Text Steganalysis Using Evolution Algorithm Approach

ROSHIDI DIN[1], T. ZALIZAM T. MUDA[2], PURIWAT LERTKRAI[3], MOHD NIZAM OMAR[4], ANGELA AMPHAWAN[5], FAKHRUL ANUAR AZIZ[6],
InterNetWorks Research Laboratory[4, 5], School of Computing[1, 2, 3], UUM College of Arts and Sciences
Universiti Utara Malaysia
UUM Sintok, 06010, Kedah
MALAYSIA
roshidi@uum.edu.my[1], zalizam@uum.edu.my[2], puriwat_lertkrai@hotmail.com[3],
niezam@uum.edu.my[4], angela@uum.edu.my[5], fakhrul@uum.edu.my[6]
http://www.internetworks.my[4,5] http://www.soc.uum.edu.my[1,2,3,6]

Abstract: - This study presents a new alternative of steganalysis method in order to detect hidden messages in text steganalysis called Evolution Detection Steganalysis System (EDSS) based on the evolution algorithm approach under Java Genetic Algorithms Package (JGAP). The result of the EDSS can be divided into two groups based on fitness values which are good fitness and bad fitness. Hopefully, this study can produce a good idea to other researchers for understanding the text steganalysis in order to develop a steganalysis system that can contribute a better performance in other domains.

Key-Words: - Text steganalysis, steganography, evolution algorithm, computational intelligence

1 Introduction

Steganography is the art and science of communicating in such a way that the presence of a hidden message cannot be detected [1]. There are two aspects of steganography, namely the technical steganography and natural language steganography. Technical steganography concentrates on channel capacity which is concerned about a cover medium to hide messages, while natural language steganography concentrates on using written natural language to conceal secret messages [2]. Technical steganography work has been carried out on image steganography [3, 4], video steganography [5, 6], and audio steganography [7] which have produced good results. On the other hand, natural language steganography is the art of using natural language to conceal secret messages. It focuses on hiding information in text by using text steganography and linguistic steganography. Currently, natural language steganography is developed based on an attracting method of the steganography itself, which is called steganalysis [8].

Surprisingly, very little work has attempted to formalise steganalysis. This is due largely to the relative lack of redundant information in a natural language in comparison with an image, video, or audio. A few detection algorithms in natural language steganalysis has been proposed, which includes a statistical analysis of a kind of word-shift text steganography that contributes to both text-steganalysis and text-steganography by using neighbour difference (length difference of two consecutive spaces) in PDF text document [9]. Other studies had proposed a steganalysis method based on a dictionary, for example, the MobyDick algorithm can simultaneously find hundreds of different words and each of them present in a small subset of the sequences. In a kind of character distribution, another study had proposed a steganalysis for text steganography based on font format that uses Support Vector Machine (SVM) to train the classifier and use the resulting trained classifier to detect the existence of hidden information within the text document [10]. Based on Chandramouli and Subbalakshmi [11] who had studied a critical analysis for most steganalysis methodologies, found that steganalysis with CI approaches can be implemented to solve steganalysis problems. One of the strongest methods in CI methods is the Evolution Algorithm (EA) especially genetic algorithm. This is because; EA is a better solution to solve complex problems [12] which is able to produce a systematic rule for feature selection of solution and it is very powerful for optimisation [13]. However, it has been found to be effective in audio steganalysis [14] and image steganalysis [15].

Thus, the main objective of this study is to design and develop an EA method for natural language steganalysis. This study presents a new

alternative steganalysis method in order to detect hidden messages in text steganalysis based on the dictionary approach due to justify the correct words within a text file. It is only considered a text based document by using the EA approach under Java Genetic Algorithms Package (JGAP).

2 Evolution Algorithm Approach

EA is the idea of evolution, and as evolution itself must have evolved to reach its current state, it is used not only for finding solutions to solve complex problems, but also used to fine-tune the algorithm to a particular problem and can be hybridised with other techniques. This is because the flexibility of EA that can handle general optimisation problems using virtually any reasonable representation and performance [16]. EA also can be used to develop classification of adaptation based on the mechanism of adaptation and level in occurrence [17]. Meanwhile, EA can be applied with any type of cost functions that do not require any high order information. Based on EA, cost functions may not always be used to compute with poor numerical accuracy. EA is one method that can be used to evaluate the population in parallel, because it has also implemented several mechanisms and selection strategies developed to support this type of parallelism [18].

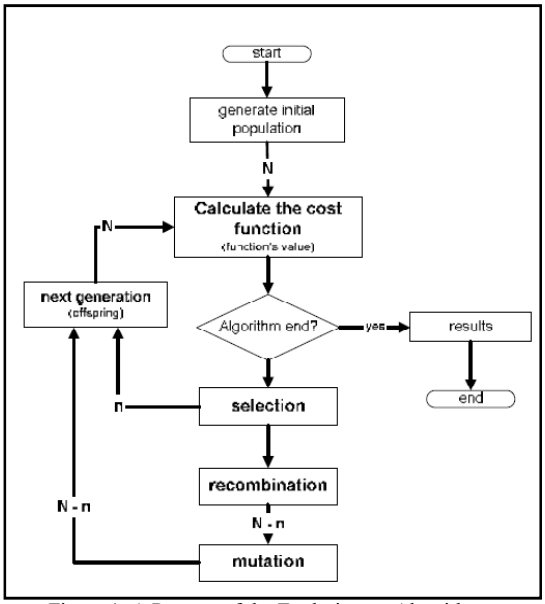

Figure 1: A Process of the Evolutionary Algorithms

EA is also used in a network area, where the application of EA can optimise networks and improve robustness to protect the network from attacks with a significant success [19]. The term optimisation refers to a function that is maximised or minimised and it is evaluated for every individual. The selection will choose the best gene combinations (individuals), which through crossover and mutation should generate better solutions in the next population. One of the most often used schemes of EA is shown in Figure 1. Algorithm 1 has shown the flow of Evolution Algorithm used in this study.

Algorithm 1: Evolution Algorithm

1. *Generate initial population* – first generation is randomly generated, by selecting the genes of the chromosomes among the allowed alphabet for the gene.
2. *Calculation of the values* of the function that is required to minimise or maximise.
3. *Check for termination of the algorithm* – for most optimisation algorithms, it is possible to stop the genetic optimisation by:
   - *Value of the function* – the value of the function of the best individual is within a defined range around a set value.
   - *Maximal number of iterations* – this is the most widely used stopping criteria. It guarantees that the algorithms will give some results within some required time.
   - *Stall generation* – if within initially set number of generations there is no improvement of the value of the fitness function of the best individual, the algorithms stops.
4. *Selection* – between all individuals in the current population are chosen those who will continue, and by means of crossover and mutation, will produce an offspring population.
5. *Recombination* – the individuals chosen by selection recombine with each other and new individuals will be created. The aim is to get offspring individuals.
6. *Mutation* – by means of random change of some of the genes, it is guaranteed that even if none of the individuals contain the necessary gene value for the extremum, it is still possible to reach the extremum.
7. *New generation* – the elite individuals chosen from the selection are combined with those who passed the crossover and mutation, and form the next generation.

3 Text Steganalysis Model

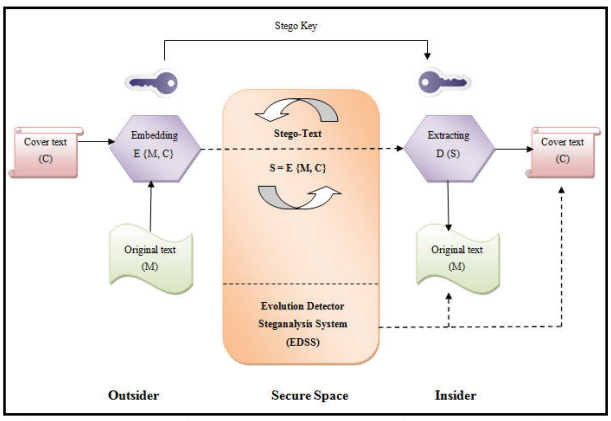

Figure 2: A Steganology Processes on Natural Language Environment

The model for detecting data within the hidden data can be described as follows. The embedded data is the message that one wishes to send secretly. It is usually hidden in an appropriate text, showing the stego-text or other stego-object. A stego-key is used to control the hiding process so as to restrict detection and/or recovery of the embedded data to parties who know it. The processes of steganography and steganalysis can be represented in Figure 2.

The Outsider and Insider are locked up in separate cells far apart from each other. They are allowed to communicate by means of sending messages via Secure Space who does not suspect such communication is taking place. Secure Space who plays the role of the adversary will break all communication that comes into space. If Secure Space detects any sign of conspiracy, it will suppress all messages. Both the Outsider and Insider are well aware of these facts. Let us assume that,

S *stego-cover*
M *hidden message*
C *cover-message*
fe *embedding message*

Thus, the Outsider is trying to send a hidden message M, within a cover message C, which involves a stego key K through an embedding process known as S. The first step is applying the invertible function e: $\{M, C\} \rightarrow S$. Then, the Outsider can map a hidden message M to a stego message S, using key K through $e(M, C) = S$. Since S is a stego message, Secure Space will not find it suspicious, and since the function is invertible, the Insider will be able to compute $e\text{-}1(S) = \{M, C\}$ in order to reconstruct the hidden message M and cover message C with a stego key K. The embedding process f_e of hiding original message M should exploit the embedding key K with the pre-processing random characteristics r (such as white noise) on cover-message C as f_p is known as actual cover Cr [20].

$$S = f_e(C, M, K)$$
$$S = f_p(C, r) + M + K$$
$$S = C_r + M + K \qquad (1)$$

At the same time, Secure Space can also use this information to decide the presence or absence of a hidden message.

From the conditional state of steganography system, the only knowledge available is that

$$y(k) = s(k) + \alpha w(k),\ k = 1, 2, \ldots, N \qquad (2)$$

where

$y(k)$ *analyzed message*
$s(k)$ *cover-message*
$w(k)$ *stego-cover*
α *message strength $\alpha > 0$ based on perceptual characteristics, robustness properties etc.*

It can be assumed that the signal distribution of analysed message $y(k)$ and common transform coefficient distribution of cover-message $w(k)$ is justified as a Gaussian distribution.

3.1 Developing the Evolution Detection Steganalysis System (EDSS)

Evolution Detection Steganalysis System (EDSS) has been developed with JAVA programming language by using Netbean IDE 6.9.1 as a tool to develop the system. The NetBeans IDE is a tool integrated development environment available for Windows, Mac, Linux, and Solaris. The NetBeans project contains an open-source IDE and an application platform that help developers to create GUI applications easier. The user interface was designed with this tool as shown in Figure 3 and explanations of EDSS components in Table 1.

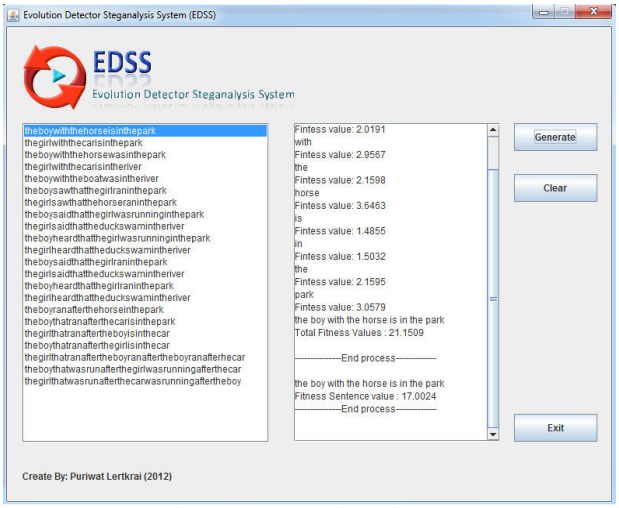

Figure 3: User Interface of EDSS

The EDSS used JGAP to provide basic genetic mechanisms that can be easy to use for applying evolution principles to problem solutions. The system will come out with correct words and fitness values of each word and sentence that are generated by EDSS. Thus, this study used the weight of each character to calculate the fitness value, which means that the weight represents the fitness.

Table 1 : Components of EDSS

Number	Component	Description
1	Stego text area	The area to show the stego-text that is imported from *HiddenStegoText.TXT*.
2	Result area	The area to show the result from the process of EDSS.
3	Generate Button	The button to start the process of EDSS.
4	Clear Button	The button to clear all the text in result area.
5	Exit Button	The button to close the system.

3.2 Implementation of EDSS

The EDSS algorithm can be separated into two parts. The first part of the algorithm if for words and the second part is the algorithm for sentences. The algorithm for words is implemented according to the following steps:

i. First step: import the important files, which are *StegoDictionary.txt* and *HiddenStegoText.TXT*, to the system for generating correctly words based on this dictionary.
ii. Second step: select one line from *HiddenStegoText.TXT* refer to variable "*S*" and store into variable char array "*c*". Thus, the formula of this step is $S = \{c_0, c_1, c_2, \ldots c_n\}$.
iii. Third step: compare "S" with *StegoDictionary.txt* since first character until end of character in "*c*". If c_0 matches an entry in the dictionary, then store character in String variable "*w*", go to step 4. If c_0 does not match the dictionary, thus the formula of this step is $w = \sum_0^n c(n)$.
iv. Fourth step: get word "*w*" from previous step, and import *StegotextWeight.TXT*, after that get weight for each character from this file to calculate fitness value. For example:

Position	c_0	c_1	c_2
Character	b	o	y
Weight	*0.774*	*0.637*	*0.871*

v. Fifth/Final step: get weights of each character to calculate in fitness function, identify variable "*F*". The "*F*" function will return the fitness value "*f*" of each word that matched entries in the dictionary from step 3. Thus, the formula for total fitness value of this step is $f = \sum_0^n (F(n) \sum_0^n w(n))$. Then, fitness function process will be determined as 2.145.

The second part of this algorithm in detecting sentences is shown in the following steps:

i. First step: import *HiddenStegoText.TXT*.
ii. Second step: select one line from *HiddenStegoText.TXT*, refer to variable "S" and store into variable char array "*c*". The formula of this step is $S = \{c_0, c_1, c_2, \ldots c_n\}$.
iii. Third step: import *StegotextWeight.TXT* and get sentence from previous step. After that used loop to find the weight of each character.

Position	c_0	c_1	c_2	c_3	c_4	c_5
Character	T	h	e	b	o	y
Weight	0.778	0.825	0.612	0.774	0.637	0.871

vi. Fourth step: import *RoshidiDictionary.txt* to EA system to find the correct words of the sentence and calculate fitness value of this sentence. Thus, the formula of this step is $f = F(\sum_o^n w(n))$. Then, fitness function process will be determined as 17.521.

Table 2 below shows the data set called stego-text named *HiddenStegoText.TXT*. This is a text file containing 22 lines of *HiddenStegoText.TXT* taken from [21]. The file consists of 893 bytes in size and occupies 4.00 KB of memory on disk.

Table 2: *HiddenStegoText.TXT*

Text	StegoText
T_i	*theboywiththehorseisinthepark*
T_{i+1}	*thegirlwiththecarisinthepark*
T_{i+2}	*theboywiththehorsewasinthepark*
T_{i+3}	*thegirlwiththecarisintheriver*
T_{i+4}	*theboywiththeboatwasintheriver*
T_{i+5}	*theboysawthatthegirlraninthepark*
T_{i+6}	*thegirlsawthatthehorseraninthepark*
T_{i+7}	*theboysaidthatthegirlwasrunninginthepark*
T_{i+8}	*thegirlsaidthattheduckswamintheriver*
T_{i+9}	*theboyheardthatthegirlwasrunninginthepark*
T_{i+10}	*thegirlheardthattheduckswamintheriver*
T_{i+11}	*theboysaidthatthegirlraninthepark*
T_{i+12}	*thegirlsaidthattheduckswamintheriver*
T_{i+13}	*theboyheardthatthegirlraninthepark*
T_{i+14}	*thegirlheardthattheduckswamintheriver*
T_{i+15}	*theboyranafterthehorseinthepark*
T_{i+16}	*theboythatranafterthecarisinthepark*
T_{i+17}	*thegirlthatranaftertheboyisinthecar*
T_{i+18}	*theboythatranafterthegirlisinthecar*
T_{i+19}	*thegirlthatranaftertheboyranaftertheboyranafterhecar*
T_{i+20}	*theboythatwasrunafterthegirlwasrunningafterthecar*
T_{i+21}	*thegirlthatwasrunafterthecarwasrunningaftertheboy*

ISBN: 978-1-61804-126-5

4. Result

Before the detecting process, the system does not know what words are included in each line. After generating correctly the words by using EDSS, the result came out with the calculated fitness values. The result of the EDSS can be divided into two groups based on fitness values which are good fitness and bad fitness.

4.1 Good Fitness

Table 3 has shown the sentences with good fitness levels after the detecting process of the hidden messages.

Table 3: Good Fitness Value

Detected Analysed Text	*Fitness Value*
T_i	21.1590
T_{i+1}	20.9527
T_{i+2}	21.9986
T_{i+3}	21.2726
T_{i+4}	22.3126
T_{i+8}	26.5073
T_{i+11}	24.4072
T_{i+12}	26.5081

Good fitness values generated by EDSS based on the word algorithm found or detected eight sentences of the 22 sentences of the text steganalysis, which is equivalent to 36.36% success rate. From the results, it could be observed that the EDSS can detect 100% of the hidden message in each sentence and fitness values are greater than 20. Since, fitness values depend on the weight of each character, if the EDSS can match the words in the dictionary, it would have an effect on the overall fitness values. It can be said that the EDSS will return good fitness values when it can detect all hidden messages within the sentences.

5.2 Bad Fitness

Table 4 has shown the sentences with bad fitness levels after the detecting process of the hidden messages.

Table 4: Good Fitness Value

Detected Analysed Text	*Fitness Value*
T_{i+5}	4.1830
T_{i+6}	5.1884
T_{i+7}	19.8298
T_{i+9}	7.0429
T_{i+10}	8.0477
T_{i+13}	7.0469
T_{i+14}	8.0527
T_{i+15}	6.5308
T_{i+16}	9.6411
T_{i+17}	10.6517
T_{i+18}	9.6487
T_{i+19}	10.6516
T_{i+20}	11.7920
T_{i+21}	12.7962

Bad fitness values generated by the EDSS based on the word algorithm had found 14 sentences of 22 sentences, which an equivalent of 63.63% success rate of text steganalysis. From here, it can be observed that the EDSS can only detect 20-60% of the hidden message in each sentence and thus the fitness values are lower than 20. It can be said that the EDSS will return bad fitness values if the number of words that can be detected by EDSS based on the dictionary is low in number, which means that the fitness values become low, in this study the author uses the term “bad”. Thus, fitness values become lower than the value of 20, because fitness values depend on the weight of each character, and if EDSS cannot match the word in the dictionary, it would have a decreasing effect on the calculated fitness values.

6 Conclusion

This study had developed EDSS based on EA in order to detect hidden messages in text steganalysis. EDSS is an alternative method to detect hidden messages within a text based natural language environment to support the text based document by using EA approach. Hopefully, this study can enable other future researchers to understand better the steganalysis tools in order to develop a text steganalysis system that can contribute to better performance.

References:

[1] C. Cachin, “An Information-Theoretic Model for Steganography”, *Information and Computation Academic Press,* vol. 192(1), pp. 1 – 14, 2004.

[2] M. Chapman, G. I. Davida, and M. Rennhard, “A Practical and Effective Approach to Large-Scale Automated Linguistic Steganography”, Proceedings of the Information Security Conference (ISC '01), pp. 156 – 165, 2001.

[3] N. F. Johnson, and S. Katzenbeisser, “A Survey of Steganographic Techniques”, Information Hiding: Techniques for Steganography and Digital Watermarking, pp. 43 – 78, 2000.

[4] R. Chandramouli, and N. Memon, "Analysis of LSB Based Image Steganography Techniques", Proceedings of the International Conference on Image Processing, vol. 3, pp. 1019 – 1022, 2001.

[5] A. Westfeld, and G. Wolf, "Steganography in a Video Conferencing System", Proceedings of Information Hiding – 2nd International Workshop, pp. 32 – 47, 1998.

[6] G. A. Doerr, and J. L. Dugelay, "Security Pitfalls of Frame by Frame Approaches to Video Watermarking", *IEEE Transactions on Signal Processing*, vol. 51(10), pp. 2955 – 2964, 2004.

[7] K. Gopalan, "Audio Seganography Using Bit Modification", Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing *(ICASSP '03)*, vol. 2, pp. 421 – 424, 2003.

[8] C. Kevin, and B. Karen, "An Evaluation of Image Based Steganography Methods", Multimedia Tools and Applications, vol. 30(1), pp. 55 – 88, 2006.

[9] L. Li, L. Huang, X. Zhao, W. Yang, and Z. Chen, "A Statistical Attack on a Kind of Word-Shift Text-Steganography", International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp. 1503 – 1507, 2008.

[10] T. T. Liu, and W. H. Tsai, "A New Steganographic Method for Data Hiding in Microsoft Word Documents by a Change Tracking Technique", *IEEE Transactions on Information Forensics and Security*, vol. 2, pp. 24 – 30, 2007.

[11] R. Chandramouli, and S. K. Subbalakshmi, "Steganalysis: A Critical Survey", Control, Automation, Robotics and Vision Conference (ICARCV), vol. 2, pp. 964 – 967, 2007.

[12] G. Wu, "A Neural Network used for PD Pattern Recognition in Large, Turbine Generators with Genetic Algorithm", Conference Record of the 2000 IEEE International Symposium on Electrical Insulation, pp. 1 – 4, 2000.

[13] D. T. Pham, and D. Karaboga, *Intelligent Optimisation Techniques: Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks*, 1st Edition, Springer Press, 2000.

[14] S. Geetha, S. S. Sivatha Sindhu, and A. Kannan, "StegoBreaker: Audio Steganalysis using Ensemble Autonomous Multi-Agent and Genetic Algorithm", Annual India Conference, New Delhi, pp. 1-6, September 2006.

[15] A. M. Fard, M. R. Akbarzadeh-T, and, F. Varasteh-A, "A New Genetic Algorithm Approach for Secure JPEG Steganography", IEEE International Conference on Engineering of Intelligent Systems, pp. 1 – 6, 2006.

[16] D. B. Fogel, "The advantages of Evolutionary Computation", Proceedings of Bio Computing and Emergent Computation (BCEC' 97), pp. 1 – 11, 1997.

[17] R. Hinterding, Z. Michalewicz, and A. E. Eiben, "Adaptation in Evolutionary Computation: A Survey", IEEE International Conference on Evolutionary Computation, pp. 65 – 69, 1997.

[18] D. Whitley, "An Overview of Evolutionary Algorithms: Practical Issues and Common Pitfalls", *Information and Software Technology,* vol. 43, pp. 817 – 831, 2001.

[19] N. Yazdani, H. Herrmann, F. Daolio, and M. Tomassini, "EA opitmization of Networks Against Malicious Attacks", pp. 1 – 10, 2011.

[20] R. Din, Z. Che Ani, and A. Samsudin, "A Formulation of Conditional States on Steganalysis Approach", *WSEAS Transactions on Mathematics,* vol. 11(3), pp. 173 – 182, 2012.

[21] J. Davila, "Genetic Optimization of Neural Network Topologies for the Task of Natural Language Processing", International Joint Conference on Neural Networks, vol. 2, pp. 821 – 826, 1999.