A MULTI-LAYER HYBRID TEXT STEGANOGRAPHY FOR SECRET COMMUNICATION USING WORD TAGGING AND RGB COLOR CODING

Ali F. Al-Azzawi[1]

[1]Department of Software Engineering, IT Faculty, Philadelphia University, Amman, Jordan

ABSTRACT

This paper introduces a multi-layer hybrid text steganography approach by utilizing word tagging and recoloring. Existing approaches are planned to be either progressive in getting imperceptibility, or high hiding limit, or robustness. The proposed approach does not use the ordinary sequential inserting process and overcome issues of the current approaches by taking a careful of getting imperceptibility, high hiding limit, and robustness through its hybrid work by using a linguistic technique and a format-based technique. The linguistic technique is used to divide the cover text into embedding layers where each layer consists of a sequence of words that has a single part of speech detected by POS tagger, while the format-based technique is used to recolor the letters of a cover text with a near RGB color coding to embed 12 bits from the secret message in each letter which leads to high hidden capacity and blinds the embedding, moreover, the robustness is accomplished through a multi-layer embedding process, and the generated stego key significantly assists the security of the embedding messages and its size. The experimental results comparison shows that the purpose approach is better than currently developed approaches in providing an ideal balance between imperceptibility, high hiding limit, and robustness criteria.

KEYWORDS

Text Stenography, Python Programming language, Multi-layer encoding, Natural Language Prepossessing, Color space

1. INTRODUCTION

Text steganography can involve whatever besides altering the format of a text content, to editing words within a text, to randomize character sequences, and the use of context-free grammars to create a readable text [1]. Text steganography is accepted to be the trickiest and critical because of insufficiency of redundant data where it is available in an image, sound or a video file and in another hand can easily be discovered[2]. The text file is identical to what we watch, while in different in another sort of file, for example, in the image, the structure of image file is different as what we watch [3]. Unnoticed changes can be made to an image, an audio file or a video file, but, in text files, any change can be noticed by a reader even an extra letter or punctuation. Text steganography is preferable over other steganography approaches, since it used a text file that requires less storing memory and faster sending/receiving time over communication channels[4, 5].

Text steganography can be extensively divided into two sorts: Format based strategy, and Linguistic strategy. Format based strategy deals with changing the physical format of a text to embed information which can be classified by word shift, line shift, extend the space and encode features, while Linguistic strategy deals with natural language to embed information and Linguistic approach can be classified into the syntax and semantics, these methods as shown in Figure[6, 7].

DOI: 10.5121/ijnsa.2018.10601

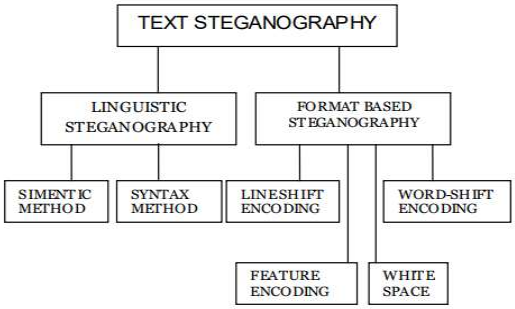

Figure 1. Text steganography types

Linguistic steganography approach considers analyzing linguistic properties of natural language text, and its linguistic structure to hide secret messages[8], and most of these approaches use the steganographic knowledge to hide secret messages inside the syntactical structure itself. The linguistic steganography may use either semantic properties or syntactic properties[9]. In syntactic Steganography includes modifying the text structure without significantly altering the cover text meaning to hide secret messages [10], while in semantic steganography strategy is a way to deal with considered semantic by introducing an adjustment within the meaning of the text to hide secret message [8, 11].

Format based text Steganography can be classified into different types: Format based method which focuses around the inclusion of spaces or non-showed characters, and resizing of text styles [1], but if the stego text is opened, incorrect spellings, additional space characters will get distinguished and changed text styles sizes can stimulate doubt to a human [3].

Text steganography is one among the hardest regions of information hiding since the human eye can be effortlessly recognized any change that shows up between the original text content and the stego text content and it might essentially distinguish [12]. Essentially, there are three fundamental criteria to examine the execution of Text steganographic approach: imperceptibility, embedding hiding limit, and robustness criteria [12], section 2.4 will demonstrate these parameters.

Most of the existing text Steganography methods aren't sufficiently secure enough to stop detection and are planned to be either overcome hiding limit, or robustness problems. In this work, the imperceptibility, high hiding limit, and robustness problems have been considered to investigate the performance of the proposed text steganography approach, and the most objective is to get vital increment within the capacity of hiding data in the cover text and generate a stego keys for security improvement. To realize this objective, we proposed stenography based on a linguistic technique and a format-based technique. The linguistic technique is used to divide the cover text into embedding layers where each layer consists of sequence of words that have single part of speech detected by POS tagging, while the format-based technique is used to hide secret data in those layers by coloring letters with near RGB color..

The following sections are: Section two investigates the background, Section three discusses the proposed work, Section 4, discusses the results of the proposed work, and finally, in Section five, shows the conclusion and future works.

2. BACKGROUND

In this section, we review the fundamental concepts to play out our end goal to implement an efficient text steganography approach considering using part of speech as layer, selecting a

suitable range of RGB color by using CIELAB visual perception, and then considering natural Language processing Toolkit (NLTK), python-docx package, and Python programming language as a development environment and what are the most common criteria in designing of an efficient text steganography algorithm?.

2.1. PART OF SPEECH

Parts-of-speech (POS) are valuable in light of the huge information may give about a word and word's neighbors. Knowing the tag of the word is important to deal with probably neighbor words and concerning grammar structure around the word (the noun is typically a part of a noun phrase), that makes part-of-speech tagging a very important element of grammar parsing[13]. While there are several lists of POS, an up-to-date process on the English language are used 45-tag labeled with a Penn Treebank tag set as a large type-of corpora [14], and Table 1 shows some of these tags.

Table 1. Some Part-of-speech tags for Penn Treebank

Tag	Description	Example
CC	coordin. Conjunction	and, but,
DT	determiner	a, the
VB	verb base	form eat
VBD	verb past tense	ate
WDT	wh-determiner	which, that
NN	noun, sing	boy
WRB	wh-adverb	how, where

In this work, the cover text words are classified according to 45-tag of the Penn Treebank tag set, and the classification process may divide the cover text into 45 hiding layers or less.

2.2. NATURAL LANGUAGE PROCESSING USING PYTHON

To play out our specific end goal, we implement the proposed approach using Natural Language Processing Toolkit (NLTK), python-docx package,and Python programming language. NLTK is an arrangement of projects and libraries that gives many functionality for example, tokenization, part of speech tagging, classification, parsing, and semantic analysis for the English language. It is written in the Python programming language and was produced by Edward Loper and Steven Bird at Pennsylvania University [15], while the python-docx package is used for Microsoft document file editing, and altering RGB letters for Microsoft document file [16].

2.3. COLOR PERCEPTUAL COMPARISON

The CIELAB model (CIE, Commission International de l'Eclairage), is one LAB space models employed to measure the colors differences with relevance to human vision, where LAB expresses by L for lightness, and A, B for green-red and blue-yellow colors respectively. The main purpose is to make a linear area for the colors so that the distance between points can define in terms of the perceptual difference between colors [17].

Consider (L_1, a_1, b_1), and (L_1, a_1, b_1) are two colors in CIELAB model space, but there are different formula series for LAB application such as CIE76, CIE94, and CIE2000, and this work CIE76 formula is used for its simplicity, and for a limited used RGB range (0-15). The color difference can be defined in following [17, 18]:

$$\Delta E_{ab} = \sqrt{(L_2 - L_1)^2 + (a_2 - a_1)^2 + (b_2 - b_1)^2} \quad (1)$$

which indicates:
not perceptible if $\Delta E < 1$
unnoticeable, if $0 < \Delta E < 1$
only experienced can notice, $1 < \Delta E < 2$
only inexperienced can notice $2 < \Delta E < 3.5$
noticeable, if $3.5 < \Delta E < 5$
different colors, if $5\ \Delta E > 5$

and the formulas to convert RGB to Lab is defined as follows:

$$L = 116\left(g\left(\frac{Y}{Y_n}\right)\right), \quad a = 500\left(g\left(\frac{X}{X_n}\right)\right) - \left(g\left(\frac{Y}{Y_n}\right)\right), \quad b = 200\left(g\left(\frac{Y}{Y_n}\right)\right) - \left(g\left(\frac{Z}{Z_n}\right)\right) \quad (2)$$

$$g(t) = \begin{cases} t^{1/3} \ for\ t > 0.0082 \\ 7.83 + \frac{16.3}{117} \ for\ t \le 0.00882 \end{cases} \quad (3)$$

The proposed approach in this work depends on changing color letters of the cover text from black to a near chosen color so that the human eye cannot see any difference in that change. The RGB color space is selected within the range 0 to 15 for each parameter R, G, and B. All differences between black color and other RGB colors used in the range 0-15 were measured according to above metric CIELAB visual perception, and these differences were found less than one and within the range not perceptible by the human eye.

2.4. EVALUATION CRITERIA OF TEXT STEGANOGRAPHY

The researchers consider many criteria in designing algorithms for text steganography, and the most common criteria for recently proposed approaches concern with hiding capacity, invisibility, robustness; however, an appropriate algorithm must provide an optimal trade-off between these evaluation criteria in line with the appliance needs of approach. [7,19, 20, 21].

1. HIDING CAPACITY:

The hiding capacity can be detected by calculating the maximum embedding length of the secret message in bits that can be inserted in the cover text to construct the stego-text (4), but it is unhelpful for an algorithm to provide high capacity without having a desirable protection[19, 22, 23]. The maximum length of the secret message can be determined with respect to bit per location where location means a specific embed position in the cover text (e.g., after punctuations and space between words).

$$Maximum\ Embedding = Bit\ Per\ Location \times Total\ Locations \quad (4)$$

2. IMPERCEPTIBILITY:

Imperceptibility is the ability of a concealed message to be not seen by the human eye. This is achieved by the high level of likeness among the cover text content, and the stego text content. The source cover content needs to stand almost the same after embedding the message and should not be corrupt cover text. The most ideal method of measuring the degree of invisibility is to look at the variety of the cover text before and after inserting the secret message and the

imperceptibility may be formulated through the following mathematical expression which shows that the cover text will be approximately same after inserting the secret message [19, 25].

$$CT \approx ST, \text{ where CT is the cover text and ST is the stego text} \quad (5)$$

3. ROBUSTNESS:

The robustness of a steganography algorithm is the resistance to different steg-analysis attacks to modify, or to extract of a concealed secret message by an intruder without having authorization key. The robustness can be numerically estimated by using losing probability (LP) which indicates how much of embedded secret message will be lost from the stego text, and lower probability prompts a more robust algorithm. The distortion robustness (DR) can found by:

$$DR = 1 - LP \quad (6)$$

where

$$LP = \frac{NEL}{CTL}, 1 < NEL < CT$$

NEL =No. Of Embedding Locations, CTL=Cover Text Length

There is a safety level of security that keeps attackers from recognizing the secret message visually or from extracting it [7], which relies upon embedding capacity, invisibility, and robustness and a productive steganography approach must give appropriate balance between the three criteria [24].

3. THE PROPOSED APPROACH

The proposed approach embeds any kind of information as a string of bits in a text document. For instance, a secret text message is converted to a string of bits, and the essential hiding element is a chunk of 12 bits of the secret message, that's 12 bits embedded in the single letter by changing its color with a different RGB color close to black color. So our plan is to replace the original RGB color for each letter in the cover text to a close RGB color to embed 12 bits from a secret message.

At first, the NLTK tokenizer and tagging modules are used to divide the cover text into a sequence of words with its associated part of speech, all words with the same part of speech are collected to construct a layer, the frequencies (no. of words with the same tag) of these layers are calculated, and sorted in ascending order according to its frequencies. Each part of speech with its frequency represents an embedding layer and the capacity of the cover text is determined using the number of layers and the number of words found in each layer through the following equation:

$$No.\ of\ embeded\ bits = \sum_{i=1}^{M}\sum_{j=1}^{N} len\left(word\ (j)\right) * 12,\ \forall\ Wj\ \in Layer\ i\ ,\ i = 1..N, j = 1..M \quad (7)$$

Maximum number of embedding layers can be selected from the total number of layers, a list of non-repeated random integers from 0 to maximum layer is generated to be used in encoding or decoding process, and the layers are selected in the sequence that appears in the random layer list and a word is selected from that layer to encode or decode a part of the secret message. As long as, a binary chunk of secret message remains to hide, a word is taken from a layer until there are no more binary chunk to hide, at that point, the cover text contains the secret message and turned into a stego text, and a secret stego key is produced which is utilized to identify the integrity of

stego text at extracting process. The following two sections explain the detailed proposed algorithms for embedding and extracting process.

PROCESS. 3.1. EMBEDDING DATA

Embedding secret message algorithm
Input : Cover text file, the Secret message to hide
Output : Stego text file, key

1. The NLTK tokenize module is used to divide the cover text into a sequence of words, denoted by W = (W_1, ..., Wm)

2. The NLTK tagging module is used to assign a part of speech to each word in the sequence W and generate new sequence TW as shown below, where each word associates with its part of speech.

TW = ((W_1, T_1),..., ($W_{m,\ Tk}$)), where W_i is a word that has part of speech T_i.

3. The sequence TW of the cover text is divided into a sequence of K layers, denoted by L_1, L_2, …, L_k, where each layer contains all words with the same part of speech (POS), these layers are indexed by an integer 1, 2, ., k, where k is a total number of entire POS discovered into the cover text.

4. Sort the K layers in ascending order according to the length of layer, where

Length of i layer= Number of the words $\in$ L_i

5. Select number of embedding layers K, and generate a non-repeated random list for the layers indices, i.e SL= (7, 1, k-1, ., 4), where each layer index <= k.

6. Construct an embedding word sequence:

WS= {{WS_{11}, WS_{12}, …, WS_{1k}}, {WS_{21}, WS_{22}, .., WS_{2k} }, {WS_{31}, WS_{32}, …,WS_{3k}}…,....}

such that WS_{ij} is a word at the location i of j layer

7. Convert the secret message to binary string BS.

8. Split the bits of BS into a sequence of chunks CL= [C_1, C_2,…, C_t] such that each chunk contains 12 bits.

9. For each word W in embedding word sequence WS:
For each letter in W:
If still there is a chunk in CL:
Get next chunk C and split it into three groups of 4 bits (R, G, B)
Color the letter with (R, G, B)
Else exit

10. Produce a key from the number of embedding letters, and the selected layer list SL.

3.2. EXTRACTING DATA

Extracting Secret message Algorithm

Input : Stego text file, key
Output : Secret message

1. The NLTK tokenizer module is used to divide the cover text into a sequence of words, denoted by $W = (W_1, ..., Wm)$

2. The NLTK tagging module is used to assign a part of speech to each word in the sequence TW and generate new sequence TW that associate each word with its part of speech, denoted by $TW = ((W_1, T_1),..., (W_{m, Tk}))$, where W_i is a word that has part of speech T_i.

3. The sequence TW of the cover text is divided into a sequence of K layers, denoted by L_1, L_2,…, L_k, each layer contains all words in the cover text with the same tag, these layers are indexed by an integer 1, 2, ., k, where k is the total number of part of speech (tag) found in the cover text.

4. Sort the K layers in ascending order according to the length of layer, where

$$\text{Length of } i \text{ layer} = \text{Number of the words} \in L_i$$

5. Determine number of embedding letters EL and list layers SL from the key.

6. Construct an embedding word sequence:

$WS = \{\{WS_{11}, WS_{12}, \ldots, WS_{1k}\}, \{WS_{21}, WS_{22}, .., WS_{2k}\}, \{WS_{31}, WS_{32}, \ldots, WS_{3k}\} \ldots, ... \}$

such that WS_{ij} is a word at the location i of j layer.

7. For each word W in the embedding word sequence WS:
   For each letter in W:
   If there are more embedding letters than:
   Read the RGB color of embedding letter
   Convert RGB to a string of 12 bits and add it binary secret message BS
   Else exit

8. Convert the binary string BS to Unicode representation.

3.3. AN EXAMPLE FOR THE PROPOSED APPROACH

Consider a cover document contains the text **"Count your age by friends, not years. Count your life by smiles, not tears"**, and a secret message to embed is ‘‘**May you live every day of your life**.”
Using the algorithm in section 3.1:

1. In step 1 and step 2, the cover text is divided into a sequence of words, and a part of speech is detected for each word. The list of words with its associate part of speech is as shown in Figure 2.

```
[('Count', 'NNP'), ('your', 'PRP$'), ('age', 'NN'), ('by', 'IN'), ('friends', 'NNS'), ('not',
'RB'), ('years', 'NNS'), ('Count', 'NNP'), ('your', 'PRP$'), ('life', 'NN'), ('by', 'IN'),
('smiles', 'NNS'), (',', ','), ('not', 'RB'), ('tears', 'NNS')]
```

Figure 2. Words with its POS

2. In step 3 and step 4, a sequence of layers for the cover text is detected, each layer contains all words with the same POS then are sorted according to its length, which is shown Figure 3, there

are six layers, where the first layer is tagged by “NNS” that contains four words, second layer is tagged by “NNP” that contains two words and so on.

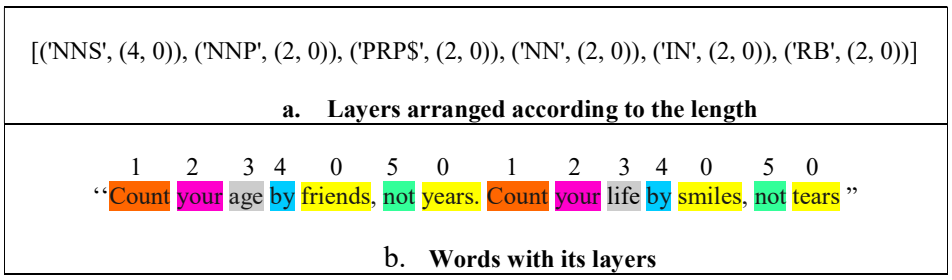

Figure 3. POS Layers and words layers

3. In step 5 and step 6, select a number of layers to be used for embedding and a non-repeated random list of is generated. Consider we select four layers for embedding, Figure 4 shows the selected layers list and the word in these layers.

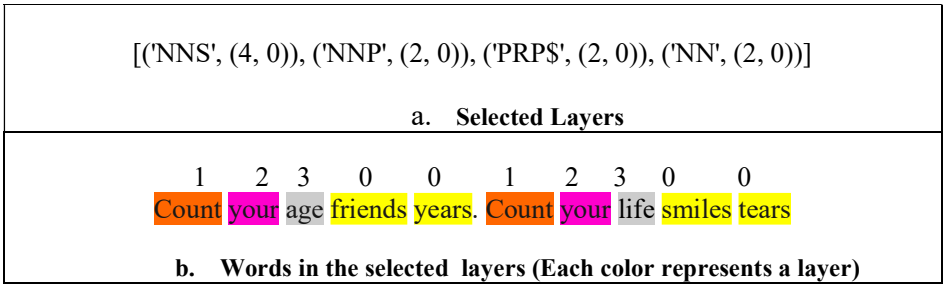

Figure 4. Selected layers and words in it

Assume the generated random list is [2, 0, 3, 1], and the words in the embedding layer will be as shows in Figure 5:

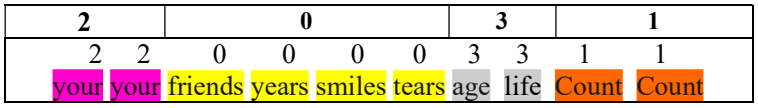

Figure 5. Words in the embedding layers

For embedding, each time a word is selected from a layer according to the order shown in Figure 5, but if the layer does not contain any word, it will move again to the next layer. The embedding word sequence appears in Figure 6.

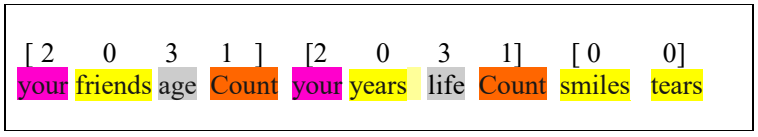

Figure 6. Embedding word sequence

4. In step 7, the secret message is converted to a binary bit form as below:

0100110101100001011110010010000001111001011011110111010100100000011011000110100101111011001100101001000000111001011011110111010101100100100000011001010111011001100101001000000110010101110110011001010111001001111001001000000110010000101111001001000000110111101100110001000000111100101101111011101010111001000100000011011000110100101100110011001100101

5. In step 8 and step 9, the binary bit string is divided into a sequence of 12 bit chunks, each chunk is divided into 4 bit groups, and the values of these groups represent R, G, and B colors respectively. The embedding word sequence, bit string selected for each word, and RGB color selected for each letter are shown in Figure 7.

Word		Word coloring
your	bits	**0100**1101**0110** 0001**0111**1001 **0010** 0000**0111** 1001**0110**1111
	RGB	[4, 13, 6] [1, 7, 9] [2, 0, 7] [9, 6, 15]
friends	bits	**0111**0101**0010** **0000**0110**1100** **0110**1001**0111** **0110**0110**0101** **0010**0000**0110** **0101**0111**0110** **0110**0101**0111**
	RGB	[7, 5, 2] [0, 6, 12] [6, 9, 7] [6, 6, 5] [2, 0, 6] [5, 7, 6] [6, 5, 7]
age	Bits	**0010**0111**1001** **0010**0000**0110** **0100**0110**0001**
	RGB	[2, 7, 9] [2, 0, 6] [4, 6, 1]
Count	Bits	**0111**1001**0010** **0000**0110**1111** **0110**0110**0010** **0000**0111**1001** **0110**1111**0111**
	RGB	[7, 9, 2] [0, 6, 15] [6, 6, 2] [0, 7, 9] [6, 15, 7]
your	Bits	**1010**1110**0010** **0010**0000**0110** **1100**0110**1001** **0110**0110**0110**
	RGB	[5, 7, 2] [2, 0, 6] [12, 6, 9] [6, 6, 6]
years	Bits	**0101**0000**0000**
	RGB	[5, 0, 0]

Figure 7. Word bit string, and RGB for letters

The following stego text will be generated after coloring each letter in the selected words to a near RGB color, that will be identical to the cover text:

'Count your age by friends, not years. Count your life by smiles, not tears "

6. In step 10, the key is generated from the number of embedding letters and the random list. For the example, the key will be 24, [2, 0, 3, 1] which means the secret message needs 24 letters to be hidden, and will be distributed into the words of the layers 2, 0, 3, and 1. Using the equation (7), the maximum message length that can be hidden in the cover text example will be 59*12=708 bits which is equal to 88 characters.

3.4. Implementation of Embedding and Extracting

The embedding and extracting processes are implemented by using Python programming language, natural language framework NLTK, and python-docx package as described below:

1. NLTK is utilized to classify each word part-of-speech in the cover text content.
2. Python-docx package is used to read the Microsoft document file, write a Microsoft document file, and change RGB letters in a Microsoft document file.

The complete implementation for hiding and extracting processes and tests are found in the link

https://drive.google.com/file/d/1-L8hmZ3AiOXILAfK2srca-k1jWDe0tUG/view?usp=sharing

4. Experimental Results

There are a lot of approaches that have similar properties to the proposed text steganography, such as Space mimic, WhiteSteg, SNOW, TWSM, wbSteg040pen, UniSpaCh, and AITSteg. But since UniSpaCh and AITSteg perform better than all others [19,26], the comparison of the

proposed approach is done on AITSteg and UniSpaCh according to the common criteria: hiding capacity, invisibility, and robustness[19]. For fair comparison, we use the same cover texts and the same secret message that is shown in the Table 2 [19] which are used to evaluate AITSteg and UniSpaCh.

Table 2. Cover Texts and Secret Massages

Tests	Secret Message	Cover Text
Test1	"Golden"	"I choose a lazy person to do a hard job."
Test2	"Bill Gates"	"I choose a lazy person to do a hard job. Because a lazy person will find an easy way to do it."
Test3	"Bill Gates Golden Words"	"Don't Compare yourself with anyone in this world…if you do so, you are insulting yourself."
Test4	"William Henry Gates III Bill Gates"	"I can understand wanting to have millions of dollars, there's a certain freedom, meaningful freedom, that comes with that. But once you get much beyond, that I have to tell you, it's the same hamburger."

The following evaluation, analysis of the proposed approach is considered:

Hiding Capacity: is a major parameter for a steganography algorithm performance the proposed approach and AITSteg approach have the same number of embedding characters for all tests as shown in Table 3. AITSteg approach inserts a ZWC character to hide 2 bits from secret message, which means that AITSteg inserts 32 ZWC characters in the cover text to hide only 8 and this will increase the cove text size unacceptably, while, the proposed algorithm can hide 12 binary in each letter of the cover text without inserting or adding any character. For UniSpaCh approach, the proposed approach is vastly improved as appeared in the Table 3.

Table 3. Comparisons with other approaches
CL= Cover Tex6t Length, SL = Secret Message Length

	Number of Embedding Characters				DR				Advantages and Limitations		
Tests	Test1 SL=6, CL=40	Test2 SL=10, CL=94	Test3 SL=23, CL=90	Test4 SL=34, CL=202	Test1 SL=6, CL=40	Test2 SL=10, CL=94	Test3 SL=23, CL=90	Test4 SL=34, CL=202	Capability	Invisibility	Robustness
proposed	6	10	23	34	90	92	84	89	High	Imperceptible	High DR>89%
AITSteg	6	10	23	34	97	98	98	99	High	Imperceptible	High DR>97%
UniSpaCh	2.2	5.25	3.5	8.5	77	78	84	83	Low	Imperceptible	Modest DR>81%

Invisibility: Figure 6 shown that the proposed approach does not alternate any cover text character after hiding the secret message since the embedding algorithm only recolors letters of the cover text to a close RGB so that the human eye cannot see any difference. All differences between the black RGB color and all other RGB colors in the range 0-15 were measured according to above metric CIELAB visual perception, and we found that these differences are less than one and within the range not perceptible by the human eye.

Robustness: the reversibility of the secret message from the stego text content is the most fundamental criteria in steganography. In this work, the measurement has been achieved by hiding the secret message in the cover text words through alerting the color of the words which are selected from multiple layers chosen randomly. The distortion robustness values of the proposed approach are calculated using the equation (7) and from Table 3, our proposed approach is more robust than UniSpaCh with a small difference than AITSteg. But AITSteg approach considered that the whole message will be embedded at one location at the beginning of the cover content, so that the first character of the message will be hidden in first the location, the next

character of the message will be hidden in the following location and so on. Which leads that the message will be hidden in one continuous stream, and its destruction is easier than the message being distributed in the cover text. Beside, AITSteg approach inserts a character for each 2 bits from a message, which leads also to destroy the embedding messages easily. As an example for that, a message for test1 has 6 characters (48 bits), needs to embed 24 characters in the cover text of length 40 characters and even it inserted in one location, it can be easily destroyed the cause of its length.

In the proposed approach, even if the hidden message is discovered, it cannot be decoded or dropped from the stego text, since the proposed hiding algorithm follows a tricky way to hide information through using not ordinary stream inserting process, a randomly multi-layer hiding technique, recoloring cover text letters with a suitable RGB color to provide imperceptibility. In the light of these evidences, it provides a good balance between embedding, imperceptibility, and robustness criteria.

5. CONCLUSIONS

This work presents a text Steganography based on word tagging and RGB coding and suggests using a word tagging technique to supply a high security, and RGB coding technique to supply high hiding capacity, and imperceptibility. This text Steganography approach groups together all words with the same POS in a layer to build multi hiding layers to ensure the security and changes the RGB color of each letter of the cover text to close RGB color to get high hiding capacity. Also, we tend to provide a text steganography approach that dynamically selects number multilayers, part-of-speech tagging, and its sequence for hiding a secret message to make the steganalysis process more difficult, besides coloring the words in multilayers tagging of the cover text letters with closest RGB is an improvement of the linguistic steganography approach that provides clearer good smart tools for data concealment

REFERENCES

[1] K. Benett, (2004), “Linguistic steganography- survey, analysis and robustness concerns for hiding information in text”, Purdue University, CERIAS Tech. Report 2004-13,

[2] S Bhattacharya, P Indu, Duta, SA Biswas, G Sanyal, (2011) “Hiding data in text through in alphabet letter patterns (CALP)”. Journal of Global Research in Computer Science, 2(3): 33-39

[3] Agarwal, M., (2013). "Text steganographic approaches: a comparison”. arXiv preprint arXiv:1302.2718.

[4] Shirali-Shahreza, M.H. and Shirali-Shahreza, M., (2006), July. “A new approach to Persian/Arabic text steganography”. In Computer and Information Science, 2006 and 2006 1st IEEE/ACIS International Workshop on Component-Based Software Engineering, Software Architecture and Reuse. ICIS-COMSAR 2006. 5th IEEE/ACIS International Conference on(pp. 310-315). IEEE.

[5] S. H. Low, N. F. MaxemchUK, J. T. Brassil, and L. O. Gorman, (1995). “Document marking and identification using both line and word shifting”, INFOCOM95 Proceedings of the Fourteenth Annual Joint Conf. Of the IEEE Computer and Communication Societies, 1995, pp. 853-860.

[6] Singh, H., Singh, P.K. and Saroha, K., (2009), February. “A survey on text based steganography”. In Proceedings of the 3rd National Conference (pp. 26-27).

[7] Taleby Ahvanooey, M., Li, Q., Shim, H.J. and Huang, Y., (2018). “A Comparative Analysis of Information Hiding Techniques for Copyright Protection of Text Documents”. Security and Communication Networks, 2018.

[8] Banerjee, I., Bhattacharyya, S. and Sanyal, G., (2011), January. “Novel text steganography through special code generation”. Int. Conf. On Systemics, Cybernetics, and Informatics (pp. 298-303).

[9] Shirali-Shahreza, M., (2008), February. “Text steganography by changing words spelling. In Advanced Communication Technology”, ICACT 2008. 10th International Conference on (Vol. 3, pp. 1912-1913). IEEE.

[10] Topkara, M., Topkara, U. and Atallah, M.J., (2006), October. "Words are not enough: sentence level natural language watermarking":. In Proceedings of the 4th ACM international workshop on Contents protection and security (pp. 37-46). ACM.

[11] Kumar, K.A. and Pabboju, S., (2018). “AN OPTIMIZED TEXT STEGANOGRAPHY APPROACH USING DIFFERENTLY SPELT ENGLISH WORDS".

[12] Krishnan, R. B., Thandra, P. K., & Baba, M. S. (2017). "An overview of text steganography. In Signal Processing, Communication and Networking" (ICSCN), 2017 Fourth International Conference on (pp. 1-6). IEEE

[13] Jurafsky, D. and Martin, J.H., (2016\4). "Speech and language processing". London: Pearson.

[14] Marcus, M.P., Marcinkiewicz, M.A. and Santorini, B., (1993). “Building a large annotated corpus of English: The Penn Treebank. Computatioal linguistics”, 19(2), pp.313-330.

[15] Hardeniya, N., Perkins, J., Chopra, D., Joshi, N. and Mathur, I., (2016). “Natural Language Processing: Python and NLTK”. Packt Publishing Ltd.

[16] http://textx.readthedocs.io/en/v1.4.x/#projects-using-textx, last visit, September, (2018).

[17] Mokrzycki, W.S. and Tatol, M., 2011. "Colour differenceΔ E-A survey". Machine Graphics and Vision, 20(4), pp.383-411.

[18] Patel, I. and Goud, J., (2012). "Colour recognition for blind and colour blind people". Int J. Eng Innovat Technol, 2(6), pp.38-42.

[19] Ahvanooey, M.T., Li, Q., Hou, J., Mazraeh, H.D. and Zhang, J., (2018). "AITSteg: An Innovative Text Steganography Technique for Hidden Transmission of Text Message via Social Media". IEEE Access.

[20] Taleby Ahvanooey, M., Dana Mazraeh, H. and Tabasi, S.H., (2016). "An innovative technique for web text watermarking" (AITW). Information Security Journal: A Global Perspective, 25(4-6), pp.191-196.

[21] Aman, M., Khan, A., Ahmad, B. and Kouser, S., (2017). "A hybrid text steganography approach utilizing Unicode space characters and zero-width character". International Journal on Information Technologies and Security, 9(1), pp.85-100.

[22] Alotaibi, R.A. and Elrefaei, L.A., (2018). "Improved capacity Arabic text watermarking methods based on open word space". Journal of King Saud University-Computer and Information Sciences, 30(2), pp.236-248.

[23] Kouser, S. and Khan, A., (2017). "a novel feature extraction approach: capacity based zero-text steganography". international journal on information technologies and security, 9(3), pp.85-98.

[24] Zhang, W., Meng, J. and Ma, C., (2018), June. “Research progress of applying digital watermarking technology for printing”. In 2018 Chinese Control And Decision Conference (CCDC) (pp. 4479-4482). IEEE.

[25] Kamaruddin, N.S., Kamsin, A., Por, L.Y. and Rahman, H., (2018). "A Review of Text Watermarking: Theory, Methods, and Applications". IEEE Access, 6, pp.8011-8028.

[26] Kumar, R., Malik, A., Singh, S., Kumar, B. and Chand, S., (2016), April. "A Space based reversible high capacity text steganography scheme using Font type and style". In Computing, Communication, and Automation (ICCCA), 2016 International Conference on (pp. 1090-1094). IEEE.