Fitness Value Based Evolution Algorithm Approach for Text Steganalysis Model

Roshidi Din, Azman Samsudin, T. Zalizam T. Muda, P. Lertkrai, Angela Amphawan, and Mohd. Nizam Omar

***Abstract*—**In this paper, we present a new alternative method for text steganalysis based on an evolution algorithm, implemented using the Java Evolution Algorithms Package (JEAP). The main objective of this paper is to detect the existence of hidden messages based on fitness values of a text description. It is found that the detection performance has been influenced by two groups of fitness values which are good fitness value and bad fitness value. This paper provides a valuable insight into the development and enhancement of the text steganalysis domain.

***Keywords*—**Text steganalysis, Natural language steganalysis, Evolution algorithm, Fitness values.

I. Introduction

IN recent decades, the application of computers in different realms of life and work has come to the fore. One of the many issues raised is the security of information, which has gained considerable importance. This is because information is now treated as a commodity or a resource comparable to labour and capital [1]. One of the concerns in the area of information security is the concept of information hiding [2]. The information hiding concept has received attention from the research community and has been rapidly evolving since the first academic conference on the subject organised in 1996. Information hiding is a broad term for a scientific discipline that studies various topics such as covert and subliminal communication channels, detection of hidden information, watermarking of digital object, fingerprint, and anonymity services. There are two main directions in information hiding; firstly to protect against the detection of secret messages by a passive adversary, and secondly to hide data so that even an active adversary cannot access or modify the data [3]. A survey of current information hiding is given by [4] which revealed a recent important sub-discipline known as steganography.

Steganography is the art and science of communicating in such a way that the presence of a hidden message cannot be detected [5]. There are two aspects of steganography, namely 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 [6]. Technical steganography has been carried out on image steganography [7, 8], video steganography [9, 10], and audio steganography [11] 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, text steganography is developed based on an attracting method of the steganography itself, which is called text steganalysis [12].

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 text steganalysis have 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 [13]. Other studies have 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 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 [14]. Based on Chandramouli and Subbalakshmi [15] who had studied a critical analysis for most steganalysis methodologies, found that steganalysis with CI approaches can be implemented to solve steganalysis problems. The ultimate goal of CI

R. Din is with the School of Computing, UUM College of Arts and Sciences (CAS), Universiti Utara Malaysia, Sintok, 06000 Kedah, Malaysia (corresponding author to provide phone: 6017-598-1306; fax: 604-9284753; e-mail: roshidi@uum.edu.my).

A. Samsudin is with the School of Computer Sciences, Universiti Sains Malaysia, 11800 Penang, Malaysia. (e-mail: azman@cs.usm.my).

T. Zalizam is with the School of Multimedia Technology and Communication (SMMTC), UUM College of Arts and Sciences (CAS), Universiti Utara Malaysia, Sintok, 06000 Kedah, Malaysia. (e-mail: zalizam@uum.edu.my).

P. Lertkrai is with the Rajamangkala University of Technology Srivijaya, Thailand. (e-mail: puriwat_lertkrai@hotmail.com).

A. Amphawan is with the School of Computing, UUM College of Arts and Sciences (CAS), Universiti Utara Malaysia, Sintok, 06000 Kedah, Malaysia. (e-mail: angela@uum.edu.my).

M.N. Omar is with the School of Computing,, UUM College of Arts and Sciences (CAS), Universiti Utara Malaysia, Sintok, 06000 Kedah, Malaysia. (e-mail: niezam@uum.edu.my).

This work was fully supported by 12423/ UUM-LEADS Grant, Universiti Utara Malaysia under the Higher Education Minister of Malaysia.

approaches is to create cognitive systems that could compete with humans in a large number of areas. The output of a CI approaches includes predictions and/or decisions such as done in [16, 17, 18].

Recently, many methods in digital steganalysis have been presented by researchers. All of these methods can be classified into two types which are identified as statistical steganalysis and CI steganalysis. Statistical steganalysis consists of Linear Regression such as Simple Pair Analysis, and Regular and Singular Analysis, Support Vector Machine, and Information Theory. CI includes methods such as neural networks, evolutionary computation (Genetic Algorithms (GA), Swarm Intelligence (SI) and Evolution Algorithm (EA)), and other optimisation algorithms. Meanwhile, GA is part of the EA group [19], which are techniques for handling uncertainty, such as Bayesian, fuzzy logic, and certainty theory. Bayesian and Certainty Theory appear in both of these two steganalysis classifications, as shown in Fig.1. One of the strongest methods in CI approach is the EA which can be utilized in text steganalysis. This is because; EA is a better solution to solve complex problems [20] which is able to produce a systematic rule for feature selection of solution and it is very powerful for optimisation [21]. However, it has been found to be effective in audio steganalysis [22] and image steganalysis [23].

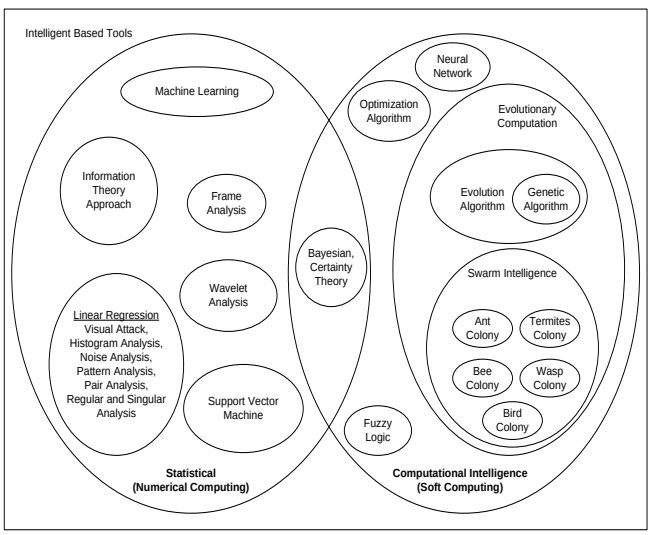

Fig.1. The Paradigm of Digital Steganalysis (adapted from [24])

II. Evolution Algorithm Description

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 [25]. EA also can be used to develop classification of adaptation based on the mechanism of adaptation and level in occurrence [26]. Meanwhile, EA can be applied on any type of cost function 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 [27]. EA is also used in networking, where the application of EA can optimise networks and improve robustness to protect the network from attacks with a significant success [28]. 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 Fig.2. **Algorithm 1** has shown the flow of EA used in this study.

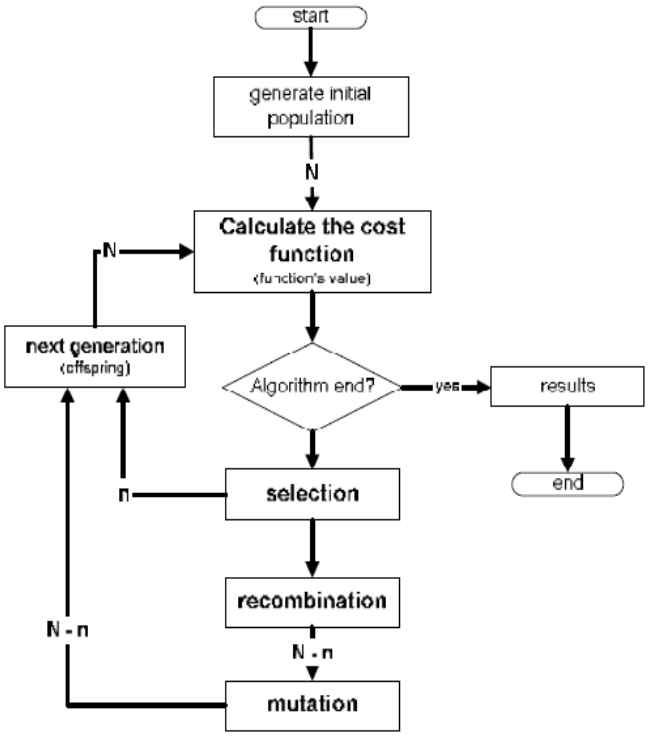

Fig.2. A process of the evolutionary algorithms

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.

III. Text Steganalysis: Statistical Based

Commonly, text steganalysis tries to find a good combination pattern of expected hidden messages in the natural language text itself. There are three major types of statistical detection on text steganalysis domain which are word-counting, dictionary based, characters distribution, and perplexity based.

A. Word-counting

This steganalysis approach can identify a large number of putative motifs. However, the statistical approach usually lacks accurate statistical models and suffers from the problems of producing too many spurious motifs. So that, this steganalysis approach uses the statistical characteristics of correlations between the general stego-text segments and normal text segments to detect stego-text that is generated by three different text steganography approaches: Markov-Chain-Based, NICETEXT, and TEXTO [29].

B. Dictionary Based

The concepts of dictionary and word usage frequencies for constructing sequences is when the detection algorithm is presented as a decomposed set of word DNA sequences; the most probable dictionary of words. An innovative dictionary based on motif finding algorithm for natural language steganalysis is WordSpy [30]. One significant feature of WordSpy is the combination of a word counting method and a statistical model. In essence, WordSpy is a deciphering algorithm that learns a dictionary and statistical model to model the stegoscript.

C. Characters Distribution

This method proposes one set of combination text models to analyse natural text. Different models analyse text in different ways, but they get similar conclusions, thus it can be summarised that one can apply different models for different needs and demands. A method has been proposed where a steganalysis for text steganography is based on font format that uses Support Vector Machine (SVM) to train the classifier and in turn uses the trained classifier to detect the existence of hidden information within a text document [31]. Meanwhile, another detection method has introduced a Tag-Mismatch algorithm to detect the hidden information embedded in letters of tags in a web page [32]. In 2009, Zhao *et al*. [33] had proposed a method to detect the existence of hidden information in mixed texts of English and Chinese using character substitution in texts by SVM as a classifier to classify the characteristic vector input to SVM.

D. Perplexity Based

This is a steganography detection algorithm based on perplexity. The research examines the drawbacks of the NICETEXT system, aiming to accurately classify stego-text and normal text in small size. An experiment showed that if they create a Language Model from normal text, then use it to calculate the perplexity of normal text and stego-text, the result is similar [34].

IV. Text Steganalysis Model

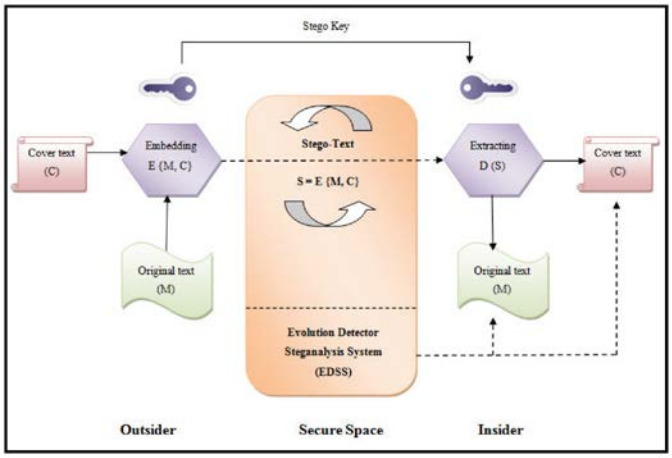

Fig.3. Steganology view on the text environment

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 Fig.3. 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 text*
M	*original text*
C	*cover text*
$E(M, C)$	*embedding text*

Thus, the Outsider is trying to send a hidden message known as original text M, within a cover text C, which involves a stego key K through an embedding process known as $E(M, C)$. The first step is applying the invertible function $e: \{M, C\} \rightarrow S$. Then, the Outsider can map a original text M to a stego text 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 $d(S) = \{M, C\}$ in order to reconstruct the original text M and cover text C with a stego key K. The embedding process f_e of hiding original text M should exploit the embedding key K with the pre-processing random characteristics r (such as white noise) on cover text C as f_p is known as actual cover Cr [35].

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

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 text*
$s(k)$	*stego text*
$w(k)$	*cover text*
α	*text strength* $\alpha > 0$ *based on the fitness value*

It can be assumed that the signal distribution of analysed text $y(k)$ and common transform coefficient distribution of cover text $w(k)$ is justified as a Gaussian distribution. Therefore, this study is tried to manipulate the text strength (fitness values) in order to detect the existing message on the analyzed text.

V. A Modeled Process of Steganalysis System

A development process for a steganalysis system is modeled with JAVA Genetic Algorithm programming (JGAP) language by using Netbean IDE 6.9.1. This system is known as Evolution Detection Steganalysis System (EDSS). The user interface of EDSS is shown in Fig.4 and the explanation of EDSS components are given in Table 1.

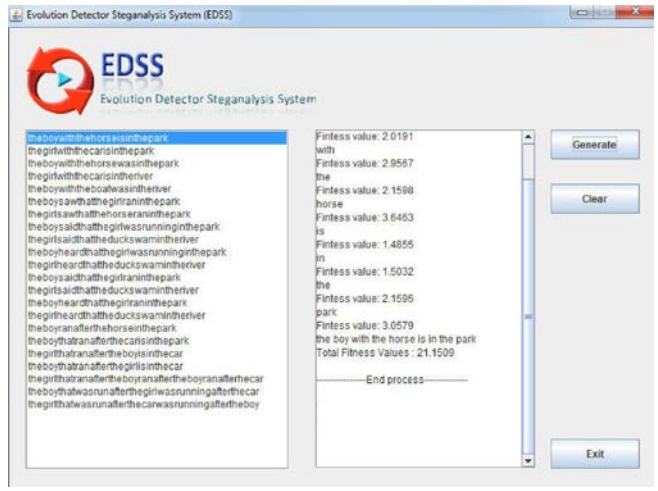

Fig.4. User interface of EDSS

Table 1. EDSS Components

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.

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

The EDSS algorithm can be separated into two parts. The first part of the algorithm is 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. For example:

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

iv. Fourth/Last step: import *StegoDictionary.txt* to EDSS to find the correct words of the sentence and calculate fitness value of this sentence. Thus, the formula of this step is $f = F(\sum_0^n w(n))$. 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 [36]. The file consists of 893 bytes in size and occupies 4.00 KB of memory on $_{i-1}$ disk.

Table 2. *HiddenStegoText.TXT*

Text_ID	*StegoText*
T_{i-1}	*theboywiththehorseisinthepark*
T_{i-2}	*thegirlwiththecarisinthepark*
T_{i-3}	*theboywiththehorsewasinthepark*
T_{i-4}	*thegirlwiththecarisintheriver*
T_{i-5}	*theboywiththeboatwasintheriver*
T_{i-6}	*theboysawthatthegirlraninthepark*
T_{i-7}	*thegirlsawthatthehorseraninthepark*
T_{i-8}	*theboysaidthatthegirlwasrunninginthepark*
T_{i-9}	*thegirlsaidthattheduckswamintheriver*
T_{i-10}	*theboyheardthatthegirlwasrunninginthepark*
T_{i-11}	*thegirlheardthattheduckswamintheriver*
T_{i-12}	*theboysaidthatthegirlraninthepark*
T_{i-13}	*thegirlsaidthattheduckswamintheriver*
T_{i-14}	*theboyheardthatthegirlraninthepark*
T_{i-15}	*thegirlheardthattheduckswamintheriver*
T_{i-16}	*theboyranafterthehorseinthepark*
T_{i-17}	*theboythatranafterthecarisinthepark*
T_{i-18}	*thegirlthatranafterthebоyisinthecar*
T_{i-19}	*theboythatranafterthegirlisinthecar*
T_{i-20}	*thegirlthatranafterthebоyranafterthebоyranafterhecar*
T_{i-21}	*theboythatwasrunafterthegirlwasrunningafterthecar*
T_{i-22}	*thegirlthatwasrunafterthecarwasrunningafterthebоy*

VI. Modeling Process of EDSS

The EDSS is used to detect the analysed text which is generated by steganography techniques. Stego-text or analysed text has been imported from *HiddenStegoText.TXT* file. EDSS used *StegoDictionary.txt* to identify the correct words of each line of *HiddenStegoText.TXT* and used *StegotextWeight.TXT* to identify weight of each character to calculate fitness values in EDSS in order to justify which text has good fitness and bad fitness for the analysed text. The process of EDSS works within the public environment between the sender and recipient as shown in Fig.5.

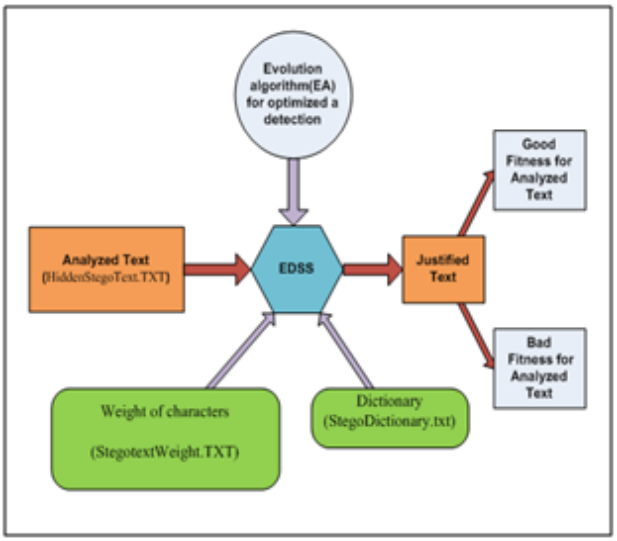

Fig.5. Model of EDSS

The sender embeds the original text in the cover text that is called the stego-text, and in this study *HiddenStegoText.TXT* was used as the stego-text file. Thus between transmissions within the stego-text file, the EDSS will catch the stego-text to detect the hidden message. EDSS can be divided into two parts; first is an algorithm for detecting words and the second part is an algorithm for detecting sentences. The difference between both algorithms is in the process for matching words with the dictionary. In the first part, if the system finds a word, the system will obtain the weights for each character of that word and calculate the fitness value. In the second part, the system will get the weights of each character of that sentence, then calculate and match the word based on a dictionary.

The important function of EDSS is the value of the fitness evaluation function. Fitness is the value assigned to an individual solution. It is based on how far or close the

individual solution is from the actual solution. Typically, a fitness function will determine which possible solutions get passed on to multiply and mutate into the next generation of solutions. EA will discard any individual solution with a low fitness value and accept any with a higher possible fitness value [37]. Thus, this study uses the term "good fitness" to refer to high level of fitness and "bad fitness" to refer to low level of fitness. This function will calculate fitness values depending on maximum allowed evolution time and number of population size as shown in **Algorithm 2**.

Algorithm 2: Function of Fitness Value

```
public double evaluate(IChromosome ic)
{
        ic is Chromosome.
        set defaultComparation to ic with getConfiguration ,
                getFitnessEvaluator and isFitter with parameter
                integer 2 and 1
        set totalWeightOfGene to the getTotalWeightOfGene of ic
        Set fitness to 0
        If defaultComparation is true then
          Set fitness to totalWeightOfGene
        End if
    Return  Math.max(1.0d, fitness)
 }
protected static double getTotalWeightOfGene(Ichromosome
a_potentialSolution)
{
        set weight to 0.0f
        For each genes of chromosome  do
                Set weight to weight plus with values of genes and
                        multiple with weight of each character
                Call getWeightOfGene with parameter chromosome
                        and position of each gene
        End for
        Return weight
   }
protected static double getWeightOfGene(IChromosome
a_potentialSolution,int position)
{
        set weightOfGene to chromosome with
          call getGene with position of gene
          call getAllele refer by getGene
        Return weightOfGene in double type
 }
```

From this pseudo code in EDSS, it can be observed that the system includes three main functions to calculate fitness values, first is *evaluate()* for returning fitness values when the system gets the total weight of each gene, second is *getTotalWeightOfGene()* to get the values of each gene multiplied with the respective weight of each character, the last is *getWeightOfGene()* to get values of each gene by the position of the gene, and return values as a double type to the second function. The fitness function is responsible for this evaluation and must return a positive number that represents the best solution being the highest the number, or the better number, the better the solution.

Normally, fitness values should be an integer number, but in this study the double type was used because the weight of each character is a double number. Finally, high fitness is best suited for situations where a "good" answer will suffice, even if it is not the absolute best answer.

VII. Result

Before the detection process, the system is unable to discover the words in the text analyzed. After generating words from the text by using EDSS, the results are evaluated using fitness values, namely good fitness and bad fitness values.

A. 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-1}	*21.1509*
T_{i-2}	*20.9527*
T_{i-3}	*21.9986*
T_{i-4}	*21.2726*
T_{i-5}	*22.3126*
T_{i-9}	*26.5073*
T_{i-12}	*24.4072*
T_{i-13}	*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. 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. For example, the first sentence of text steganalysis *'theboywiththehorseisinthepark'*, the EDSS can detected a hidden message from this sentence based on the dictionary as follows: *'the boy with the horse is in the park'* and came out with a fitness value of 21.1590. Normally, if we see the sentence of text steganalysis, we can easily understand what it means because we are humans and our brain automatically recognises the words, but in a computer system, it cannot understand what the sentence means and thus would be unable to reveal the hidden message with assistance.

Thus, EDSS has enabled the computer system to understand what it 'means' through the use of weight values for each word. Thus, it can be summarised that if EDSS can detect all the words within each sentence, the fitness values would become high and greater than the value of 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. Meanwhile, fitness values depend on the weight of characters; if the number of words that are generated by EDSS based on the dictionary is high in number, thus the fitness will be expected to be in the range of good fitness values.

B. Bad Fitness

Table 4 has shown the sentences with bad fitness levels after the detecting process of the hidden messages. Bad fitness values generated by the EDSS based on the word algorithm had found 14 sentences of 22 sentences. 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.

Table 4. Bad Fitness Value

Detected Analysed Text	*Fitness Value*
T_{i-6}	*4.1830*
T_{i-7}	*5.1884*
T_{i-8}	*19.8298*
T_{i-10}	*7.0429*
T_{i-11}	*8.0477*
T_{i-14}	*7.0469*
T_{i-15}	*8.0527*
T_{i-16}	*6.5308*
T_{i-17}	*9.6411*
T_{i-18}	*10.6517*
T_{i-19}	*9.6487*
T_{i-20}	*10.6516*
T_{i-21}	*11.7920*
T_{i-22}	*12.7962*

For example, in the sixth sentence of text steganalysis *'theboysawthatthegirlraninthepark'*, the EDSS can detect the hidden message from this sentence based on the dictionary as follows: *'the boy'* and generated a fitness value of 4.1830. From this example, EDSS cannot detect all the hidden messages within that sentence, the EDSS algorithm cannot match the proceeding words in dictionary, which means that the EA in EDSS is unable to calculate good fitness values yet. 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".

In this section, this study postulates that the EDSS cannot detect all the words within each sentence because the EDSS was implemented using a sequential searching mode (line-by-line) for matching the words in the dictionary, thus it may not be able to handle complex words such as "running", since the EDSS will match "run" in the dictionary and return the associated fitness value. This in turn would affect the search and comparison of the next word in the sentence. 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.

VIII. Conclusion

A text steganalysis model based on an evolution algorithm is developed for detecting hidden messages in text steganalysis. This presents an alternative method for detecting hidden messages within a text using a sequential searching mode based on a natural language environment. The developed system provides valuable insights for the enhancement of steganalytic systems in text domains and projects the application of EA for random searches in text steganalysis in the future.

Acknowledgment

We would like thank to Assoc. Prof. Dr. Huda Ibrahim, Dean of School of Computing, Universiti Utara Malaysia and Assoc. Prof. Dr. Mohd Fo'ad Sakdan, Director of Research and Innovation Management Centre (RIMC) for their moral support for the realization of this work.

References

[1] D. J. Hillman, "Problems of the Information Age", *ACM '82 Proceedings of the ACM '82 Conference*, New York, pp. 190 – 191, 1982.

[2] M. H. Shirali-Shahreza, and M. Shirali-Shahreza, "A New Approach to Persian/Arabic Text Steganography" *Proceedings of the 5th IEEE/ACIS International Conference on Computer and Information Science and 1st IEEE/ACIS*, pp. 310 – 315, 2006.

[3] J. Lenti, "Steganographic Methods", *Periodica Polytechnica Ser. El. Eng.*, 44(3 - 4), pp. 249 – 258, June, 2000.

[4] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information Hiding: A Survey", *Proceedings of the IEEE, Special Issue on Protection of Multimedia Content*, vol. 87(7), pp. 1062 – 1078, 1999.

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

[6] 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.

[7] Johnson, N. F. and Katzenbeisser S. (2000). A Survey of Steganographic Techniques. *Information Hiding: Techniques for Steganography and Digital Watermarking*, 43-78.

[8] 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.

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

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

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

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

[13] 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.

[14] 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.

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

[16] M. Memmedli and O. Ozdemir, "An Empirical Study of Fuzzy Approach with Artificial Neural Network Models", *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 6(1), pp. 114 – 121, 2012.

[17] V.E. Neagoe, M. Neghina, and M. Datcu, "Neural Network Techniques for Automated Land-Cover Change Detection in Multispectral Satellite Time Series Imagery", *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 6(1), pp. 130 - 139, 2012.

[18] R. Furferi, L. Governi and Y. Volpe, "Neural Network Based Classification of Car Seat Fabrics", *International Journal of Mathematical Models and Methods in Applied Sciences*, Issue 3, Volume 5(3),pp. 696 – 703, 2011.

[19] A. Popov, "Genetic Algorithms for Optimization – Application in Controller Design Problems", pp. 1 – 21, 2005. *Retrieved from http://p0p0v.com/science/downloads/Popov05a.pdf*

[20] 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.

[21] D. T. Pham, and D. Karaboga, *Intelligent Optimisation Techniques: Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks*, Springer, Berlin, 2000.

[22] S. Geetha, S.S. Sivatha Sindhu, A. and Kannan, "StegoBreaker: Audio Steganalysis Using Ensemble Autonomous Multi-Agent and Genetic Algorithm", *Annual IEEE India Conference*, pp. 1 – 6, 2006.

[23] 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.

[24] R. Din and A. Samsudin, "Digital Steganalysis: Computational Intelligence Approach" *International Journal of Computers*, vol. 3(1), pp. 161 – 170, 2009. ISSN: 1998-4308.

[25] D. B. Fogel, "The Advantages of Evolutionary Computation", *Biocomputing and Emergent Computation: Proceedings of BCEC97*, pp. 1 – 11, 1997.

[26] Hinterding, R., Michalewicz, Z., and Eiben, A. E. (1997). Adaptation in Evolutionary Computation: A Survey. *IEEE International Conference on Evolutionary Computation*, 65-69.

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

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

[29] Z. Chen, L. Huang, Z. Yu, Y. Wei, L. Li, X. Zhen, and X. Zhao, "Linguistic Steganography Detection Using Statistical Characteristics of Correlations between Words", In: Solanki, K., Sullivan, K., Madhow, U. (eds.) IH 2008. LNCS, vol. 5284, pp. 224 – 234. Springer, Heidelberg, 2008.

[30] G. Wang, and W. Zhang, "A Steganalysis-Based Approach to Comprehensive Identification and Characterization of Functional Regulatory Elements", *Genome Biol*., 7(6), R49., 2006.

[31] L. Xiang, X. Sun, G. Luo, and C. Gan, "Research on Steganalysis for Text Steganography Based on Font Format", 3rd *International Symposium on Information Assurance and Security*, pp. 490 – 495, 2007.

[32] H. J. Huang, X. M. Sun, G. Sun, and J. W. Huang, "Detection of Hidden Information in Tags of Webpage Based on Tag-Mismatch", 3rd *International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, vol. 1, pp. 257 – 260, 2007.

[33 X. Zhao, L. Huang, L. Li, W. Yang, Z. Chen, and Z. Yu, "Steganalysis on Character Substitution Using Support Vector Machine", 2nd *International Workshop on Knowledge Discovery and Data Mining*, pp. 84 – 88, 2009.

[34] P. Meng, L. Huang, Z. Chen, W. Yang, and D. Li, "Linguistic Steganography Detection Based on Perplexity", *International Conference on Multimedia and Information Technology*, pp. 217 – 220, 2008.

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

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

[37] S. X. Wu and W. Banzhaf, "A Hierarchical Cooperative Evolutionary Algorithm", In M. Pelikan and J. Branke, editors, *Proceedings of the 12th Genetic and Evolutionary Computation Conference (GECCO '10)*, ACM, New York, Portland, OR, USA, pp. 233 – 240, 2010.

Roshidi Din is a Senior Lecturer at the School of Computing (SoC), UUM College of Arts and Sciences (CAS), Universiti Utara Malaysia (UUM). He received his B.I.T and M.Sc.IT degrees from Universiti Utara Malaysia in 1996 and 1999, respectively. Since working over 14 years in UUM, he has published his work in more than 50 papers in conferences and international journals publication. His current research interests lie in information security, steganology and steganalysis.

Azman Samsudin is an Associate Professor at the School of Computer Sciences, Universiti Sains Malaysia (USM). He received his Ph.D. and M.Sc. degrees from University of Denver in 1998 and 1992, respectively, and his B.Sc. from University of Rochester in 1989. His research interests lie in the areas of computer systems especially cryptography, network security, parallel computing and distributed computing. He has published over 90 academic papers.

Tuan Zalizam Tuan Muda is a lecturer at the School of Communication and Media Technology, UUM College of Arts and Sciences (CAS), Universiti Utara Malaysia (UUM). He received his B.I.T and M.Sc.Comp.Sci degrees from Universiti Utara Malaysia in 1995 and from Fairleigh Dickinson University, New Jersey, USA in 1998, respectively. His current research interests lie in image processing and digital image.

Puriwat Lertkrai is a researcher at the School of Computing (SoC), Universiti Utara Malaysia (UUM). He is currently on leave from the Rajamangkala University of Technology Srivijaya, Thailand. His current research interests lie in information security and steganology.

Angela Amphawan is currently a Senior Lecturer at the School of Computing, UUM College of Arts and Sciences (CAS), Universiti Utara Malaysia. She received her Bachelor of Engineering (Hons) and Master in Engineering Science from Multimedia University, Selangor in 2001 and 2003 respectively. She later completed her Ph.D. from University of Oxford, United Kingdom in 2009. She has published her research work in journals in mathematics, physics, computer science and engineering.

Mohd Nizam Omar is a lecturer at the School of Computing (SoC), UUM College of Arts and Sciences (CAS), Universiti Utara Malaysia (UUM). He received his Bachelor of Computer Science with Honors and Master Science of Computer Science from Universiti Teknologi Malaysia, Skudai, Johor in 2000 and 2005, respectively. He recently received his Ph.D in Computer Science from Universiti Sains Malaysia, Penang, Malaysia in 2011.