase25imgs\9.bmp
Input =
E:\DSP Project Files\Trainingdatabase25imgs\10.bmp
Input =
E:\DSP Project Files\Trainingdatabase25imgs\11.bmp
Input =
E:\DSP Project Files\Trainingdatabase25imgs\12.bmp
Input =
E:\DSP Project Files\Trainingdatabase25imgs\13.bmp
Input =
E:\DSP Project Files\Trainingdatabase25imgs\14.bmp
Input=
E:\DSP Project Files\Trainingdatabase25imgs\15.bmp
Input =
E:\DSP Project Files\Trainingdatabase25imgs\16.bmp
Input =
E:\DSP Project Files\Trainingdatabase25imgs\17.bmp
Input =
E:\DSP Project Files\Trainingdatabase25imgs\18.bmp
Input=
E:\DSP Project Files\Trainingdatabase25imgs\19.bmp
```

```
Input =
E:\DSP Project Files\Trainingdatabase25imgs\20.bmp
Input =
E:\DSP Project Files\Trainingdatabase25imgs\21.bmp
Input =
E:\DSP Project Files\Trainingdatabase25imgs\22.bmp
Input =
E:\DSP Project Files\Trainingdatabase25imgs\23.bmp
Input =
E:\DSP Project Files\Trainingdatabase25imgs\24.bmp
Input =
E:\DSP Project Files\Trainingdatabase25imgs\25.bmp
```

3.2 Test Input

```
Testinput =
E:\DSP Project Files\Testingdatabase10imgs\ppl1.bmp
Testinput =
E:\DSP Project Files\Testingdatabase10imgs\ppl2.bmp
Testinput =
E:\DSP Project Files\Testingdatabase10imgs\ppl3.bmp
Testinput =
E:\DSP Project Files\Testingdatabase10imgs\ppl4.bmp
Testinput =
E:\DSP Project Files\Testingdatabase10imgs\ppl5.bmp
Testinput =
E:\DSP Project Files\Testingdatabase10imgs\ppl6.bmp
Testinput =
E:\DSP Project Files\Testingdatabase10imgs\ppl7.bmp
Testinput =
E:\DSP Project Files\Testingdatabase10imgs\ppl8.bmp
Testinput =
E:\DSP Project Files\Testingdatabase10imgs\ppl9.bmp
Testinput =
E:\DSP Project Files\Testingdatabase10imgs\ppl10.bmp
```

3.3 Simulation Results

```
Ans=
Welcome! Ms. A
Ans =
Welcome! Ms. A
Ans =
Welcome! Ms. B
Ans =
Welcome! Ms. B
Ans =
Welcome! Ms. C
Ans =
Welcome! Ms. C
Ans =
Welcome! Ms. D
Ans =
Welcome! Ms. D
Ans =
Welcome! Ms. E
Ans =
Welcome! Ms. E
```

3.4 Tests for Extruder

RESULTS OBTAINED USING THE FRAUD DATABASE AS TESTING DATABASE

In this case, the testing database was set to be imdir='E:\Fraud\DSP Project Files\Fraud database\';. The results obtained are shown below:

3.5 Input Data

```
Input =
E:\DSP Project Files\Trainingdatabase25imgs\1.bmp
Input =
E:\DSP Project Files\Trainingdatabase25imgs\2.bmp
Input =
E:\DSP Project Files\Trainingdatabase25imgs\3.bmp
Input=
 E:\DSP Project Files\Trainingdatabase25imgs\4.bmp
Input =
E:\DSP Project Files\Trainingdatabase25imgs\5.bmp
Input =
E:\DSP Project Files\Trainingdatabase25imgs\6.bmp
Input =
E:\DSP Project Files\Trainingdatabase25imgs\7.bmp
input =
E:\DSP Project Files\Trainingdatabase25imgs\8.bmp
input =
E:\DSP Project Files\Trainingdatabase25imgs\9.bmp
input =
E:\DSP Project Files\Trainingdatabase25imgs\10.bmp
input =
E:\DSP Project Files\Trainingdatabase25imgs\11.bmp
input =
E:\DSP Project Files\Trainingdatabase25imgs\12.bmp
input =
E:\DSP Project Files\Trainingdatabase25imgs\13.bmp
input =
E:\DSP Project Files\Trainingdatabase25imgs\14.bmp
input =
E:\DSP Project Files\Trainingdatabase25imgs\15.bmp
Input =
E:\DSP Project Files\Trainingdatabase25imgs\16.bmp
Input =
E:\DSP Project Files\Trainingdatabase25imgs\17.bmp
Input =
E:\DSP Project Files\Trainingdatabase25imgs\18.bmp
Input =
E:\DSP Project Files\Trainingdatabase25imgs\19.bmp
Input =
E:\DSP Project Files\Trainingdatabase25imgs\20.bmp
Input=
E:\DSP Project Files\Trainingdatabase25imgs\21.bmp
Input =
E:\DSP Project Files\Trainingdatabase25imgs\22.bmp
Input =
E:\ DSP Project Files\Trainingdatabase25imgs\23.bmp
Input =
E:\DSP Project Files\Trainingdatabase25imgs\24.bmp
```

```
Input =
E:\DSP Project Files\Trainingdatabase25imgs\25.bmp
```

3.6 Testinput =

```
E:\Fraud\DSP Project Files\Fraud database\ppl1.bmp

Testinput =

E:\Fraud\DSP Project Files\Fraud database\ppl2.bmp

Testinput =

E:\Fraud\DSP Project Files\Fraud database\ppl3.bmp

Testinput =

E:\Fraud\ DSP Project Files\Fraud database\ppl4.bmp
Testinput =
E:\Fraud\DSP Project Files\Fraud database\ppl5.bmp
```

3.7 Simulation

```
Ans =
Get Away Intruder!!
Ans =
Get Away Intruder!!
Ans =
Get Away Intruder!!
Ans =
Get Away Intruder!!
Ans =
Get Away Intruder!!
```

3.8 Simulation results on Audio Authentication System

```
ans =
Welcome!Ms. A
ans =
Welcome!Mr. B
ans =
Welcome!Mr. B
ans =
Welcome!Ms. C
ans =
Get Away Intruder!!
ans  =
Welcome! Mr. D
ans =
Welcome!Mr. D
ans =
Welcome!Mr. E
ans =
Welcome!Mr. E
```

Simulation results showed that the system was able to identify the clients. It identified the second testing voice of Ms. C as an intruder. The source voices were listened to and it was observed that voice from the source file has the farthest difference from the training ones compared to the other test voice. This is an indicator that the system may require more training for the voice section which can be accomplished by increasing the number of training images representing a single identity to say ten. As for detecting a fraud, the system has given a full assurance that it will reject any fraud voice

3.9 Implementation Using Simulink

In the previous program, each time an individual has been identified, the identity matrix is updated. As such, the program was modified in such a way that the identity matrix will show the identity of each and every individual by column. The number of column is ten since we have ten test pictures. This was uploaded into simulink and the simulink was run after completing. the following results were obtained.

```
val(:,:,1) = 4
val(:,:,2) =4
val(:,:,3) =8
val(:,:,4) =6
val(:,:,5) =12
val(:,:,6) =15
val(:,:,7) = 17
val(:,:,8) =18
val(:,:,9) =21
val(:,:,10) =23
```

Results showed the identity numbers of the test images. In file “PCALDAnn.m”, the coding was done in such a way that identity numbers 1-5 represent Ms. A, 6-10 for Mr. B, 11-15 for Ms. C 16-20 Mr D, and 21-25 Mr E. If we compare this results with the ones obtained previously above it will be observed that the system was able to identify each individual correctly.

4: SUMMARY AND FIELDS OF APPLICATION

The implementation was accomplished first by visual recognition system implemented using The Principal Component Analysis, Linear Discriminant Analysis and Nearest Neighbour Classifier. The system was tested using images that have not been trained to the system to see whether it can detect an intruder which led us to a very successful result with precised response to intruder. For verification purpose, we explored another means of implementing the visual recognition system using a Neural Network, the work on visual recognition system was converted into a simulink block set which was then implemented in a Signal wave and the system was able to identify each individual correctly. For biometric system in its identification mode may be deployed to monitor surveillance cameras and/or the telephone system within a school campus or residential to identify known specific individuals who may have been excluded from parts or all of the facilities in compound. These could be known debtors, troublemakers, shoplifters etc. In this mode the system will have been supplied with template information for specific individuals and will continuously search for a match or difference with the facial information. Although tremendous efforts were made in the field but most of this system is customized and not affordable by general public, it is the effort of the study to generate very simple algorithm which has a potential of adaptability to many different conditions and could be simply interfaced with many electromechanical or otherwise through very simple digital interfacing

5: CONCLUSION

The results showed the accuracy of the visual recognition system in identifying test images tested against the trained images. When the system was tested with images it wasn’t trained with, it was able to give a feedback that an intruder was detected. Thus the PCA, LDA and Nearest Neighbor algorithm are powerful tools in developing such systems. By increasing the number of training images as well as adjusting the parameters picnum and ppl the system can be upgraded to identify an unlimited number of clients, audio recognition was implemented and second voice C is identified as extruder and system has given a full assurance that it will reject any fraud voice

6: ACKNOWLEDGEMENT

The authors wish to thank Universiti Malaysia Perlis (UniMAP), Ministry of Science, Technology & Innovation (MOSTI) and Ministry of Higher Education for giving the opportunities to do this research in the Micro & Nano simulation and Fabrication Lab for providing us a system to generate the codes. The appreciation also goes to all the team members in the Institute of Nanoelectronic Engineering especially in the Nano Biochip Research Group and electronics communication research group for useful discussions.

7: REFERENCE

[1] Raychaudhuri, S., Stuart, J.M. and Altman, R.B. Principal components analysis to summarize microarray experiments: application to sporulation time series. Pacific Symposium on Biocomputing (2000).

[2] Jonathon Shlens, A Tutorial on Principal Component Analysis Center for Neural Science, New York University New York City, NY 10003-6603 and Systems Neurobiology Laboratory, Salk Insitute for Biological Studies La Jolla, CA 92037(2009)

[3] S. Gong et al., Dynamic Vision: from Images to Face Recognition, Imperial College Press, 2001 (pp. 168-173 and Appendix C: Mathematical Details, hard copy).

[4] H. Murase and S. Nayar, "Visual Learning and Recognition of 3D Objects from Appearance", *Interantional Journal of Computer Vision*, vol 14, pp. 5-24, 1995 (hard-copy)

[5] D. L. Swets and J. Y. Weng. Using discriminant eigenfeatures for image retrieval. IEEE Trans.Pattern Analysis and Machine Intelligence, 18(8):831–836, 1996.

[6] W. Liu, Y. Wang, S. Z. Li, and T. Tan. Null space approach of Fisher discriminant analysis for face recognition. In Proc. European Conference on Computer Vision, Biometric Authentication Workshop, 2004.

[7] Z. Jin, J. Y. Yang, Z. S. Hu, and Z. Lou. Face recognition based on the uncorrelated Discriminant transformation. Pattern Recognition, 34:1405–1416, 2001

[8] M. A. Turk and A. P. Pentland. Face recognition using Eigenfaces. In Proc. Computer Vision and Pattern Recognition Conference, pages 586–591, 1991.

[9] KAI YU, LIANG JI* and XUEGONG ZHANG, Kernel Nearest-Neighbor Algorithm, Neural Processing Letters 15: 147^156, 2002 Tsinghua University, Beijing, P.R. China, 100084; tel: 86-10-62782877 fax: 86-10-62784047)

[10] Stéphane Dupont and Juergen Luettin, Audio-Visual Speech Modeling for Continuous Speech Recognition, Eeee transactions on multimedia, vol. 2, no. 3, september 2000.

[11] E. Martinian, S. Yekhanin, and J. Yedidia, “Secure biometrics via syndromes,” in Proc. Allerton Conf. Commun., Contr. And Comput., Allerton, IL, 2005.

[12] D. Varodayan, Y.-C. Lin, A. Mavlankar, M. Flierl, and B. Girod, “Wyner-Ziv coding of stereo images with unsupervised learning of disparity,” in Proc. Picture Coding Symp.Lisbon, Portugal, 2007.

[13.] Abu-Mostafa, Y. (1993), “Hints and the VC Dimension”, Neural Computation,Vol. 5, No. 2, pp. 278–288.

[14] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, “Active shape models—their training and application,” *Comput. Vis. Image Understand.*,vol. 61, pp. 38–59, Jan. 1995.

[15] J R Parks, “Automated Personal Identification Methods for Use with Smart Cards”, Chapter 7 in Integrated Circuit Cards, Tags and Tokens edited by P Hawkes, D Davies and W Price, BSP, London, ISBN 0-632-01935-2, 1990

[16] F Deravi, “Audio-Visual Person Recognition for Security and Access Control”, JTAP project JTAP-552, local web site, URL: http://eleceng.ukc.ac.uk/~fd4/jtap.html, 1998-1999

[17] L Burbridge, Experience with the use of a multi-purpose smart card, JTAP Report 019, JISC, March 1998.

[18] C C Chibelushi, J S D Mason and F Deravi, “Feature-level Data Fusion for Bimodal Person Recognition”, Sixth International Conference on Image Processing and its Applications, IEE, Trinity College, Dublin, Ireland, , pp 339-403, 14-17 July, 1997

[19] J D Woodward, “Biometrics: Privacy’s Foe or Privacy’s Friend?”, Proceedings of the IEEE, Vol.85, No. 9, September 1997.

INSTRUCTIONS TO CONTRIBUTORS

The *International Journal of Computer Science and Security (IJCSS)* is a refereed online journal which is a forum for publication of current research in computer science and computer security technologies. It considers any material dealing primarily with the technological aspects of computer science and computer security. The journal is targeted to be read by academics, scholars, advanced students, practitioners, and those seeking an update on current experience and future prospects in relation to all aspects computer science in general but specific to computer security themes. Subjects covered include: access control, computer security, cryptography, communications and data security, databases, electronic commerce, multimedia, bioinformatics, signal processing and image processing etc.

To build its International reputation, we are disseminating the publication information through Google Books, Google Scholar, Directory of Open Access Journals (DOAJ), Open J Gate, ScientificCommons, Docstoc and many more. Our International Editors are working on establishing ISI listing and a good impact factor for IJCSS.

The initial efforts helped to shape the editorial policy and to sharpen the focus of the journal. Starting with volume 5, 2011, IJCSS appears in more focused issues. Besides normal publications, IJCSS intend to organized special issues on more focused topics. Each special issue will have a designated editor (editors) – either member of the editorial board or another recognized specialist in the respective field.

We are open to contributions, proposals for any topic as well as for editors and reviewers. We understand that it is through the effort of volunteers that CSC Journals continues to grow and flourish.

IJCSS LIST OF TOPICS
The realm of International Journal of Computer Science and Security (IJCSS) extends, but not limited, to the following:

- Authentication and authorization models
- Computer Engineering
- Computer Networks
- Cryptography
- Databases
- Image processing
- Operating systems
- Programming languages
- Signal processing
- Theory
- Communications and data security
- Bioinformatics
- Computer graphics
- Computer security
- Data mining
- Electronic commerce
- Object Orientation
- Parallel and distributed processing
- Robotics
- Software engineering

CALL FOR PAPERS

Volume: 6 - **Issue:** 2 - April 2012

i. Paper Submission: January 31, 2012 **ii. Author Notification:** March 15, 2012

iii. Issue Publication: April 2012

CONTACT INFORMATION

Computer Science Journals Sdn BhD
B-5-8 Plaza Mont Kiara, Mont Kiara
50480, Kuala Lumpur, MALAYSIA

Phone: 006 03 6207 1607
006 03 2782 6991

Fax: 006 03 6207 1697

Email: cscpress@cscjournals.org

CSC PUBLISHERS © 2011
COMPUTER SCIENCE JOURNALS SDN BHD
M-3-19, PLAZA DAMAS
SRI HARTAMAS
50480, KUALA LUMPUR
MALAYSIA

PHONE: 006 03 6207 1607
006 03 2782 6991

FAX: 006 03 6207 1697
EMAIL: cscpress@cscjournals.org