Received May 11, 2022, accepted May 30, 2022, date of publication June 13, 2022, date of current version June 23, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3182712

CSNTSteg: Color Spacing Normalization Text Steganography Model to Improve Capacity and Invisibility of Hidden Data

REEMA THABIT[1,2,3], NUR IZURA UDZIR[1], SHARIFAH MD YASIN[1,4], AZIAH ASMAWI[1], AND ADNAN ABDUL-AZIZ GUTUB[5]

[1]Department of Computer Science and Information Technology, Universiti Putra Malaysia (UPM), Serdang, Selangor 43400, Malaysia
[2]Faculty of Engineering, University of Aden, Aden, Yemen
[3]Aden Community College, Aden, Yemen
[4]Institute of Mathematical Research (INSPEM), Universiti Putra Malaysia (UPM), Serdang, Selangor 43400, Malaysia
[5]Department of Computer Engineering, College of Computer and Information Systems, Umm Al-Qura University, Makkah 24382, Saudi Arabia

Corresponding authors: Reema Thabit (rmbinthabit@gmail.com) and Nur Izura Udzir (izura@upm.edu.my)

This work was supported in part by Universiti Putra Malaysia (UPM).

ABSTRACT The rapid growth of online communication has increased the demand for secure communication. Most government entities, healthcare providers, the legal sector, financial and banking, and other industries are vulnerable to information security issues. Text steganography is one way to protect secure communication by hiding secret messages in the cover text. Hiding a large amount of secret information without raising the attacker's suspicion is the main challenge in steganography. This paper proposes the Color and Spacing Normalization stego (CSNTSteg) model to resolve the low capacity and invisibility problems in text steganography. CSNTSteg consists of two stages: the pre-embedding stage, which achieves high capacity by utilizing RGB coding and character spacing. It is designed to increase the number of bits per location and usable characters. Besides, it applies the Huffman coding technique to compress the secret message to add more capacity enhancement. The second stage is color and space normalization, which accomplishes high invisibility by normalizing the RGB coding and character spacing of the cover and stego text. CSNTSteg overcomes the color difference issue between the cover and stego texts regardless of the color of the cover text. To assess the quality of CSNTSteg, the experimental results are compared with existing works. CSNTSteg shows superior capacity over the existing studies with a percentage of 98.85%. CSNTSteg also achieves high invisibility by reducing the color difference with a percentage of 4.7% and 5.07% for black and colored cover text, respectively. Furthermore, CSNTSteg improves robustness by 94.22% by reducing the distortion in stego text. Overall, the CSNTSteg model embeds a high capacity of secret data while maintaining invisibility and security, offering a new perspective on text steganography to protect against visual and statistical attack issues.

INDEX TERMS Information and communication, data transfer, information security, steganography, text processing, RGB code.

I. INTRODUCTION

During the coronavirus pandemic, the number of Internet users in 2021 reached 4.93 billion worldwide [1]. Most communications between governments, companies, ministries, and individuals are being transformed from physical to online. Hence, secure communication is a imperative to protect sensitive information transmitted through the Internet [2]. Steganography provides secure communication by hiding sensitive information in cover media such as video, audio, image, and text [3]. Text is an ideal medium to cover communication, as text is widely utilized compared to other objects. As reported in [4], 63% of US users used the Internet to send email and message text, and 20% of them used social media sites, while 50% communicated through voice and video calls. The text steganography techniques

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Huang.

utilize text to establish secure communication on the Internet.

The efficiency of the text steganography technique is evaluated using four criteria: capacity, invisibility, security, and robustness. A good text steganography technique should provide more hiding capacity while maintaining invisibility, security, and robustness to withstand visual, statistical, and structural attacks [5]. In this paper, we focus on capacity, invisibility and robustness issues. Capacity refers to the amount of secret data hidden in the cover text [6], while invisibility means the hidden data has not caused noticeable or visible distortions that visual attacks can detect [7].

Secret information can be hidden by utilizing the structure or linguistics of the text, or by generating a new cover text. Structural text steganography mainly applies because of its high achievement in capacity and invisibility [5], [8]. One category in structural text steganography is feature-based steganography, which hides the secret information by modifying the text features such as font RGB code (color), size, style, and character spacing. Increasing the capacity of hidden data while maintaining invisibility is the main concern by the researchers [5], [9], [10], which is still a challenge in feature-based text steganography techniques still suffer from low capacity and low invisibility. The limited number of bit per embedding location (character) [5], [9], [11], [12], less usable/embeddable character [13]–[16], and uncompressing the secret message [13] are the main issues that affects the capacity in feature-based steganography; while the dissimilarity between the cover and stego text in size or appearance causes low invisibility [8], [14], [15].

Several feature-based techniques have been proposed to tackle the low capacity and low invisibility issues. Hiding the secret message using the RGB coding and forwarding email address has been proposed in [16]–[18]. However, they have low capacity, and low invisibility which resulted from generating rich color stego text that raises suspicions. Integrating the RGB coding with the part of speech (POS) is the embedding technique suggested in [11], but it can only embed 12 bits per location and fewer usable characters due to applying POS. Although, it improves the invisibility when using black cover text, it fails to achieve any improvement when the cover text is in a different color. Recently, RGB code was utilized to conceal the secret message [8], but it suffers from low capacity by hiding only eight bits per location. Further, only the English alphabets are utilized as cover characters. Although the invisibility is improved when the cover text is black, it fails with other color texts.

The main contribution of this paper is the Color and Spacing Normalization stego (CSNTSteg) model that increases the capacity while maintaining invisibility. The CSNTSteg model increases the capacity and improves invisibility and robustness by utilizing RGB coding and character spacing. The CSNTSteg model has been designed in two stages; pre-embedding and normalization, and the contributions are outlined as follows:

1. In the pre-embedding stage, the capacity is increased by reducing the secret message's length using Huffman coding; the number of bits per location is increased to carry 16 bits by combining the RGB coding and character spacing; and the number of usable characters is increased by utilizing the font attributes.
2. In the normalization stage, the invisibility is improved against visual attacks by reducing the color difference between the cover and stego texts so as not to raise suspicions.
3. The experiments are conducted in two types of cover texts: black and colored, to demonstrate the strength of CSNTSteg that is applicable in any text color while maintaining invisibility.

The rest of this paper is organized as follows: Section 2 gives some background information, provides the critical analysis of feature-based techniques, and introduces the evaluation criteria in the text steganography of capacity, invisibility, and robustness. Section 3 describes the CSNTSteg model in detail, while Section 4 presents the experimental results and the comparison with the existing studies. Finally, the paper is concluded with future work.

II. RELATED STUDIES

The survey of existing studies in text steganography is presented in this section. It starts with a brief description of information hiding and then gives a critical analysis of existing studies of feature-based text steganography. Finally, it explains the text steganography evaluation criteria, focusing on capacity and invisibility.

A. INFORMATION HIDING

Information hiding refers to hiding secret messages in a digital medium such as video, audio, image, and text. [19]–[22]. Watermarking and steganography are considered to be information hiding techniques [23]. Watermarking is the method of hiding a single or dual watermark in a cover text in the form of a tag, label, or digital signal. It is commonly used to determine who owns the copyright to a signal [24], [25]. Steganography is a younger security method than cryptography, which could be defined as the art and science of hiding secret messages in cover objects without raising attackers' suspicion. The cover object could be an image, audio, video, text, etc. [26], [27]. In image steganography, transform domains are the most well-known methods and have been integrated with a chaotic map algorithm to improve security [28]–[30]. Audio steganography hides the secret message into audio material based on the elusion of the human auditory system [31]. Video steganography is a simple extension of image steganography. Combining a chaotic map algorithm with video steganography enhances security and robustness [32]. Steganography is considered broken when the existence of the secret message is detected and extracted.

B. TEXT STEGANOGRAPHY

Text steganography is centered on the embedding method used to conceal the sensitive information in the cover text, which can be divided into three categories: coverless, linguistic, and structural [33].

Coverless steganography (or random and statistics) techniques use the statistical characteristics of the secret message to produce the cover text automatically. It does not require pre-existing cover text and instead relies on the structures and features of a given language, such as the past verb format, how to produce sentences, and so on [13], [16], [34]–[37].

Linguistic steganography hides secret message through linguistic elements such as word usage [38]–[42]. It utilizes the language properties to hide the secret message as in Arabic text steganography techniques [43]–[46].

Structural steganography changes physical document formatting to cover secret information. Deliberate misspellings, font resizing, and space injection, among others, are examples of format-based techniques used in text steganography. Structural steganography can be classified into three main classes, white-space-based, zero-width-based, and feature-based. Zero-width-based inserts the Unicode characters like Zero With Non-Join (ZWNJ), Zero With Join (ZWJ), and Zero Width Character (ZWC) that have different behavior in different scripts to hide the secret bits [15], [47], [48]. Invisibility is improved by using these Unicode characters because they are invisible to readers [7]. However, inserting Unicode characters led to an increase in the stego text size, which can be detected by most word editors. Moreover, most Unicode text steganography techniques suffer from low capacity [10].

White-space-based is used to hide information in the text by inserting extra space between words and at the end of a line or a paragraph [49]–[54].

C. FEATURE-BASED TEXT STEGANOGRAPHY

To conceal the secret message, the feature/format-based modifies some cover text features such as font size, style, RGB code, etc., to hide the secret message [55]. According to [51], [56], [57] most feature-based techniques cannot resist formatting conversion. However, they present excellent improvements in capacity and invisibility [8], [14], [17], [58], [59].

1) RGB CODE TECHNIQUES

Employing the forward mail platform and the RGB code of cover text to conceal the secret message is introduced in [16]. Some secret bits are hidden by changing the color of the cover text using a color mapping table. At the same time, the remaining secret bits are converted to symbols using the symbol table to write in the forward mail. The secret message is compressed by Huffman coding; the color mapping table and the symbol table, including the boundary color and the filling color, are used to hide the secret message. Later, work in [17] has improved that of [16] by applying the Lempel–Ziv–Welch (LZW) rather than Huffman coding to compress the secret message. Moreover, two sizes of a secret block have been determined in the improved technique and its extension, 1 and 5 bits per color change.

Changing the color of invisible characters such as space, newline, etc., is the proposed hiding technique in [14]. Similar to [17], the presented technique in [18] embeds the secret message. It applied the permutation algorithm rather than the LWZ. Integrating the linguistic approach and the feature-based method to hide the secret message is suggested in [11]. The linguistic technique splits the cover text into embedding layers. Each layer comprises a sequence of words with a single part of speech (POS) recognized by the POS tagger. The cover words are selected based on the stego key. The format-based technique recolors the cover letters with a near RGB color code of black to conceal 12 secret bits per letter.

Recently, the proposed technique in [8] embeds a secret message into cover text using the RGB color in a random location. Each secret character will be converted into the American Standard Code for Information Interchange (ASCII) and then converted to (x, y, z) or 3D representation using their proposed Second Quotient Remainder Theorem (SQRT) to map with RGB color. Besides, the RGB code has been utilized in image steganography, especially when the true color image has been used as a cover [60]–[62].

2) CHARACTER SPACING TECHNIQUES

The frequency normalization technique and character string mapping for text steganography to uniform the occurrence of the English letter in the cover text with the character spacing is the embedding technique suggested in (Ramakrishnan *et al.*, 2016). Character string mapping creates the seven stego characters, with seven different ways to hide each character. This strategy, however, necessitates a cover text with a maximum of four characters to conceal one hidden character. Later, the improvement of [58] was introduced in [63] by creating the eight stego characters, with eight different ways to hide each character.

3) FONT STYLE TECHNIQUES

Creating a new font style that transforms dots or letters horizontally or vertically is the first idea of embedding proposed in [64]. Characters were shifting by a few degrees as 1/300 in. up or down the lines of text were made. Hence, different distinctive shapes of the text are used to hide information.

Shifting matra of Hindi alphabets has been utilized to hide the secret message in [65]. The specific matra has been shifted toward the left or right by creating a new font style applied to the embedding character of the cover text.

Later, modifying the font style of the white space characters to hide a secret message has been suggested in [66].

4) FONT SIZE TECHNIQUES

The researcher in [67] proposed a technique for text steganography based on the font size of space character in Microsoft Word. The font size of the invisible space character is slightly

TABLE 1. Critical summary of feature-based techniques.

Author	*Technique*	Capacity			Invisibility		Applicability	
		Bit/ location	**Usable character**	**Compression**	**Identical in size**	**Identical in appearance**	**Language**	**Font color**
[8]	RGB coding	8 bits / char	English Alphabets	No	Yes	Yes	English	Black
[18]	RGB coding	21 bits /10 char	All symbols except the normal space	No	Yes	No	Any	Rich color
[63]	Character spacing	8 bits /char	English Alphabets, normal space and dot.	No	Yes	Yes	English	Any
[6]	Font size changing		invisible space character	Huffman coding	Yes	Yes	English	Any
[11]	RGB coding	12 bits /char	English Alphabets	No	Yes	Yes	English	Black
[17]	RGB coding	5 bits /char	All symbols except the normal space	LZW	Yes	No	Any	Rich color
[16]	RGB coding	1 bit /char	All symbols except the normal space	Huffman coding	Yes	No	Any	Rich color
[58]	Character spacing	4 bits /char	English Alphabets, normal space and dot.	No	Yes	Yes	English	Any
[66]	Create new font style	1 bit / space	white space	No	Yes	Yes	Hindi	Any
[67]	Font size changing		invisible space character	No	Yes	Yes	English	Any
[65]	shifting matra	1 bit / shifting	Specific matra	No	Yes	Yes	Hindi	Any
[68]	Change tracking	8 bits /change	spelling mistakes and typos	No	No	No	English	Any
[64]	Shifting dots	1 bit / shifting	Dotted letters	No	Yes	Yes	English	Any

changed from the standard font size, and hence, the secret message bit is hidden.

5) CHANGE TRACKING TECHNIQUES

Utilizing the document in the regeneration step using change tracking is the hiding technique proposed in [68]. In the regeneration step, the document addresses common spelling mistakes and typos. It could be included with synonym replacements, which are used to conceal the secret bits.

Redesigning the Huffman code generator instead of using the original Huffman code generator is introduced in [6]. It generates Huffman code using non-occurrence probability instead of occurrence probability to generate common substitution. Then, it hides secret message into synonyms by using track changing.

6) RESEARCH GAP ON THE FEATURE-BASED TECHNIQUES

This section presents the critical summary of the feature-based techniques in Table 1. It analyzes the main points that affect the capacity, invisibility, and applicability of feature-based techniques. Compression, the number of usable characters, and the number of bits per location can increase/decrease the embedding capacity. More identicality in appearance and size improves the invisibility of hidden information.

From Table 1, it is evident that most RGB code techniques hide more bits per location than others. However, compression techniques are less commonly employed to improve the embedding capacity.

Besides, the RGB coding techniques do not affect the file size by inserting characters to hide the secret bits. Hence, the RGB coding techniques enhance invisibility regarding identicality in cover and stego text size.

However, most RGB coding techniques suffer from low invisibility when the appearance of both texts, cover, and stego, is not identical. The appearance of non-identicality is caused mainly by generating the stego text with rich color, which raises the eavesdropper's suspicion.

The RGB coding techniques have limited applicability because they are designed for specific languages and specific colored ext. This study finds that most RGB coding techniques are applied to English texts only, which means that institutions speaking languages that do not use Latin letters do not benefit from them. In addition, it uses only black text, which helps the eavesdropper reduce the circle of eavesdropping only on the black text that travels through the public channel. Therefore, this paper tackles the previous issues by proposing a new text steganography model that utilizes RGB coding and character spacing. The CSNTSteg model achieves superiority in capacity and invisibility compared to existing techniques.

D. TEXT STEGANOGRAPHY EVALUATION CRITERIA

Four evaluation criteria must be considered when a researcher develops text steganography [7], i.e., capacity, invisibility, robustness, and security, where some of these criteria can be evaluated by calculation while others can be visualized [9]. High embedding performance is achieved by making a trade-off between these criteria [5], [7]. Most text steganography techniques focus on increasing the embedding capacity. It is worth noting that high embedding capacity affects the invisibility of the stego text. This consequently affects the text steganography technique's security, especially when it depends on invisibility and robustness properties.

In this paper, we discussed capacity, invisibility, and robustness in-depth, as well as a brief introduction to security. The evaluation criteria are interconnected; for example, an increases in capacity cause low invisibility. Hence, low invisibility leads to detecting the presence of the hidden message, making it easy for the attacker to extract the hidden message when security is low. Although robustness is the highest priority in watermarking [69], it is also important in steganography [44]. When the attacker fails to extract the hidden message, he may attempt to destroy it by tampering.

1) CAPACITY

Capacity is the main priority that concerns researchers in text steganography, and it refers to the quantity of embedded data in the cover text [5], [17], [49]. According to [13], a good embedding algorithm is an algorithm that can hide the secret message with a size that is larger than the size of the cover text. Capacity in text steganography can be increased in several ways:

- Raising the number of embeddable/usable characters/positions in the cover text [11], [58].
- Expand the secret block length by increasing the number of bits per location [9], [11], [49].
- Reducing the size of the secret message by using techniques [13], [17], [70]–[72].
- Combining more than one text feature [12], [53], [73].

The capacity in text steganography can be measured numerically by the following metrics:

a: CAPACITY RATIO

It evaluates the maximum size that can be hidden in the cover text. It is a common parameter used to measure the capacity ratio of hidden bits relating to the cover text using the following formula [13], [56], [71]:

$$Capacity\ Ratio = \frac{Amount\ of\ hidden\ bits}{Size\ of\ Cover\ Text\ in\ bits} \times 100 \quad (1)$$

b: BIT PER LOCATIONS

The hidden bits can be embedded in characters, words, sentences, or lines based on the embedded algorithm. In this study, the embedding location indicates a character hiding the secret bits by utilizing RGB coding and character spacing features. This measurement can be calculated numerically by applying the following formula [5], [11]:

$$\begin{aligned}&Bits\ per\ location\\&\quad = Total\ number\ of\ hidden\ bits\ /\ hidden\ locations \quad (2)\end{aligned}$$

c: MAXIMUM CAPACITY

The cover text has a limit to how many secret bits it can be embedded, and this limit can be evaluated using the maximum capacity metric. It computes the expected total hidden bits and calculates them using the following formula [5], [7], [11], [25]:

$$\begin{aligned}Max.\ Capacity &= bits\ per\ location\\&\quad \times total\ number\ of\ cover\ characters \quad (3)\end{aligned}$$

d: USAGE RATIO

The effectiveness of text steganography is evaluated by hiding the secret bits in less space. This can be measured by computing the number of used characters in the cover text. The following formula is used to calculate the usage ratio [10], [74]–[76]:

$$Usage\ Ratio = \frac{Total\ of\ Used\ Characters}{Total\ Caracters\ of\ Cover\ Text} \times 100 \quad (4)$$

2) INVISIBILITY

Text steganography conceals the secret message in the cover text without causing perceptible or visible distortions detected by a visual attack [77]–[79]. Some researchers have not stated invisibility as the essential criterion of steganography evaluation [56]. On the other hand, most researchers have highlighted invisibility as one of the most critical goals to protect the hidden message [5], [7], [80]–[88]. We believe that invisibility is one of the most critical features for preventing the attacker from detecting the hidden message then extracting it.

One way to detect the hidden message is to check the similarity between the cover and the stego file. The similarity depends on the type of cover and the utilized features. Text steganography algorithms modify the text or the cover text features to generate the stego text. Hence, the difference between the cover and stego text can be numerically measured by two metrics:

a: JARO–WINKLER DISTANCE

It calculates the text differences by checking the size, semantics, and lexical similarities of the texts [89]–[92]. Therefore, it is an accurate metric to evaluate the embedding techniques that replace/insert/delete the word/letter/character from the cover text to hide the secret information. However, the Jaro–Winkler distance has limitations in font attribute techniques.

b: COMMISSION INTERNATIONAL DE L'ECLAIRAGE LAB (CIELAB) MODEL

The invisibility of the text steganography techniques that utilize RGB code can be evaluated by measuring the color differences between the cover and stego text [11]. The LAB space models measure the color differences relevant to human vision. LAB refers to L for lightness and A and B for green-red and blue-yellow colors. The primary purpose is to make a linear area for the colors so that the distance between points can define the perceptual difference between colors. There are different formula series for LAB applications such as CIE76, CIE94, and CIE2000, and in this work the CIE76 formula is employed for its simplicity. The color difference can be defined by the following formula [93], [94]:

$$\Delta E_{ab} = \sqrt{(L_2 - L_1)^2 + (a_2 - a_1)^2 + (b_2 - b_1)^2} \quad (5)$$

The indication of Delta-E is shown in Table 2.

TABLE 2. ΔE indication.

Value	*Indicates*
ΔE <1	not perceptible
0 < ΔE < 1	unnoticeable
1 < ΔE < 2	only experienced can notice
2 < ΔE < 3.5	only inexperienced can notice
3.5 < ΔE < 5	noticeable
5 ΔE> 5	different colors

The formulas to convert RGB to LAB is defined as follows:

$$L = 116\left(g\left(\frac{Y}{Y_n}\right)\right), a = 500\left(g\left(\frac{x}{x_n}\right)\right) - \left(g\left(\frac{Y}{Y_n}\right)\right)$$

$$b = 200\left(g\left(\frac{Y}{Y_n}\right)\right) - \left(g\left(\frac{z}{Z_n}\right)\right)$$

$$g(t)\begin{pmatrix} t^{1/3}\, for\ t > 0.0082 \\ 7.83 + \frac{16.3}{117}\, for\ t \leq 0.00882 \end{pmatrix}$$

3) ROBUSTNESS

Robustness in steganography is the ability of the hidden data to withstand attacks, such as rotation, cropping, added noise, and compression [95]. The robustness of text steganography techniques could be quantified statistically using the Losing Probability (LP) [5], [7]. It concludes that a portion of the hidden secret bits that have been lost from the cover text is considered to be LP. The steganography technique becomes more resistant as the LP decreases. Assume the number of embedding locations in the cover text is NL, and the total length of the cover text is TC. The Distortion Robustness (DR) can then be determined using the following formula:

$$DR = [1 - LP]$$

where $1 < NL < TC, NC \ \in N, TC \ \in N \ LP = \frac{NL}{TC}$

4) SECURITY

The steganography system is considered practically secure if no attacks detect, destroy, or extract the hidden message [7], [56], [96].

III. PROPOSED METHODOLOGY

In this work, the Color and Spacing Normalization Text Steganography (CSNTSteg) model is proposed to hide the secret message in the cover text. CSNTSteg embeds a long length of the secret message while keeping its invisibility high. The CSNTSteg model utilizes RGB coding, character spacing, and Huffman coding to increase hiding capacity. CSNTSteg utilizes all letters, numbers, and symbols as cover characters except the white-space characters such as space, tab, line feed (newline), carriage return, form feed, and vertical tab because this will increase the size of the stego text, which may raise suspicions. At the same time, high invisibility is achieved by applying the normalization between the cover and stego text to reduce the color differences that raise suspicion.

The CSNTSteg model includes two main procedures: embedding and extraction. The embedding procedure hides the secret message in the cover text on the sender side. Then the sender transmits the stego text to the receiver through a public channel. On the receiver side, the recipient gets the stego text and then applies the extraction procedure to retrieve the secret message from the stego text.

A. EMBEDDING PROCEDURE

The embedding procedure consists of two main stages: the pre-embedding and the color and spacing normalization. The secret message and the cover text are prepared for embedding in the pre-embedding stage using pre-shared arrays. Then, the normalization stage embeds the secret message by modifying the RGB code and character space with low visible distortion.

1) PRE-EMBEDDING STAGE

In this stage, the secret message and cover text are framed before embedding. The secret message size is compressed and split into secret blocks. The secret block size is enlarged to hide 16 secret bits per embedding location. Hence, this stage increases the capacity of CSNTSteg from several aspects. Later, the RGB code and character space are extracted from the cover text and prepared for the next stage using pre-shared arrays. So, the pre-embedding stage creates six components: compressed secret message, Secret Blocks (SB) matrix, Color & Spacing of cover text (CSct) matrix, 4bit array, color array, and spacing array.

a: COMPRESSED SECRET MESSAGE

Various existing studies have determined the significance of compression techniques to enhance the embedding capacity [1]–[6]. Compressing the secret message reduces its length, hence increasing the hiding capacity. Current works [7]–[13] show that the Huffman coding is a suitable compression technique for a text. Accordingly, it is applied in this study to decrease the size of the secret message. Then, the compressed secret message is generated as a bitstream.

b: SECRET BLOCKS (SB) MATRIX

The size of the secret block indicates the number of hidden bits per location [14]–[16]. There are various sizes of secret blocks; Table 3 presents the bit per location in the current text steganography techniques.

TABLE 3. Bit per location of the existing techniques.

Author	*Bit/Location*
[17]	1 and 5 bits / char
[5]	2 bits / 1 char (ZWJ)
[9]	2 bits / 1 char (ZWC/Unicode space)
[15]	2 bits/char
[51]	4 bits / HL
[16]	1 bit / char
[6]	3.2 bits/Synonym
[13]	8 bits1/ 1 char
[58]	8 bits/char
[50]	3 bits / line
[8]	8 bits / char
[11]	12 bits / char
[18]	21 bits /10 char

From Table 3, the maximum length of the secret block reaches 12 bits, as in [11]. According to [5], [9], when the secret block size is enlarged, the embedding capacity is increased. Thus, the secret block was constructed to carry 16 secret bits in this work. The bitstream is divided into secret blocks of 16-bit. Then, each secret block is split into four parts of 4-bit. The four bits have 16 cases ranging from 0000 to 1111. The 16 cases are the accepted limit in this study, which was tested to meet the high invisibility of RGB code and character space modification. The secret blocks are collected in the Secret Block (SB) matrix. The total length of the secret block (sbl) is indicated by the number of records. In case the last secret block is shorter than 16 bits, the blanks are filled with zeros. In contrast, the number of columns is four.

c: COLOR AND SPACING OF COVER TEXT (CSCT) MATRIX

The CSct is a matrix consisting of sbl records and four columns. It holds the extracted RGB codding and spacing of the cover text. The first three columns in CSct store the RGB color code (Red, Green, Blue), while the last column is filled with zeros as the default character spacing.

d: PRE-SHARED ARRAYS

Three one-dimensional arrays are created for all embeddings in the sender and then shared with the receiver for all extractions. The items on the pre-shared arrays are reordered randomly for the first time to add more complexity.

The first array is 4bit, which indicates the 16 items containing the binary numbers from 0000 to 1111. The index in the 4bit array pointed to the item's location that will be used to map the secret blocks with color and spacing values. The second array is a color array that contains 16 color values ranging from 1 to 16. The color values refer to the amount which will be added/subtracted to/from the cover RGB code to the closest stego RGB code. More than 16 makes the changes in the stego RGB code increase the color differences between the cover and stego text, which raises suspicion. Color arrays are used to embed the first three 4-bit of the secret block. The third array is the spacing array, which has 16 items that are used to conceal the last 4-bit of the secret block. The concealing is done by resetting the stego character spacing from the default value (zero) to 16 possible values. In this study, two spacing properties are used; expanding and condensing. Expanding increases the space between letters, while condensing decreases the space between letters. The spacing values start from 0.1 until 0.8, expanding and condensing in both cases.

2) NORMALIZATION STAGE

This stage reduces the differences between the cover and stego text, improving the invisibility of the hidden message's existence against the visual attack. The normalization stage produces the stego RGB code and stego character spacing by altering the cover text's RGB code and character spacing. The altering causes less distortion on stego text and makes it similar to cover text because the altering is done with the closest RGB code to cover text. The CSNTSteg model is superior to the existing works by applying the normalization that makes the embedding works on any text color with high invisibility. The normalization applies to the RGB code (color) and the character spacing as below:

a: COLOR NORMALIZATION

The main aim of this process is to the color differences between the cover and stego text, which is an excellent solution to overcome the low invisibility issue. Each 4-bit in the secret block matrix has a predefined color value in the color array. The predefined color value for the specific 4-bit is returned by finding the 4-bit value in the 4bit array and then returning its index. Next, the returned index points to the particular value in the color array. Hence, the predefined color value is returned and saved in the Color & Saving Mapping (CSM) matrix. The CSM matrix has the same number of columns and records as the SB matrix.

Later, the color values will be used to generate a stego RGB code from the cover RGB code by the altering process. There are two altering processes in color normalization: subtracting and addition. The color consists of three items in the RGB code: Red, Green, and Blue, each in the range of 0 to 255. This technique subtracts values in the range from 16 to 1 if Red, Green, or Blue is near to the maximum number (255).

TABLE 4. Applying color normalization on the basic RGB codes.

Cover color	Name	CSct	CSM	CSst	Stego color
■	Black	(0,0,0)	(13,1,10)	(13,1,10)	■
□	White	(255,255,255)	(13,1,10)	(242,254,240)	□
■	Red	(255,0,0)	(13,1,10)	(242,1,10)	■
■	Lime	(0,255,0)	(13,1,10)	(13,254,10)	■
■	Blue	(0,0,255)	(13,1,10)	(13,1,240)	■
■	Yellow	(255,255,0)	(13,1,10)	(242,254,10)	■
■	Aqua	(0,255,255)	(13,1,10)	(13,254,245)	■
■	Fuchsia	(255,0,255)	(13,1,10)	(222,1,245)	■
■	Silver	(192,192,192)	(13,1,10)	(205,191,202)	■
■	Gray	(128,128,128)	(13,1,10)	(141,129,138)	■
■	Maroon	(128,0,0)	(13,1,10)	(141,1,10)	■
■	Olive	(128,128,0)	(13,1,10)	(141,129,10)	■
■	Green	(0,128,0)	(13,1,10)	(13,129,10)	■
■	Purple	(128,0,128)	(13,1,10)	(141,1,138)	■
■	Teal	(0,128,128)	(13,1,10)	(13,129,138)	■
■	navy	(0,0,128)	(13,1,10)	(13,1,138)	■

In reverse, the technique adds values ranging from 16 to 1 if Red, Green, or Blue's value is near the minimum number (zero). Finally, all the stego RGB codes are collected into the Color Spacing Stego (CSst) matrix. To illustrate the color normalization, the basic RGB codes are used as examples when the CSM's record is equal to (13,1,10), as shown in Table 4. It shows that the cover and stego colors look the same, achieving high invisibility and applying to any text color.

b: CHARACTER SPACING NORMALIZATION

To accomplish high invisibility, the closest character spacing values to zero (default) are determined when modifying the character spacing of the cover text. Two properties of character spacing are employed; condense and expand, ranging from 0.2 to 0.8. To demonstrate the spacing normalization, the letter "a" in the word "spacing" is presented as an example in Table 5.

TABLE 5. Applying spacing normalization on the character "a."

Spacing value	*Expanded*	*Condensed*
0.1	spacing	spacing
0.2	spacing	spacing
0.3	spacing	spacing
0.4	spacing	spacing
0.5	spacing	spacing
0.6	spacing	spacing
0.7	spacing	spacing
0.8	spacing	spacing

To produce stego character spacing, the last column in the CSM matrix is added to the last column in the CSst matrix. It is evident from Table 5 that the stego character spacing values modified the letter "a" with high invisibility. The color normalization and the character normalization are combined to hide the secret block, as given in the embedding algorithm (Algorithm 1).

The first three columns in the SCct and CSM matrix work to produce the stego RGB code. While the end column in the matrices is utilized to create the stego character spacing.

The stego RGB code and stego character space generation are repeated until the latest CSct and CSM matrices' record (i = sbl). All the generated records are collected into the Color Spacing Stego (CSst) matrix.

In the final step of embedding, the number of selected characters from the cover text equals the sbl.

The embedding is sequential, which means the first secret block is embedded in the first character of the cover text and the same for the rest. The stego text is generated by altering the two font features of the cover text. Each record in the CSst matrix adjusts the RGB code and character space of one cover character to hide one secret block. Then the stego text is transmitted from sender to receiver via a public channel with high invisibility against visual attacks. The embedding algorithm is given in Algorithm 1, and Fig. 1 demonstrates the data flow of the embedding procedure.

B. EXTRACTION PROCEDURE

Two stages are passed to extract the hidden message on the receiver side: color and spacing extraction and secret block retrieval. During the extraction, the stego text is verified for tampering attacks. Verification demonstrates if the stego text has tampered through the transmission or not.

Algorithm 1 Embedding of the CSNTSteg Model.

```
Input: Secret message (SM) and cover text (CT), 4bit array (4bit), color array (Cr), and spacing array (Sg).
Output: Stego text (ST).
1. Start
----------Start the pre-embedding stage ----------
2. Bitstream ← Compress SM using Huffman coding.
3. Create SB matrix:
4.   SB matrix ← Split the bitstream into blocks of 16-bit.
5.   sbl ← Count the records of SB matrix.
6. Create the pre-shared arrays: 4bit, Cr & Sg.
7.   4bit array ← Fill the array with binary numbers
       ranging from 0000 to 1111 randomly ordered.
8.   Cr array ← Fill the array with random numbers
       ranging from 1 to 16.
9.   Sg array ← Fill the array with random numbers
       ranging from (+/_ 0.1) to (+/-0.8).
10. Create CSct matrix.
11.   CSct matrix ← Extract the RGB code and
       character space of CT.
------------Start the normalization stage ----------
12. j, i & n = 0.
13. Create CSst matrix:
14.   For i ≠ sbl ? do
15.     For j ≤ 2 ? do
16.     If 255 ≥ CSct[i][j] ≥ 239 ? then
17.       CSst[i][j] ← CSct[i][j] - CSM[i][j]
18.     Else
19.       CSst[i][j] ← CSct[i][j] + CSM[i][j]
20.     End if
21.     j = j +1
22.     End for
23.     End for
24.   If j = 4? then
25.   i = i + 1
26.   End if
27. repeat step (14 to 26).
28. For n = 0 to sbl do
29.   ST ← Modified the RGB code and space of CT
       at n by using the CSst matrix.
30. End for
31. n = n + 1
32. Return the ST.
33. End
```

Algorithm 2 Extraction of CSNTSteg Model.

```
Input: Cover text (CT), stego text (ST), 4bit array (4bit), color array (Cr) and spacing array (Sg)
Output: Secret message (SM)
1. Start
---Start secret block retrieval stage ---
2. Call the pre-shared arrays: 4bit, Cr and Sg.
3.   Create the matrices CSct, CSst, CSM:
4.   CSct ← extracted color code &
5.   space of CT.
6.   CSst ← extracted color code & space of ST.
7.   CSM ← CSct - CSst
8. sbl ← calculate records number of CSM matrix.
9. i, j =0
10. For i= 0 to sbl do
11.   If ( CSM[i][0] and CSM[i][1] and CSM[i][2]
      and CSM[i][3]) = 0 then
12.     Remove CSM[i][j] (Unused characters).
13.   End If
14. If ( CSM[i][0] or CSM[i][1] or CSM[i][2] ) Not
    in Cr or ( CSM[i][3] Not in Sg) then
15.   Display "Fail extraction because stego text has
      been tampered."
16. End If
-Start the color and spacing extraction stage-
17. Sbl ← Calculate the number of records after removing
    all unused records (unused characters).
18. For i = 0 to sbl do
19.   For j = 0 to 2 do
20.     Find CSM[i][j] in Cr.
21.     Return index of Cr item.
22.     SB[i][j] ← Return the item from 4bit that
      has the Cr index.
23.   j = j + 1
24.   Find CSM[i][j] in Sg.
25.   Return index of Sg item
26.   SB[i][j] ← Return the item from 4bit that
      has the Sg index.
27. i = i + 1
28. Bitstream ← Convert SB matrix to bitstream.
29. SM ← Decompress the bitstream using Huffman code.
30. Return SM
31. End
```

1) COLOR AND SPACING EXTRACTION STAGE

On the receiver side, the stego and cover text are uploaded to retrieve the hidden message by extracting the RGB code and character space of both texts. The extracted features of stego and cover text are saved in the two matrices, CSst and CSst. Then, the differences between the two matrices are calculated and held in the CSM matrix. When the record of the CSM matrix is filled with zeros, it indicates unused characters which are removed from the matrix. The stego text is verified from the tampering attack by checking that each item in the CSM matrix exists in the color or spacing arrays. The tampered stego text led to a failed extraction, while the untampered text led to a successful extraction. The output of this stage is the CSM matrix, which will be employed in the following stage to recover the secret block matrix. The total records of the CSM matrix is equal to the sbl.

2) SECRET BLOCK RETRIEVAL STAGE

To recover the SB matrix from the CSM matrix, the two arrays, color and spacing, are utilized as below:

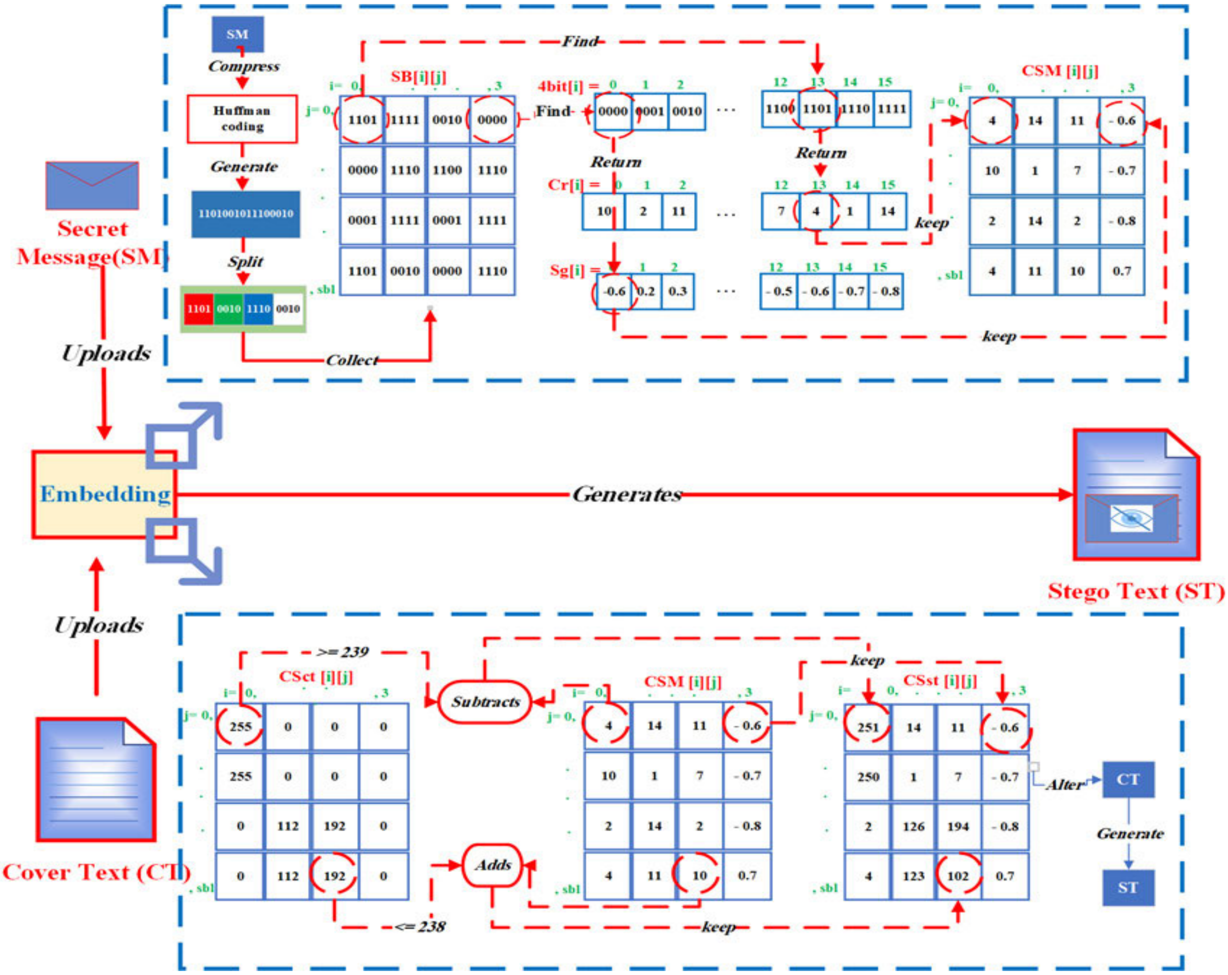


FIGURE 1. Data flow of embedding procedure.

- The first three members of the CSM matrix will find their values in the color array. Then, its index in the color array will be returned. This index points to the four secret bits retained in the 4bit array.
- The last member of the CSM matrix will locate its values in the spacing array. Then, its index in the spacing array will be returned. This index points to the four secret bits retained in the 4bit arrays.

The process will be repeated until the end (sbl) of the CSM matrix. Finally, the secret block matrix turns into a single bitstream, and then decomposes using the Huffman coding technique to obtain the secret message. The extraction algorithm is presented in Algorithm 2, and Fig. 2 illustrates the data flow of the extraction procedure.

IV. EXPERIMENTAL RESULTS AND COMPARISONS

To evaluate the efficiency of the CSNTSteg model, CSNTSteg has been implemented with the C# programming language using the Integrated Development Environment (IDE) to develop the system. Moreover, the experimental results have been compared with the existing works regarding capacity and invisibility. The secret message and cover text dataset in this study have been utilized in previous studies [8], [16], [17], [71], [74], [97], [98]. The selected secret message has 198 characters, and the cover text has 874 characters [18], as presented in Table 6. The original text of the cover is black and colored text generated randomly to show how the CSNTSteg model is applicable in both texts.

A. CAPACITY

Most researchers discuss capacity results from one perspective based on their concerning issue. In this study, the capacity has improved by considering three issues: the low number of bits per location; the low efficiency of compression technique; and the less usable/embeddable characters. Hence, the results from capacity experiments are calculated by four metrics: capacity ratio, bit per location, maximum capacity, and usage ratio.

1) CAPACITY RATIO COMPARISON

The capacity ratio indicates the maximum amount that can be hidden in the cover text. It is measured by applying the

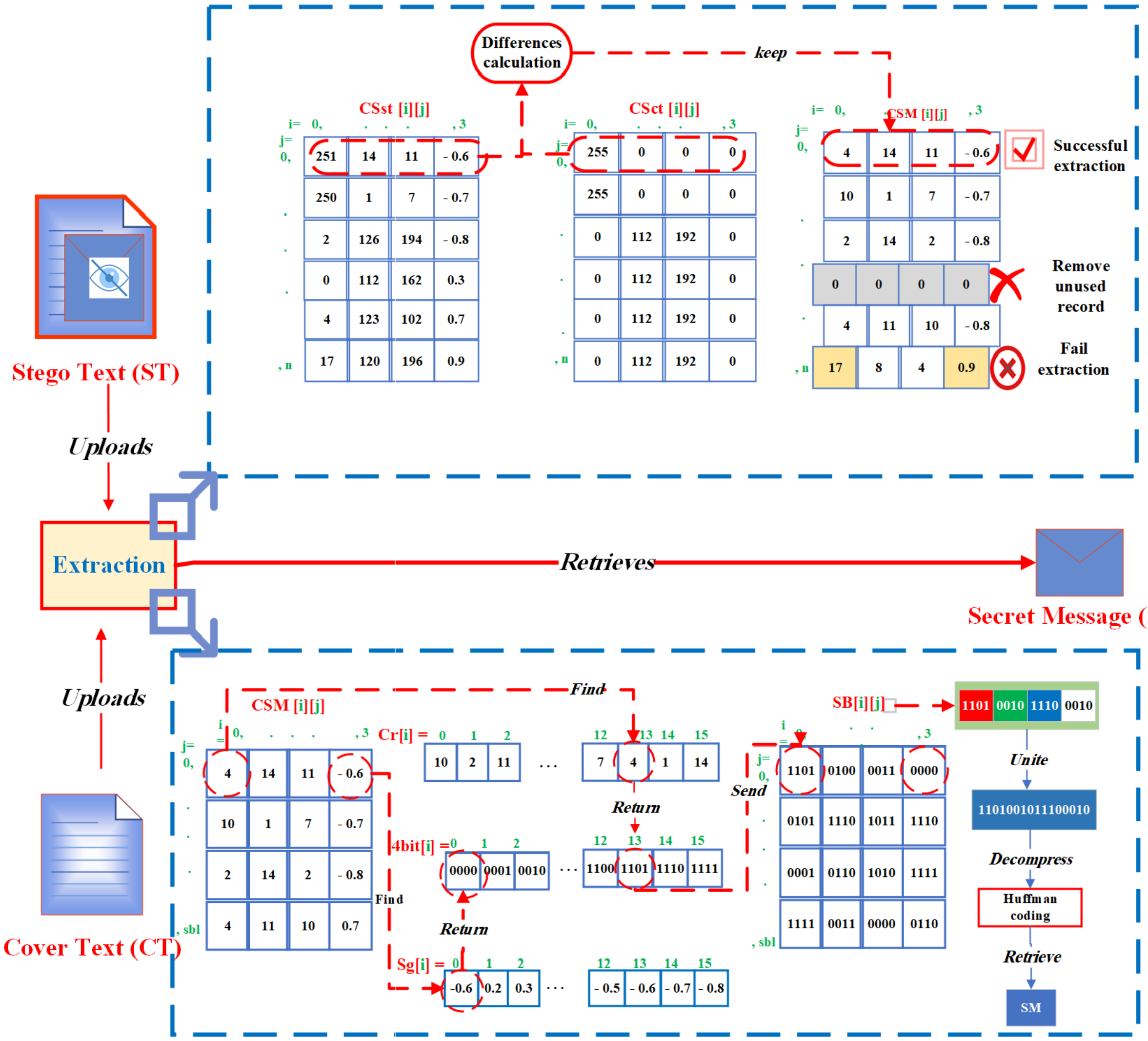


FIGURE 2. Data flow of extraction procedure.

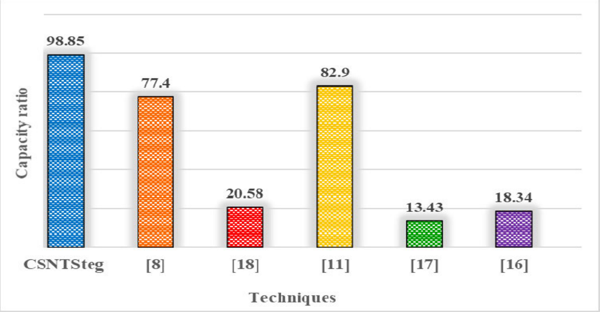


FIGURE 3. Capacity ratio comparison.

embedding method using the same cover text. Then, the size of the secret message is increased until the embedding fails. The failure of embedding happens when the secret message is too large to fit into the cover text. The comparison of capacity ratio results is presented in Fig. 3.

The results show that the capacity ratio increases when increasing the secret message size within the exact size of the cover text. It is evident that the capacity ratio has increased by more than 200%. The CSNTSteg model can hide a secret message twice the length of the cover text. As a result, utilizing compression with an increased bit per location and more useable characters achieves higher capacity.

The average capacity ratio is 98.82% and performs better compared to current RGB techniques and other techniques in text steganography. The capacity ratio in the CSNTSteg model accomplished 98.82%, outperformed the technique in [8] by 21.45%. The high-capacity ratio is achieved by reducing the secret message's size, increasing the bit per location, and adopting all letters, numbers, and symbols in the cover text as embeddable locations.

2) BIT PER LOCATION COMPARISON

This experiment used the same secret message and cover text to compare the bits per location of CSNTSteg with the current RGB techniques in text steganography. The bits per location refers to how many bits can be hidden, depending on the

TABLE 6. Dataset samples.

Secret message	*behind using a cover text is to hide the presence of secret message s the presence of embedded messages in the resulting stego text cannot be easily discovered by anyone except the intended recipient*
Black cover text	*in the research area of text steganography, algorithms based on font format have advantages of great capacity, good imperceptibility and wide application range. However, little work on steganalysis for such algorithms has been reported in the literature. based on the fact that the statistic features of font format will be changed after using font-format-based steganographic algorithms, we present a novel support vector machine-based steganalysis algorithm to detect whether hidden information exists or not. this algorithm can not only effectively detect the existence of hidden information, but also estimate the hidden information length according to variations of font attribute value. as shown by experimental results, the detection accuracy of our algorithm reaches as high as 99.3% when the hidden information length is at least 16 bits.*
Colored cover text.	*in the research area of text steganography, algorithms based on font format have advantages of great capacity, good imperceptibility and wide application range. However, little work on steganalysis for such algorithms has been reported in the literature. based on the fact that the statistic features of font format will be changed after using font-format-based steganographic algorithms, we present a novel support vector machine-based steganalysis algorithm to detect whether hidden information exists or not. this algorithm can not only effectively detect the existence of hidden information, but also estimate the hidden information length according to variations of font attribute value. as shown by experimental results, the detection accuracy of our algorithm reaches as high as 99.3% when the hidden information length is at least 16 bits.*

TABLE 7. Capacity ratio using same cover text and different secret messages.

Experiments	Secret message size	Cover text size	Capacity ratio %
1	198	847	23.38
2	217	847	25.62
3	359	847	42.38
4	622	847	73.44
5	801	847	94.57
6	976	847	115.23
7	1532	847	180.87
8	1993	847	235.30

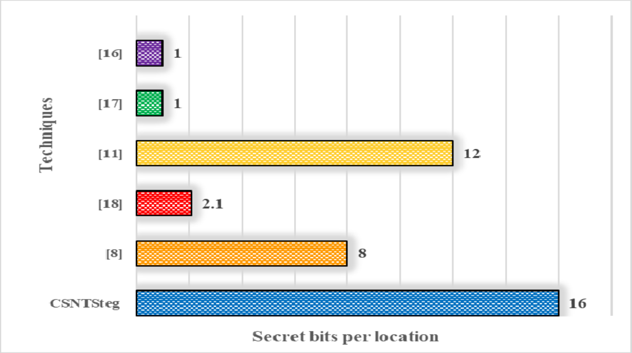


FIGURE 4. Bit per location comparison.

secret block size. Increasing the number of bits per location is the primary concern in existing works. Fig. 4 shows the comparison of bit per location. CSNTSteg embeds 16 secret bits, which is the highest number compared to the others.

This increase relates to enlarged secret block size by combining RGB coding and character spacing. The CSNTSteg model is better than [9], which hides 12 secret bits using RGB coding. While [1] hides only eight secret bits using the RGB code. In addition, the other RGB coding techniques hide fewer secret bits. As a result, combining more text features increases the number of bits per location. However, the combination of the features should not cause visible distortion, which attracts the eavesdropper.

3) MAXIMUM CAPACITY COMPARISON

According to [12], a good embedding algorithm is an algorithm that can hide the secret message with a size that is larger than the size of the cover text. The exact cover text is used to calculate this metric, and the secret message size increases until the embedding fails. Fig. 5 presents the maximum capacity comparison of the same cover text. It shows that the maximum hidden bits in the CSNTSteg model exceed the cover text size by 4792 bits, while [1] it exceeds by 1900 bits. This increase in the maximum capacity of CSNTSteg refers

TABLE 8. Stego RGB codes comparison when the cover is the black text.

Secret block	*Cover RGB Code*	*Stego RGB code*					
		CSNTSteg	**[8]**	**[18]**	**[11]**	**[17]**	**[16]**
1	(0,0,0)	(10,7,9)	(1 ,3 ,3)	(148,0,211)	(6,2,6)	(0, 255, 0)	(255, 112, 197)
2	(0,0,0)	(11,6,9)	(2 ,13 ,14)	(0,0,128)	(5,6,8)	(0,0,255)	(158, 253, 56)
3	(0,0,0)	(7,13,2)	(2 ,0 ,1)	(30,144,255)	(6,9,6)	(0, 255, 0)	(196,86,85)
4	(0,0,0)	(13,11,10)	(2 ,14, 8)	(154,205,50)	(14,6,4)	(0,0,255)	(34,139,34)
5	(0,0,0)	(16,15,4)	(1 ,13 ,7)	(0,128,0)	(2,0,7)	(0, 255, 255)	(38, 144, 215)
6	(0,0,0)	(14,1,15)	(0, 3 ,7)	(0,191,255)	(5,7,3)	(0, 255, 0)	(0,255,255)
7	(0,0,0)	(6,9,15)	(1 ,4, 2)	(255,0,0)	(6,9,6)	(0,0,255)	(78,22,9)
8	(0,0,0)	(12,15,4)	(0 ,1, 9)	(255,215,0)	(14,6,7)	(0, 255, 255)	(160, 2, 92)
9	(0,0,0)	(12,9,5)	(2, 6, 4)	(255,255,0)	(2,0,6)	(255,192,203)	(34,139,34)
10	(0,0,0)	(4,9,3)	(1 ,0 ,0)	(128,0,0)	(1,2,0)	(0,0,0)	(160, 2, 92)

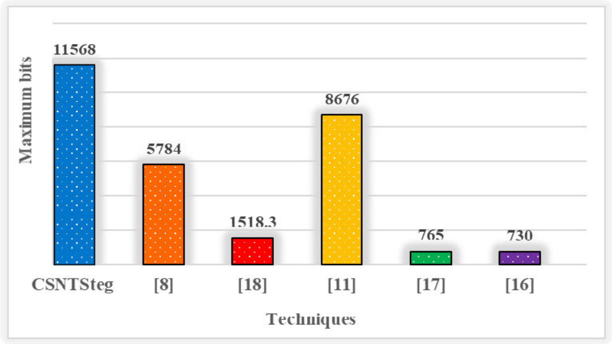


FIGURE 5. Maximin capacity comparison.

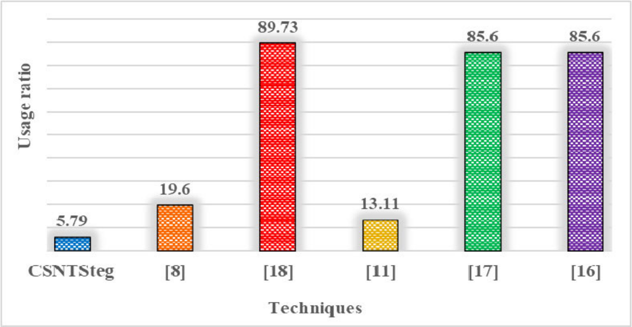


FIGURE 6. Usage ratio comparison.

to the increasing secret block size and the number of usable characters. The number of usable characters in CSNTSteg has increased by making all symbols except normal space in the text embeddable characters. In contrast, techniques in [1], [9] utilize the English alphabet only as embeddable characters.

4) USAGE RATIO COMPARISON

The usage ratio indicates how much space is used, and the remainder is the saved usage. The used secret message has 198 characters, and the used cover text has 847 characters.

The better embedding technique occupies less usage ratio while increasing the hidden amount. Fig. 6 shows the usage ratio comparison, proving that CSNTSteg uses a minimum usage of 5.79% compared to the current works. Therefore, CSNTSteg has a saving usage of 94.21%, and then it can hide more secret bits. On the other hand [7], [8], [11], they use most characters in stego text to conceal the same secret message. Reducing the secret message size and increasing the number of bits per location reduces the usage ratio of cover text.

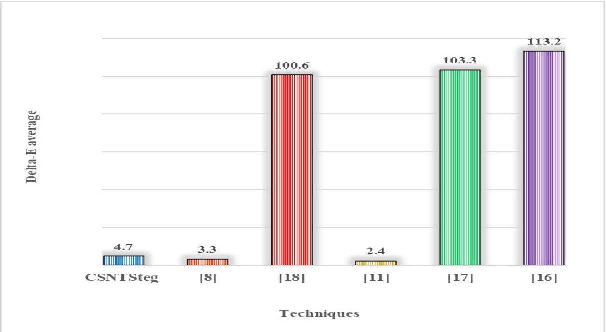
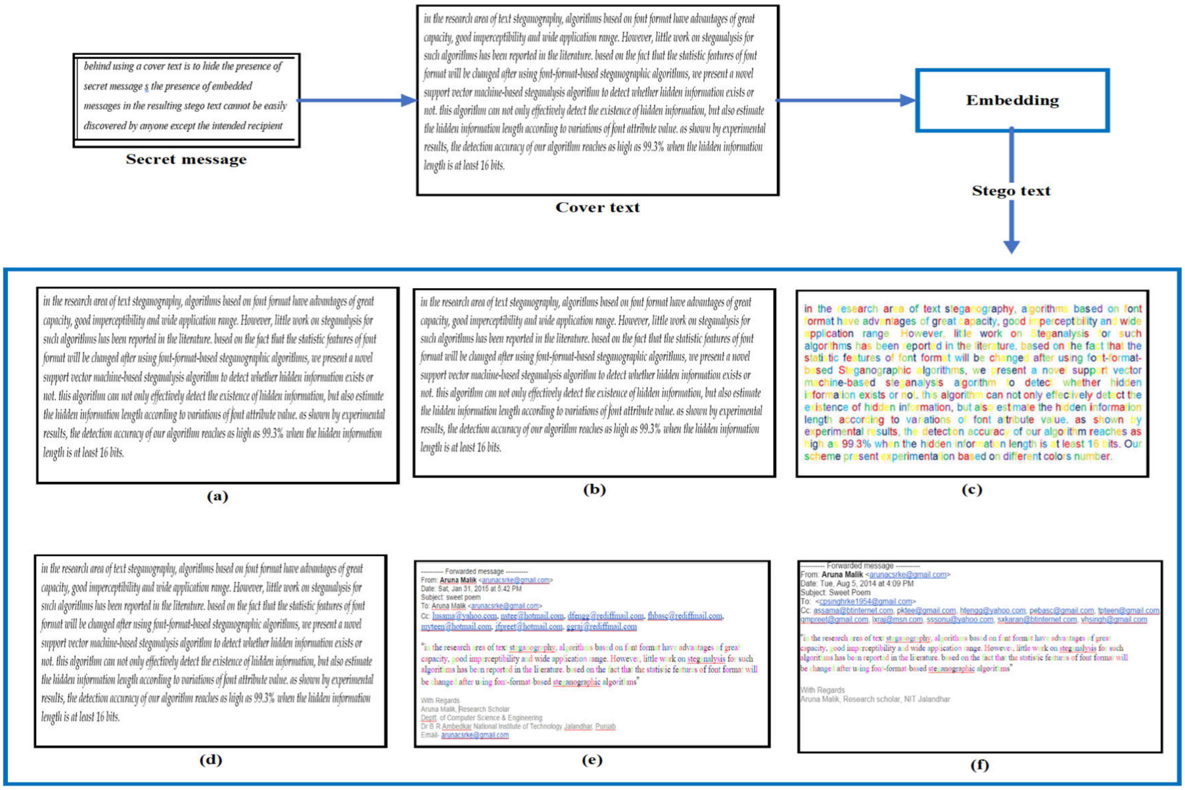


FIGURE 7. Delta-E average comparison when the cover is the black text.

B. INVISIBILITY

This experiment assesses the invisibility of hidden message by altering the RGB code, while the character spacing has been chosen before to be too close to the default value (zero). The results from this experiment have been compared with the current RGB techniques in text steganography. The same secret message and cover text samples were used in this experiment. Most RGB techniques use the cover text in black, while the generated stego text is black or color.



behind using a cover text is to hide the presence of secret message s the presence of embedded messages in the resulting stego text cannot be easily discovered by anyone except the intended recipient

in the research area of text steganography, algorithms based on font format have advantages of great capacity, good imperceptibility and wide application range. However, little work on steganalysis for such algorithms has been reported in the literature. based on the fact that the statistic features of font format will be changed after using font-format-based steganographic algorithms, we present a novel support vector machine-based steganalysis algorithm to detect whether hidden information exists or not. this algorithm can not only effectively detect the existence of hidden information, but also estimate the hidden information length according to variations of font attribute value. as shown by experimental results, the detection accuracy of our algorithm reaches as high as 99.3% when the hidden information length is at least 16 bits.

in the research area of text steganography, algorithms based on font format have advantages of great capacity, good imperceptibility and wide application range. However, little work on steganalysis for such algorithms has been reported in the literature. based on the fact that the statistic features of font format will be changed after using font-format-based steganographic algorithms, we present a novel support vector machine-based steganalysis algorithm to detect whether hidden information exists or not. this algorithm can not only effectively detect the existence of hidden information, but also estimate the hidden information length according to variations of font attribute value. as shown by experimental results, the detection accuracy of our algorithm reaches as high as 99.3% when the hidden information length is at least 16 bits.

in the research area of text steganography, algorithms based on font format have advantages of great capacity, good imperceptibility and wide application range. However, little work on steganalysis for such algorithms has been reported in the literature. based on the fact that the statistic features of font format will be changed after using font-format-based steganographic algorithms, we present a novel support vector machine-based steganalysis algorithm to detect whether hidden information exists or not. this algorithm can not only effectively detect the existence of hidden information, but also estimate the hidden information length according to variations of font attribute value. as shown by experimental results, the detection accuracy of our algorithm reaches as high as 99.3% when the hidden information length is at least 16 bits.

in the research area of text steganography, algorithms based on font format have advantages of great capacity, good imperceptibility and wide application range. However, little work on Steganalysis for such algorithms has been reported in the literature. based on the fact that the statistic features of font format will be changed after using font-format-based Steganographic algorithms, we present a novel support vector machine-based steganalysis algorithm to detect whether hidden information exists or not. this algorithm can not only effectively detect the existence of hidden information, but also estimate the hidden information length according to variations of font attribute value. as shown by experimental results, the detection accuracy of our algorithm reaches as high as 99.3% when the hidden information length is at least 16 bits. Our scheme present experimentation based on different colors number.

in the research area of text steganography, algorithms based on font format have advantages of great capacity, good imperceptibility and wide application range. However, little work on steganalysis for such algorithms has been reported in the literature. based on the fact that the statistic features of font format will be changed after using font-format-based steganographic algorithms, we present a novel support vector machine-based steganalysis algorithm to detect whether hidden information exists or not. this algorithm can not only effectively detect the existence of hidden information, but also estimate the hidden information length according to variations of font attribute value. as shown by experimental results, the detection accuracy of our algorithm reaches as high as 99.3% when the hidden information length is at least 16 bits.

---------- Forwarded message ----------
From: **Aruna Malik** <arunacsrke@gmail.com>
Date: Sat, Jan 31, 2015 at 5:42 PM
Subject: sweet poem
To: Aruna Malik <arunacsrke@gmail.com>
Cc: lusama@yahoo.com, nstee@hotmail.com, dfengg@rediffmail.com, fhbasc@rediffmail.com, myteen@hotmail.com, jfpreet@hotmail.com, ggraj@rediffmail.com

"in the research area of text steganography, algorithms based on font format have advantages of great capacity, good imperceptibility and wide application range. However, little work on steganalysis for such algorithms has been reported in the literature. based on the fact that the statistic features of font format will be changed after using font-format-based steganographic algorithms"

With Regards
Aruna Malik, Research Scholar
Deptt. of Computer Science & Engineering
Dr B R Ambedkar National Institute of Technology Jalandhar, Punjab
Email- arunacsrke@gmail.com

---------- Forwarded message ----------
From: **Aruna Malik** <arunacsrke@gmail.com>
Date: Tue, Aug 5, 2014 at 4:09 PM
Subject: Sweet Poem
To: <cpsinghrke1954@gmail.com>
Cc: assama@btinternet.com, pktee@gmail.com, htengg@yahoo.com, pebasc@gmail.com, tpteen@gmail.com, gmpreet@gmail.com, lxraj@msn.com, sssonu@yahoo.com, sxkaran@btinternet.com, vhsingh@gmail.com

"in the research area of text steganography, algorithms based on font format have advantages of great capacity, good imperceptibility and wide application range. However, little work on steganalysis for such algorithms has been reported in the literature. based on the fact that the statistic features of font format will be changed after using font-format-based steganographic algorithms"

With Regards
Aruna Malik, Research scholar, NIT Jalandhar

FIGURE 8. Stego text comparison when the cover is the black text: (a) CSNTSteg (b) [8] (c) [18] (d) [11] (e) [17] (f) [16].

Delta-E was used to measure the color difference between two RGB codes [9], [13], [14]. The invisibility experiments use both black and colored texts to evaluate the applicability of the CSNTSteg model on any font color while maintaining invisibility.

1) INVISIBILITY COMPARISON WHEN COVER TEXT IS BLACK

The used cover RGB code is black with a (0,0,0) value, while the stego RGB code varies in each technique.

In this part, the cover RGB code is black with a (0,0,0) value, while the stego RGB code varies in each technique. Table 8 displays the RGB code comparison for ten secret blocks when the cover text is black.

This comparison shows that some techniques generate stego RGB code closer to the cover RGB code while others produce rich stego RGB code. To measure the color difference in Table 8, Delta-E was calculated and exhibited in Table 9 for ten secret blocks.

TABLE 9. Delta-E comparison when the cover is the black text.

Secret Block	*Delta-E*					
	CSNTSteg	[8]	[18]	[11]	[17]	[16]
1	2.43	0.90	111.12	2.15	148.47	94.03
2	2.73	4.35	81.31	1.84	137.66	132.33
3	3.22	0.70	87.43	2.68	148.47	70.79
4	6.31	5.35	108.33	3.24	137.66	83.9347
5	7.19	5.10	85.44	2.90	103.99	72.6925
6	3.73	1.95	85.94	2.43	148.47	103.99
7	6.21	1.46	117.34	2.68	137.66	37.25
8	3.15	3.26	123.10	3.22	103.99	70.02
9	3.48	1.95	137.21	2.50	74.69	83.9347
10	5.63	0.28	66.41	0.97	0.00	70.02

Based on Table 2, two groups of Delta-E are displayed. One includes Delta-E values from not perceptible to a slightly different color given in CSNTSteg and [8], [11]. In return, most Delta-E values in [17], [18], [66] have a high color difference. The average Delta-E in each work has been counted and presented in Fig. 7. This illustrates that techniques in [8], [11] achieve slightly better invisibility than the CSNTSteg, because they use the RGB code (0,0,0) as stego, while the CSNTSteg does not consider this value as a stego RGB code. The CSNTSteg model holds the RGB code (0,0,0) to distinguish the unused character from the used characters in the stego text. Therefore, the RGB code limit of CSNTSteg is from (1,1,1) to (16,16,16), while the RGB code limit in [8], [91] is from (0,0,0) to (15,15,15). Thus, the Stego RGB code in [8], [11] is ''only inexperienced'' can notice, while in CSNTSteg, it is noticeable. However, the CSNTSteg reaches only 2.3% less invisibility than [11] and 1.4 less than [8]. Hence, the CSNTSteg model is still not significantly higher [8], [11].

TABLE 10. Stego RGB codes comparison when the cover is the colored text.

Secret Block	*Cover RGB code*	*Stego RGB code*					
		CSNTSteg	**[8]**	**[18]**	**[11]**	**[17]**	**[16]**
1	(255,0,0)	(245,7,9)	(1 ,3 ,3)	(148,0,211)	(6,2,6)	(0, 255, 0)	(255, 112, 197)
2	(0,0,0)	(7,13,1)	(2 ,13 ,14)	(0,0,128)	(5,6,8)	(0,0,255)	(158, 253, 56)
3	(0,0,0)	(11,6,9)	(2 ,0 ,1)	(30,144,255)	(6,9,6)	(0, 255, 0)	(196,86,85)
4	(192,0,0)	(205,11,10)	(2 ,14, 8)	(154,205,50)	(14,6,4)	(0,0,255)	(34,139,34)
5	(0,0,0)	(16,15,4)	(1 ,13 ,7)	(0,128,0)	(2,0,7)	(0, 255, 255)	(38, 144, 215)
6	(0,0,0)	(14,1,15)	(0, 3 ,7)	(0,191,255)	(5,7,3)	(0, 255, 0)	(0,255,255)
7	(112,48,160)	(118,57,175)	(1 ,4, 2)	(255,0,0)	(6,9,6)	(0,0,255)	(78,22,9)
8	(255,192,0)	(243,207,4)	(0 ,1, 9)	(255,215,0)	(14,6,7)	(0, 255, 255)	(160, 2, 92)
9	(0,0,0)	(12,9,5)	(2, 6, 4)	(255,255,0)	(2,0,6)	(255,192,203)	(34,139,34)
10	(0,0,0)	(4,9,3)	(1 ,0 ,0)	(128,0,0)	(1,2,0)	(0,0,0)	(160, 2, 92)

TABLE 11. Delta-E comparison when the cover is the colored text.

Secret Block	*Delta-E*					
	CSNTSteg	**[8]**	**[18]**	**[11]**	**[17]**	**[16]**
1	5.46	117.57	138.88	116.61	170.58	88.18
2	6.05	4.37	81.84	1.86	138.53	131.88
3	2.76	0.71	2.25	2.65	148.04	70.82
4	3.97	94.47	109.65	90.36	163.71	115.24
5	6.27	5.07	85.17	2.94	103.97	73.19
6	2.76	1.99	86.35	2.41	148.04	103.97
7	4.89	78.13	122.66	77.87	66.53	76.80
8	13.10	118.41	13.64	114.52	114.19	112.37
9	3.13	1.93	136.82	2.53	2.38	83.64
10	3.45	0.28	66.49	0.96	0.00	88.18

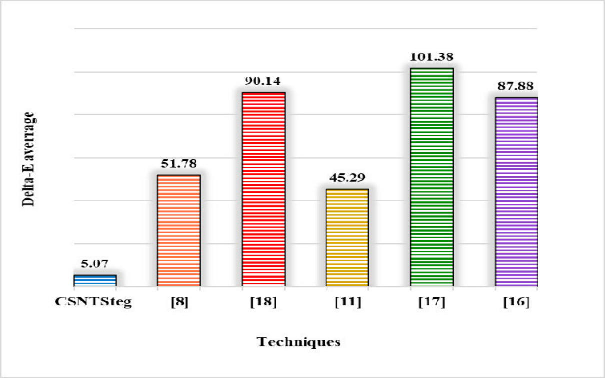


FIGURE 9. Delta-E average comparison when the cover is the colored text.

On the other hand, CSNTSteg still achieves better invisibility than techniques in [17], [18], [66]; these techniques have exceeded the upper limit values of noticeable differences (i.e., 5) of Delta-E and range from 90.3% to 113.2%. Fig. 8 shows the generated stego text from altering the RGB code of the cover text. As reported in [93], the human eye is more tolerant to color variances in small objects or small areas of an image. Hence, the color differences that have been done by CSNTSteg and [8], [11] are more tolerant to the human vision than others.

2) INVISIBILITY COMPARISON WHEN COVER TEXT IS COLORED

This experiment compares the invisibility of the CSNTSteg model and current RGB coding techniques when using cover text's variant color. Table 10 presents the stego RGB code of CSNTSteg and the existing works for ten secret blocks. Then the color differences between the cover and stego texts are calculated using the Delta-E and are displayed in Table 11.

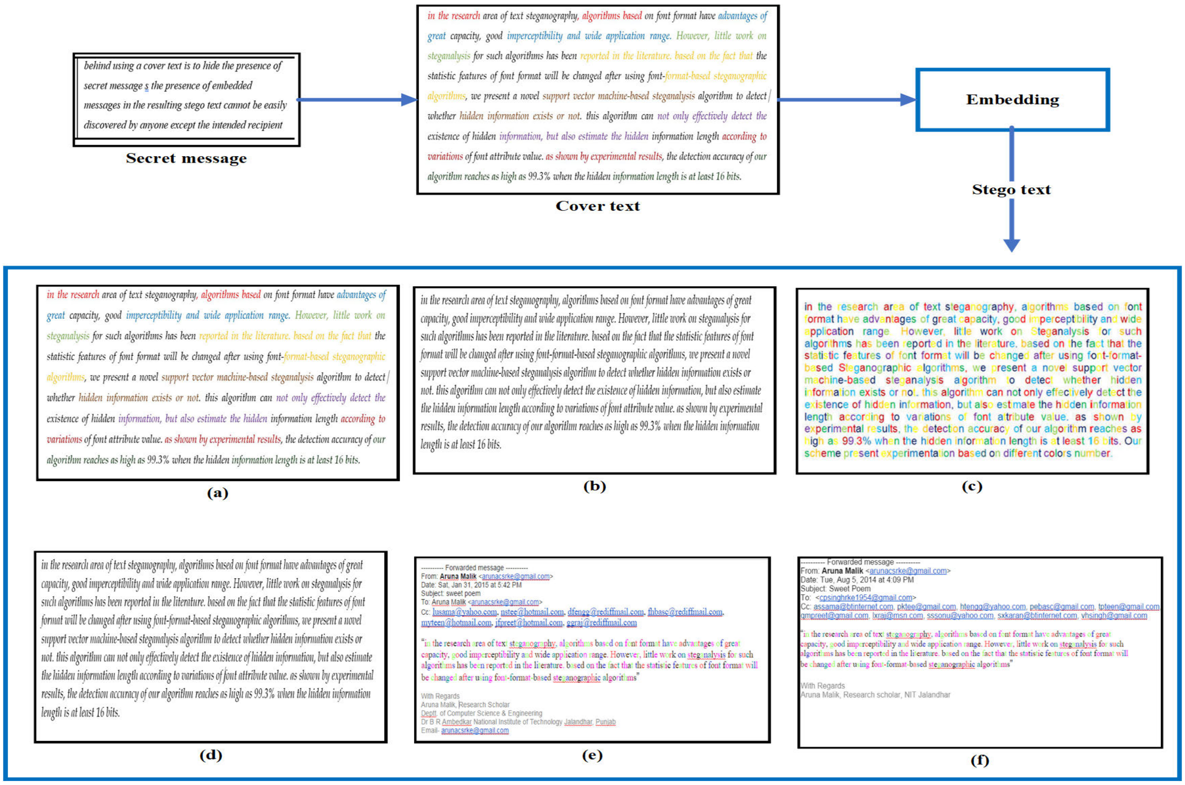
The average comparison of Delta-E for the CSNTSteg and current techniques is exhibited in Fig. 9. It is clear that the CSNTSteg model showed superiority in invisibility when using a colored cover compared to the others.

CSNTSteg improved the invisibility when using colored text by 5.07%, as it did when using black text. Contrarily, in [8], [11], they failed to achieve an invisibility improvement in the colored text, other than its improvement in the black text. This is due to the construction of the normalization mechanism in the CSNTSteg model, which allows hiding the secret message in any color of text. In contrast, the other techniques could not significantly improve invisibility because they did not apply the normalization mechanism. The current RGB coding techniques produce a fixed stego text in black or rich color, regardless of the cover text color.

Conversely, the CSNTSteg model produces closer stego text to the cover text that might be formatted with any color. Fig. 10 presents the stego text in the CSNTSteg model and the existing techniques when the cover text is colored.

C. ROBUSTNESS

This experiment compares the robustness of the CSNTSteg with previous studies. It adopted the same secret message and cover text for all techniques. Table 12 shows the results obtained from this experiment.


behind using a cover text is to hide the presence of secret message s the presence of embedded messages in the resulting stego text cannot be easily discovered by anyone except the intended recipient

in the research area of text steganography, algorithms based on font format have advantages of great capacity, good imperceptibility and wide application range. However, little work on steganalysis for such algorithms has been reported in the literature. based on the fact that the statistic features of font format will be changed after using font-format-based steganographic algorithms, we present a novel support vector machine-based steganalysis algorithm to detect whether hidden information exists or not. this algorithm can not only effectively detect the existence of hidden information, but also estimate the hidden information length according to variations of font attribute value. as shown by experimental results, the detection accuracy of our algorithm reaches as high as 99.3% when the hidden information length is at least 16 bits.

in the research area of text steganography, algorithms based on font format have advantages of great capacity, good imperceptibility and wide application range. However, little work on steganalysis for such algorithms has been reported in the literature. based on the fact that the statistic features of font format will be changed after using font-format-based steganographic algorithms, we present a novel support vector machine-based steganalysis algorithm to detect whether hidden information exists or not. this algorithm can not only effectively detect the existence of hidden information, but also estimate the hidden information length according to variations of font attribute value. as shown by experimental results, the detection accuracy of our algorithm reaches as high as 99.3% when the hidden information length is at least 16 bits.

in the research area of text steganography, algorithms based on font format have advantages of great capacity, good imperceptibility and wide application range. However, little work on steganalysis for such algorithms has been reported in the literature. based on the fact that the statistic features of font format will be changed after using font-format-based steganographic algorithms, we present a novel support vector machine-based steganalysis algorithm to detect whether hidden information exists or not. this algorithm can not only effectively detect the existence of hidden information, but also estimate the hidden information length according to variations of font attribute value. as shown by experimental results, the detection accuracy of our algorithm reaches as high as 99.3% when the hidden information length is at least 16 bits.

in the research area of text steganography, algorithms based on font format have advantages of great capacity, good imperceptibility and wide application range. However, little work on Steganalysis for such algorithms has been reported in the literature. based on the fact that the statistic features of font format will be changed after using font-format-based Steganographic algorithms, we present a novel support vector machine-based steganalysis algorithm to detect whether hidden information exists or not. this algorithm can not only effectively detect the existence of hidden information, but also estimate the hidden information length according to variations of font attribute value. as shown by experimental results, the detection accuracy of our algorithm reaches as high as 99.3% when the hidden information length is at least 16 bits. Our scheme present experimentation based on different colors number.

in the research area of text steganography, algorithms based on font format have advantages of great capacity, good imperceptibility and wide application range. However, little work on steganalysis for such algorithms has been reported in the literature. based on the fact that the statistic features of font format will be changed after using font-format-based steganographic algorithms, we present a novel support vector machine-based steganalysis algorithm to detect whether hidden information exists or not. this algorithm can not only effectively detect the existence of hidden information, but also estimate the hidden information length according to variations of font attribute value. as shown by experimental results, the detection accuracy of our algorithm reaches as high as 99.3% when the hidden information length is at least 16 bits.

---------- Forwarded message ----------
From: **Aruna Malik** <arunacsrke@gmail.com>
Date: Sat, Jan 31, 2015 at 5:42 PM
Subject: sweet poem
To: Aruna Malik <arunacsrke@gmail.com>
Cc: lusama@yahoo.com, nstee@hotmail.com, dfengg@rediffmail.com, fhbasc@rediffmail.com, myteen@hotmail.com, jfpreet@hotmail.com, ggraj@rediffmail.com

"in the research area of text steganography, algorithms based on font format have advantages of great capacity, good imperceptibility and wide application range. However, little work on steganalysis for such algorithms has been reported in the literature. based on the fact that the statistic features of font format will be changed after using font-format-based steganographic algorithms"

With Regards
Aruna Malik, Research Scholar
Deptt. of Computer Science & Engineering
Dr B R Ambedkar National Institute of Technology Jalandhar, Punjab
Email- arunacsrke@gmail.com

---------- Forwarded message ----------
From: **Aruna Malik** <arunacsrke@gmail.com>
Date: Tue, Aug 5, 2014 at 4:09 PM
Subject: Sweet Poem
To: <cpsinghrke1954@gmail.com>
Cc: assama@btinternet.com, pktee@gmail.com, htengg@yahoo.com, pebasc@gmail.com, tpteen@gmail.com, gmpreet@gmail.com, lxraj@msn.com, sssonu@yahoo.com, sxkaran@btinternet.com, vhsingh@gmail.com

"in the research area of text steganography, algorithms based on font format have advantages of great capacity, good imperceptibility and wide application range. However, little work on steganalysis for such algorithms has been reported in the literature. based on the fact that the statistic features of font format will be changed after using font-format-based steganographic algorithms"

With Regards
Aruna Malik, Research scholar, NIT Jalandhar

FIGURE 10. Stego text comparison when the cover is the color text: (a) CSNTSteg (b) [8] (c) [18] (d) [11] (e) [17] (f) [16].

TABLE 12. Losing probability & distortion robustness ratio comparison.

Method	*Cover text size in character*	*Secret text size in character*	*Number of embedding locations*	*Losing probability*	*Distortion robustness ratio*
CSNTSteg	847	198	49	0.0578	94.22
[8]	847	198	166	0.1959	80.41
[18]	847	198	760	0.8972	10.28
[11]	847	198	132	0.1558	84.42
[17]	847	198	723	0.8536	14.64
[16]	847	198	723	0.8536	14.64

From the previous table, using fewer embedding places makes robustness high. It finds that the proposed algorithms in [16], [17] show a lower level of robustness due to the consumption of more embedding positions than the others. Therefore, increasing the number of embedding places leads to poor robustness. Fig. 11 displays the comparison of robustness.

It can be noticed from Fig. 11 that the CSNTSteg model shows better robustness results by 94.22%. This result is slightly close to the results in [8], [11], [17]. Robustness has been increased by reducing the number of embedding places that can be destroyed during structural attacks. In comparison, the rest of the techniques suffer from low robustness because they utilize the most places in the cover text.

As a result, the robustness can be increased when the distortion places are decreased by reducing the number of embedding locations. Reducing this number is achieved by increasing the number of bits per location and reducing the secret message length. It allows a long secret message to be embedded in a few places. Therefore, the CSNTSteg model increases the number of bits per location and uses an efficient compression technique to reduce the embedding locations.

A brief summary of the results of the comparison experiments is presented in Table 13. The CSNTSteg model achieved a high-capacity ratio of 98.85% in a small space of cover text by 5.79%. The CSNTSteg model and techniques proposed by [8], [11] have improved the invisibility by 4.7%, 3.3%, and 2.4%, respectively. Although there is a

TABLE 13. Brief of comparison results.

Author	*Capacity*				*Invisibility*		*Robustness*	*Applicability*	
	Bi/ location	**Maximum capacity**	**Used ratio%**	**Capacity ratio%**	**Average Delta-E in black text**	**Average Delta-E in colored text**	**Distortion robustness ratio**	**Any language**	**Any font color**
CSNTSteg	16	11568	5.79	98.85	4.7	5.07	94.22	×	✓
[8]	8	5784	19.60	77.4	3.3	51.78	80.41	×	×
[18]	2.1	1518.3	89.73	20.58	100.6	90.14	10.28	✓	×
[11]	12	8676	13.11	25.5	2.4	45.29	84.42	×	×
[17]	1	765	86.60	13.43	103.3	101.38	14.64	✓	×
[16]	1	730	85.60	18.34	113.2	87.88	14.64	✓	×

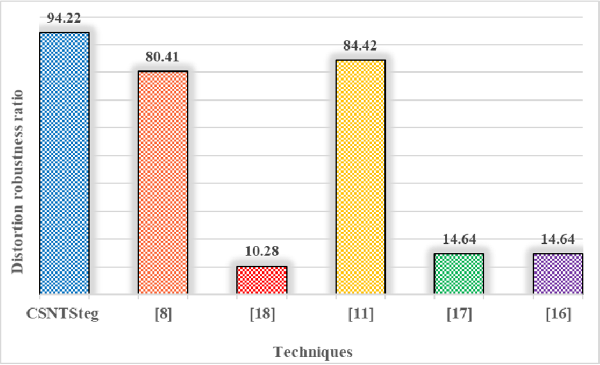


FIGURE 11. Brief of comparison results.

slight improvement in [8], [11] compared to the proposed CSNTSteg model, they failed to maintain a high level of invisibility in the colored text by 51.78 and 45.29. In comparison, CSNTSteg still maintains high invisibility by 5.07 when applying the colored cover text. In addition, the other techniques failed to improve invisibility, whether using colored or black cover text, as the color difference between the two texts is very large.

Despite the fact that CSNTSteg applies to any text color, it has limitations for character spacing when used in languages that join the letters together, such as Arabic, Persian, and Jawi [43], [44], [99], [100]. In contrast, the CSNTSteg model has better results in robustness by 94.22 percent.

V. CONCLUSION

Text steganography provides a secure communication channel by hiding the secret message in text to be delivered safely through the public channel. To overcome the low capacity and invisibility issues of existing text steganography techniques, we proposed the Color and Spacing Normalization stego (CSNTSteg) model, which hides the secret message by altering the cover text's RGB code and character space. The proposed model increases capacity by enlarging the size of the secret block, hiding 16 bits in one location compared to 12 bits as in previous work. The results of this study were compared with existing studies, and they showed a superior capacity ratio of 98.85%, with a 21.85% improvement over the current studies. Although the capacity ratio can be increased by utilizing the other white-space characters, it will increase the size of stego text and raise suspicions. The invisibility result of CSNTSteg with 4.7% showed significant improvements when using black cover text, like some existing techniques, and it also showed higher invisibility than the existing techniques when using colored cover text. The model also enhances robustness to 94.22% by minimizing the stego text distortion. For future, work this paper recommends improving the security against statistical attacks by creating a strong stego key. Also, the normalization of RGB code can be expanded to include image steganography.

ACKNOWLEDGMENT

The authors would like to acknowledge Universiti Putra Malaysia (UPM) for providing support on part works. The first author would also like to express his gratitude to the Ministry of Higher Education and Scientific Research of Yemen for their support in tuition fees.

REFERENCES

[1] C. Mcclain, E. Vogels, A. Perrin, S. Sechopoulos, and L. Rainie. *Key Internet Statistics to Know in 2021 (Including Mobile)*. Washington, DC, USA, Accessed: Dec. 7, 2021. [Online]. Available: https://www.broadbandsearch.net/blog/internet-statistics#post-navigation-1

[2] J. Boehm, J. Kaplan, M. Sorel, N. Sportsman, and T. Steen. *Cybersecurity tactics for the Coronavirus Pandemic*. Accessed: Dec. 7, 2021. [Online]. Available: https://www.mckinsey.com/business-functions/risk-and-resilience/our-insights/cybersecurity-tactics-for-the-coronavirus-pandemic

[3] A. H. Mohsin, A. A. Zaidan, B. B. Zaidan, S. A. B. Ariffin, O. S. Albahri, A. S. Albahri, M. A. Alsalem, K. I. Mohammed, and M. Hashim, "Real-time medical systems based on human biometric steganography: A systematic review," *J. Med. Syst.*, vol. 42, no. 12, pp. 1–20, Oct. 2018.

[4] C. Mcclain, E. Vogels, A. Perrin, S. Sechopoulos, and L. Rainie. *The Internet and the Pandemic*. Washington, DC, USA. Accessed: Dec. 7, 2021. [Online]. Available: https://www.pewresearch.org/internet/2021/09/01/the-internet-and-the-pandemic/pi_2021-09-01_covid-and-tech_0-02/

[5] M. T. Ahvanooey, Q. Li, J. Hou, H. D. Mazraeh, and J. Zhang, "AITSteg: An innovative text steganography technique for hidden transmission of text message via social media," *IEEE Access*, vol. 6, pp. 65981–65995, 2018.

[6] S. Mahato, D. A. Khan, and D. K. Yadav, "A modified approach to data hiding in Microsoft word documents by change-tracking technique," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 32, no. 2, pp. 216–224, Feb. 2020.

[7] M. Taleby Ahvanooey, Q. Li, J. Hou, A. R. Rajput, and C. Yini, "Modern text hiding, text steganalysis, and applications: A comparative analysis," *Entropy*, vol. 21, no. 4, p. 355, Apr. 2019.

[8] B. Osman, "Message hiding technique in text steganography using RGB colour approach and random location," Ph.D. dissertation, Dept. Sch. Comp., UUM Univ., Kedah, Malaysia, 2020.

[9] M. Aman, A. Khan, B. Ahmad, and S. Kouser, "A hybrid text steganography approach utilizing Unicode space characters and zero-width character," *Int. J. Inf. Technol. Secur.*, vol. 9, no. 1, pp. 85–100, Feb. 2017.

[10] R. Thabit, N. I. Udzir, S. M. Yasin, A. Asmawi, N. A. Roslan, and R. Din, "A comparative analysis of Arabic text steganography," *Appl. Sci.*, vol. 11, no. 15, p. 6851, Jul. 2021.

[11] A. F. Al-Azzawi, "A multi-layer hybrid text steganography for secret communication using word tagging and RGB color coding," *Int. J. Netw. Secur. Appl.*, vol. 10, pp. 1–12, Nov. 2018.

[12] S. M. A. Al-Nofaie and A. A.-A. Gutub, "Utilizing pseudo-spaces to improve Arabic text steganography for multimedia data communications," *Multimedia Tools Appl.*, vol. 79, nos. 1–2, pp. 19–67, Jan. 2020.

[13] N. Naqvi, A. T. Abbasi, R. Hussain, M. A. Khan, and B. Ahmad, "Multilayer partially homomorphic encryption text steganography (MLPHE-TS): A zero steganography approach," *Wireless Pers. Commun.*, vol. 103, no. 2, pp. 1563–1585, May 2018.

[14] M. Khairullah, "A novel steganography method using transliteration of Bengali text," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 31, no. 3, pp. 348–366, Jul. 2019.

[15] M. Azeem, J. He, K. G. Rana, and F. A. Rajpoot, "A cryptographic data hiding algorithm with high cover text capacity," *Int. J. Electron. Secur. Digit. Forensics*, vol. 11, pp. 225–244, Jan. 2019.

[16] R. Kumar, A. Malik, S. Singh, and S. Chand, "A high capacity email based text steganography scheme using Huffman compression," in *Proc. 3rd Int. Conf. Signal Process. Integr. Netw. (SPIN)*, Noida, India, Feb. 2016, pp. 53–56.

[17] A. Malik, G. Sikka, and H. K. Verma, "A high capacity text steganography scheme based on LZW compression and color coding," *Eng. Sci. Technol. Int. J.*, vol. 20, no. 1, pp. 72–79, 2017.

[18] J. K. Sadié, L. Moyou Metcheka, and R. Ndoundam, "Two high capacity text steganography schemes based on color coding," 2020, *arXiv:2004.00948*.

[19] E. Walia, P. Jain, and N. Navdeep, "An analysis of LSB & DCT based steganography," *Global J. Comput. Sci. Technol.*, vol. 10, no. 1, pp. 4–8, Apr. 2010.

[20] X. Liao, J. Yin, M. Chen, and Z. Qin, "Adaptive payload distribution in multiple images steganography based on image texture features," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 2, pp. 897–911, Mar./Apr. 2022.

[21] I. J. Kadhim, P. Premaratne, P. J. Vial, and B. Hallorana, "Comprehensive survey of image steganography: Techniques, evaluations, and trends in future research," *Neurocomputing*, vol. 335, pp. 299–326, Mar. 2019.

[22] A. Kumar and K. Pooja, "Steganography–A data hiding technique," *Int. J. Comput. Appl.*, vol. 9, no. 7, pp. 19–23, Nov. 2010.

[23] E. Almehmadi and A. Gutub, "Novel Arabic e-text watermarking supporting partial dishonesty based on counting-based secret sharing," *Arabian J. Sci. Eng.*, vol. 47, pp. 2585–2609, Nov. 2021.

[24] V. Aslantas, "A singular-value decomposition-based image watermarking using genetic algorithm," *AEU-Int. J. Electron. Commun.*, vol. 62, no. 5, pp. 386–394, May 2008.

[25] M. Taleby Ahvanooey, Q. Li, H. J. Shim, and Y. Huang, "A comparative analysis of information hiding techniques for copyright protection of text documents," *Secur. Commun. Netw.*, vol. 2018, pp. 1–22, 2018.

[26] K. Bailey, K. Curran, and J. Condell, "Evaluation of pixel-based steganography and stegodetection methods," *Imag. Sci. J.*, vol. 52, no. 3, pp. 131–150, Sep. 2004.

[27] E. Setyaningsih, R. Wardoyo, and A. K. Sari, "Securing color image transmission using compression-encryption model with dynamic key generator and efficient symmetric key distribution," *Digit. Commun. Netw.*, vol. 6, no. 4, pp. 486–503, Nov. 2020.

[28] M. Y. Valandar, M. J. Barani, P. Ayubi, and M. Aghazadeh, "An integer wavelet transform image steganography method based on 3D sine chaotic map," *Multimedia Tools Appl.*, vol. 78, no. 8, pp. 9971–9989, 2019.

[29] M. Y. Valandar, P. Ayubi, and M. J. Barani, "A new transform domain steganography based on modified logistic chaotic map for color images," *J. Inf. Secur. Appl.*, vol. 34, pp. 142–151, Jun. 2017.

[30] P. Ayubi, S. Setayeshi, and A. M. Rahmani, "Deterministic chaos game: A new fractal based pseudo-random number generator and its cryptographic application," *J. Inf. Secur. Appl.*, vol. 52, Jun. 2020, Art. no. 102472.

[31] S. Jiang, D. Ye, J. Huang, Y. Shang, and Z. Zheng, "SmartSteganography: Light-weight generative audio steganography model for smart embedding application," *J. Netw. Comput. Appl.*, vol. 165, Sep. 2020, Art. no. 102689.

[32] P. Ayubi, M. Jafari Barani, M. Yousefi Valandar, B. Yosefnezhad Irani, and R. Sedagheh Maskan Sadigh, "A new chaotic complex map for robust video watermarking," *Artif. Intell. Rev.*, vol. 54, no. 2, pp. 1237–1280, Feb. 2021.

[33] Z.-L. Yang, X.-Q. Guo, Z.-M. Chen, Y.-F. Huang, and Y.-J. Zhang, "RNN-stega: Linguistic steganography based on recurrent neural networks," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 5, pp. 1280–1295, May 2019.

[34] J. Zhang, Y. Xie, L. Wang, and H. Lin, "Coverless text information hiding method using the frequent words distance," in *Proc. 3rd Int. Conf. Cloud Comput. Secur.* Cham, Switzerland: Springer, Oct. 2017, pp. 121–132.

[35] L. Xiang, G. Guo, J. Yu, V. S. Sheng, and P. Yang, "A convolutional neural network-based linguistic steganalysis for synonym substitution steganography," *Math. Biosci. Eng.*, vol. 17, no. 2, pp. 1041–1058, 2020.

[36] Z. Jalil and A. M. Mirza, "A robust zero-watermarking algorithm for copyright protection of text documents," *J. Chin. Inst. Eng.*, vol. 36, no. 2, pp. 180–189, Mar. 2013.

[37] R. Kumar, S. Chand, and S. Singh, "An efficient text steganography scheme using unicode space characters," *Int. J. Forensic Comput. Sci.*, vol. 10, no. 1, pp. 8–14, Sep. 2015.

[38] M. Gaur and M. Sharma, "A new PDAC (parallel encryption with digit arithmetic of cover text) based text steganography approach for cloud data security," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 3, no. 3, pp. 1344–1352, Mar. 2015.

[39] A. Wilson, P. Blunsom, and A. D. Ker, "Linguistic steganography on Twitter: hierarchical language modeling with manual interaction," *Proc. SPIE*, vol. 9028, Feb. 2014, Art. no. 902803.

[40] A. Odeh, K. Elleithy, and M. Faezipour, "Steganography in text by using MS word symbols," in *Proc. Zone 1 Conf. Amer. Soc. Eng. Educ.*, Apr. 2014, pp. 1–5.

[41] B. Gupta Banik and S. K. Bandyopadhyay, "Novel text steganography using natural language processing and Part-of-Speech tagging," *IETE J. Res.*, vol. 66, no. 3, pp. 384–395, Jul. 2018.

[42] J. Gu and Y. Cheng, "A watermarking scheme for natural language documents," in *Proc. 2nd IEEE Int. Conf. Inf. Manage. Eng.*, Chengdu, China, Apr. 2010, pp. 461–464.

[43] N. Alanazi, E. Khan, and A. Gutub, "Involving spaces of unicode standard within irreversible Arabic text steganography for practical implementations," *Arabian J. Sci. Eng.*, vol. 46, pp. 8869–8885, May 2021.

[44] S. Al-Nofaie, A. Gutub, and M. Al-Ghamdi, "Enhancing Arabic text steganography for personal usage utilizing pseudo-spaces," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 33, no. 8, pp. 963–974, Oct. 2021.

[45] A. A.-A. Gutub and K. A. Alaseri, "Refining Arabic text stego-techniques for shares memorization of counting-based secret sharing," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 33, no. 9, pp. 1108–1120, Nov. 2021.

[46] A. Gutub and M. Fattani, "A novel Arabic text steganography method using letter points and extensions," in *Proc. WASET Int. Conf. Comput. Inf. Sys. Sci. Eng. (ICCISSE)*, May 2007, pp. 25–27.

[47] A. Ditta, Y. Cai, M. Azeem, K. G. Rana, H. Yu, and M. Q. Memon, "Information hiding: Arabic text steganography by using Unicode characters to hide secret data," *Int. J. Electron. Secur. Digit. Forensics*, vol. 10, pp. 61–78, Jan. 2018.

[48] S. S. Baawi, M. Mokhtar, and R. Sulaiman, "New text steganography technique based on a set of two-letter words," *J. Theor. Appl. Inf. Technol.*, vol. 95, pp. 6247–6255, Nov. 2017.

[49] A. Taha, A. S. Hammad, and M. M. Selim, "A high capacity algorithm for information hiding in Arabic text," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 32, no. 6, pp. 658–665, Jul. 2020.

[50] H. J. Shiu, B. S. Lin, B. S. Lin, P. Y. Huang, C. H. Huang, and C. L. Lei, "Data hiding on social media communications using text steganography," in *Proc. Int. Conf. Risks Secur. Internet Syst.*, Dinard, France, Sep. 2017, pp. 217–224.

[51] B. Khosravi, B. Khosravi, B. Khosravi, and K. Nazarkardeh, "A new method for pdf steganography in justified texts," *J. Inf. Secur. Appl.*, vol. 45, pp. 61–70, Apr. 2019.

[52] S. Al-Nofaie, M. Fattani, and A. Gutub, "Capacity improved Arabic text steganography technique utilizing 'Kashida' with whitespaces," in *Proc. 3rd Int. Conf. Math. Sci. Comput. Eng. (ICMSCE)*, Kuala Lumpur, Malaysia, Feb. 2016, pp. 38–44.

[53] N. Alanazi, E. Khan, and A. Gutub, "Efficient security and capacity techniques for Arabic text steganography via engaging unicode standard encoding," *Multimedia Tools Appl.*, vol. 80, no. 1, pp. 1403–1431, Jan. 2021.

[54] A. A. Hamzah, S. Khattab, and H. Bayomi, "A linguistic steganography framework using Arabic calligraphy," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 33, no. 7, pp. 865–877, Sep. 2021.

[55] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding—A survey," *Proc. IEEE*, vol. 87, no. 7, pp. 1062–1078, Jul. 1999.

[56] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, *Digital Watermarking and Steganography*, 2nd ed. San Mateo, CA, USA: Morgan Kaufmann, 2007.

[57] J. Wen, X. Zhou, M. Li, P. Zhong, and Y. Xue, "A novel natural language steganographic framework based on image description neural network," *J. Vis. Commun. Image Represent.*, vol. 61, pp. 157–169, May 2019.

[58] B. K. Ramakrishnan, P. K. Thandra, and A. S. M. Srinivasula, "Text steganography: A novel character-level embedding algorithm using font attribute," *Secur. Commun. Netw.*, vol. 9, no. 18, pp. 6066–6079, Feb. 2016.

[59] S. S. Baawi, M. R. Mokhtar, and R. Sulaiman, "A comparative study on the advancement of text steganography techniques in digital media," *ARPN J. Eng. Appl. Sci.*, vol. 13, pp. 1854–1863, Mar. 2018.

[60] X. Liao, Y. Yu, B. Li, Z. Li, and Z. Qin, "A new payload partition strategy in color image steganography," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 3, pp. 685–696, Jan. 2019.

[61] M. Y. Valandar, M. J. Barani, and P. Ayubi, "A blind and robust color images watermarking method based on block transform and secured by modified 3-dimensional Hénon map," *Soft Comput.*, vol. 24, no. 2, pp. 771–794, Jan. 2020.

[62] M. Jafari Barani, M. Yousefi Valandar, and P. Ayubi, "A new digital image tamper detection algorithm based on integer wavelet transform and secured by encrypted authentication sequence with 3D quantum map," *Optik*, vol. 187, pp. 205–222, Jun. 2019.

[63] S. T. Ali Shah, D. A. Khan, and D. A. Hussain, "Text steganography using character spacing after normalization," *Int. J. Sci. Eng. Res.*, vol. 11, no. 2, pp. 949–957, Feb. 2020.

[64] S. H. Low, N. F. Maxemchuk, J. T. Brassil, and L. O'Gorman, "Document marking and identification using both line and word shifting," in *Proc. INFOCOM*, vol. 2, Apr. 1995, pp. 853–860.

[65] S. Changder, N. C. Debnath, and D. Ghosh, "A new approach to Hindi text steganography by shifting matra," in *Proc. Int. Conf. Adv. Recent Technol. Commun. Comput.*, Kottayam, India, 2009, pp. 199–202.

[66] R. Kumar, A. Malik, S. Singh, B. Kumar, and S. Chand, "A space based reversible high capacity text steganography scheme using font type and style," in *Proc. Int. Conf. Comput., Commun. Autom. (ICCCA)*, Greater Noida, India, Apr. 2016, pp. 1090–1094.

[67] S. Mahato, D. K. Yadav, and D. Khan, "A novel approach to text steganography using font size of invisible space characters in Microsoft word document," in *Proc. Int. Conf. Adv. Comput. Netw. Inform.*, New Delhi, India, Jun. 2013, vol. 234.

[68] T. Y. Liu and W. H. Tsai, "A new steganographic method for data hiding in Microsoft word documents by a change tracking technique," *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 1, pp. 24–30, Mar. 2007.

[69] A. Gutub, F. Al-Haidari, K. M. Al-Kahsah, and J. Hamodi, "e-text watermarking: Utilizing 'Kashida' extensions in Arabic language electronic writing," *J. Emerg. Technol. Web Intell.*, vol. 2, no. 1, pp. 48–55, Feb. 2010.

[70] L. Xiang, W. Wu, X. Li, and C. Yang, "A linguistic steganography based on word indexing compression and candidate selection," *Multimedia Tools Appl.*, vol. 77, no. 21, pp. 28969–28989, May 2018.

[71] E. Satir and H. Isik, "A Huffman compression based text steganography method," *Multimedia Tools Appl.*, vol. 70, no. 3, pp. 2085–2110, Jun. 2014.

[72] A. H. Ali, L. E. George, A. A. Zaidan, and M. R. Mokhtar, "High capacity, transparent and secure audio steganography model based on fractal coding and chaotic map in temporal domain," *Multimed. Tools Appl.*, vol. 77, no. 23, pp. 31487–31516, Dec. 2018.

[73] N. Alanazi, E. Khan, and A. Gutub, "Inclusion of unicode standard seamless characters to expand Arabic text steganography for secure individual uses," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 4, pp. 1343–1356, Apr. 2022.

[74] R. Din, R. A. Thabit, N. I. Udzir, and S. Utama, "Traid-bit embedding process on Arabic text steganography method," *Bull. Electr. Eng. Informat.*, vol. 10, no. 1, pp. 493–500, Feb. 2021.

[75] N. A. Roslan, N. I. Udzir, R. Mahmod, Z. A. Zukarnain, M. I. H. Ninggal, and R. Thabit, "Character property method for Arabic text steganography with biometric multifactor authentication using liveness detection," *J. Theor. Appl. Inf. Technol.*, vol. 98, pp. 4140–4157, Dec. 2020.

[76] N. A. Roslan, R. Mahmod, N. I. Udzir, and Z. A. Zurkarnain, "Primitive structural method for high capacity text steganography," *J. Theor. Appl. Inf. Technol.*, vol. 67, no. 2, pp. 373–383, Dec. 2014.

[77] R. Zhang, S. Dong, and J. Liu, "Invisible steganography via generative adversarial networks," *Multimedia Tools Appl.*, vol. 78, no. 7, pp. 8559–8575, Apr. 2019.

[78] M. Tancik, B. Mildenhall, and R. Ng, "StegaStamp: Invisible hyperlinks in physical photographs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2117–2126.

[79] N. Hamid, A. Yahya, R. B. Ahmad, and O. M. Al-Qershi, "Image steganography techniques: An overview," *Int. J. Comput. Sci. Secu.*, vol. 6, no. 3, pp. 168–187, Nov. 2012.

[80] V. Korzhik, I. Fedyanin, and N. D. Cuong, "Detection of stegosystems using block ciphers for encryption of the embedded messages," in *Proc. 20th Conf. Open Innov. Assoc. (FRUCT)*, Saint Petersburg, Russia, Apr. 2017.

[81] M. Zaheer, "Secure communication using steganography in image processing," Ph.D. dissertation, Dept. Elect. Eng., Air Univ., Islamabad, Pakistan, 2017.

[82] K. Vaishakh, A. Pravalika, D. Abhishek, N. Meghana, and G. Prasad, "A semantic approach to text steganography in Sanskrit using numerical encoding," in *Recent Findings in Intelligent Computing Techniques*, vol. 707. Singapore: Springer, Nov. 2019, pp. 181–192.

[83] A. Gutub and K. Alaseri, "Hiding shares of counting-based secret sharing via Arabic text steganography for personal usage," *Arabian J. Sci. Eng.*, vol. 45, no. 4, pp. 2433–2458, Apr. 2020.

[84] S. G. R. Ekodeck and R. Ndoundam, "PDF steganography based on Chinese remainder theorem," *J. Inf. Secur. Appl.*, vol. 29, pp. 1–15, Aug. 2016.

[85] O. Afanasyeva, "Analysis of aspects of messages hiding in text environments," *J. KONBiN*, vol. 34, no. 1, pp. 5–16, Sep. 2015.

[86] M.-W. Chao, C.-H. Lin, C.-W. Yu, and T.-Y. Lee, "A high capacity 3D steganography algorithm," *IEEE Trans. Vis. Comput. Graphics*, vol. 15, no. 2, pp. 274–284, Mar. 2009.

[87] N. Li, J. Hu, R. Sun, S. Wang, and Z. Luo, "A high-capacity 3D steganography algorithm with adjustable distortion," *IEEE Access*, vol. 5, pp. 24457–24466, 2017.

[88] J. Kour and D. Verma, "Steganography techniques–A review paper," *Int. J. Emerg. Res. Manage. Technol.*, vol. 3, no. 5, pp. 132–135, Oct. 2014.

[89] M. A. Jaro, "Advances in record-linkage methodology as applied to matching the 1985 census of tampa, Florida," *J. Amer. Stat. Assoc.*, vol. 84, no. 406, pp. 414–420, Jun. 1989.

[90] T. Kellen. (2001). *Hiding in Plain View: Could Steganography be a Terrorist Tool?*. [Online]. Available: https://www.sans.org/white-papers/551/

[91] W. E. Winkler, "String comparator metrics and enhanced decision rules in the Fellegi-Sunter model of record linkage," in *Proc. Sec. Surv. Res.*, Washington, DC, USA, Mar. 1990, pp. 354–359.

[92] E. Zielinska, W. Mazurczyk, and K. Szczypiorski, "Development trends in steganography," *Commu. ACM*, vol. 2, no. 1, pp. 1–13, Jul. 2013.

[93] W. Mokrzycki and M. Tatol, "Colour differenceΔ EA survey," *Mach. Graph. Vis.*, vol. 20, no. 4, pp. 383–411, Apr. 2011.

[94] I. Patel and J. Goud, "Colour recognition for blind and colour blind people," *Int. J. Eng. Technol. Innov.*, vol. 2, no. 6, pp. 38–42, Apr. 2012.

[95] M. Zamani, A. Manaf, R. B. Ahmad, F. Jaryani, H. Taherdoost, and A. M. Zeki, "A secure audio steganography approach," in *Proc. Int. Conf. for Internet Technol. Secured Trans., (ICITST)*, London, U.K., Nov. 2009, pp. 1–6.

[96] R. S. Sabri, R. Din, and A. Mustapha, "Analysis review on performance metrics for extraction schemes in text steganography," *Indonesian J. Electr. Eng. Comput. Sci.*, vol. 11, no. 2, p. 761, Aug. 2018.

[97] R. Din, S. Utama, S. H. Hanizan, M. M. Hilal, M. A. M. Hanif, A. Zulhazlin, and G. M. Fazali, "Evaluating the feature-based technique of text steganography based on capacity and time processing parameters," *Adv. Sci. Lett.*, vol. 24, no. 10, pp. 7355–7359, Oct. 2018.

[98] B. Osman, A. Yasin, and M. N. Omar, "An analysis of alphabet-based techniques in text steganography," *J. Telecommun., Electron. Comput. Eng.*, vol. 8, no. 10, pp. 109–115, Jan. 2016.

[99] M. Alkhudaydi and A. Gutub, "Securing data via cryptography and Arabic text steganography," *Social Netw. Comput. Sci.*, vol. 2, no. 1, pp. 1–18, Feb. 2021.

[100] Y. Khan, A. Algarni, A. Fayomi, and A. M. Almarashi, "Disbursal of text steganography in the space of double-secure algorithm," *Math. Problems Eng.*, vol. 2021, Dec. 2021, Art. no. 7336474.

REEMA THABIT received the B.Sc. degree in computer science and engineering from Aden University, Yemen, in 2004, the M.Sc. degree in information technology from Universiti Utara Malaysia (UUM), Malaysia, in 2016, and the Ph.D. degree in computer science and information technology from Universiti Putra Malaysia (UPM), Malaysia, in 2022. From 2005 to 2013, she was a Lecturer with the Aden Community College, Yemen. Her current research interests include information hiding, security in multimedia, data compression, AI, and the IoT.

NUR IZURA UDZIR received the bachelor's degree in computer science and the M.Sc. degree from Universiti Putra Malaysia (UPM), in 1995 and 1998, respectively, and the Ph.D. degree in computer science from the University of York, U.K., in 2006. She has been an Associate Professor at the Faculty of Computer Science and Information Technology, UPM, since 1998. Her research interests include computer security, intrusion detection systems, access control, secure operating systems, coordination models and languages, and distributed systems. She is a member of the IEEE Computer Society, and a certified Professional Technologist from the Malaysian Board of Technologists (MBOT), Information Security Professionals Association of Malaysia (ISPA.my), and the Malaysian Society for Cryptology Research (MSCR). She has won the MIMOS Prestigious Award, in 2015, the Young Scientist Award, in 2021, and seven Best Paper Awards at international conferences. In 2014, 2019, and 2020, she was invited as a Visiting Lecturer/Foreign Scientist at the M. Auezov South Kazakhstan State University, Kazakhstan.

SHARIFAH MD YASIN received the bachelor's degree in mathematics and statistics from the University of Bradford, U.K., in 1991, the master's degree in information technology from Universiti Kebangsaan Malaysia, Malaysia, in 2002, and the Ph.D. degree in computer security from Universiti Putra Malaysia (UPM), in 2011. She is currently a Senior Lecturer at the Faculty of Computer Science and Information Technology, UPM. She is also a Research Associate with the Institute of Mathematical Research (INSPEM), UPM. Her current research interests include searchable encryption, block cipher, elliptic curve cryptography, attribute-based encryption, authenticated encryption, secret sharing scheme, and blockchain.

AZIAH ASMAWI received the Ph.D. degree in database security from Universiti Putra Malaysia (UPM), in 2017. She is currently a member of the Department of Computer Science, Faculty of Computer Science and Information Technology, University Putra Malaysia (UPM). She is also doing research for teaching and learning (T&L) scopes. She became the leader of several research projects. Her main research interests include security in computing, intrusion detection systems, and digital forensics. She has published many papers that are related to her field and is an editorial board members of several international journals.

ADNAN ABDUL-AZIZ GUTUB received the B.S. degree in electrical engineering and the M.S. degree in computer engineering from the King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia, and the Ph.D. degree in electrical and computer engineering from Oregon State University, Corvallis, OR, USA, in 2002. He worked as an Associate Professor, an Assistant Professor, a Lecturer, and a Graduate Assistant in computer engineering at KFUPM. He was a leader of the Center of Research Excellence in Hajj and Omrah (HajjCoRE) serving as the HajjCoRE Director for around three years until the end of 2013. Then, he was assigned his position (until March 2016) as the Vice Dean of the Custodian of the Two Holy Mosques Institute of the Hajj and Umrah Research, known publicly as the Hajj Research Institute (HRI), within Umm Al-Qura University (UQU), Makkah, Saudi Arabia. He is currently a Full Professor in information and computer security with the Department of Computer Engineering, College of Computers and Information Systems, UQU. He worked on designing efficient integrated circuits for the Montgomery inverse computation in different finite fields. He has done some work in modeling architectures for RSA and elliptic curve crypto operations. His interest in computer security also involved steganography and secret-sharing focusing on image-based steganography and Arabic text steganography as well as counting-based secret sharing. His research work can be observed through his more than 150 publications (international journals and conferences) as well as his five U.S. patents registered officially by USPTO. His main research interests include optimizing, modeling, simulating, and synthesizing VLSI hardware for crypto and security computer arithmetic operations.