



Enhancing Imperceptibility: Zero-width Character-based Text Steganography for Preserving Message Privacy

Saqib Ishtiaq¹ , Naveed Ejaz² , Muhammad Usman Hashmi³ , Syed Imran Hussain Shah¹ 

¹ Department of Computing and Technology, Iqra University, Islamabad, Pakistan

² School of Computing, Queen's University, Canada

³ Department of Computer Science, Bahria University, Islamabad, Pakistan

Corresponding author: Saqib Ishtiaq (saqibishtiaq26507@gmail.com)

Editorial Record

First submission received:

December 27, 2024

Revisions received:

April 11, 2025

May 31, 2025

Accepted for publication:

June 1, 2025

Academic Editor:

Zdenek Smutny
Prague University of Economics
and Business, Czech Republic

This article was accepted for publication
by the Academic Editor upon evaluation of
the reviewers' comments.

How to cite this article:

Ishtiaq, S., Ejaz, N., Hashmi, M.U., & Shah,
S.I.H. (2025). Enhancing Imperceptibility:
Zero-width Character-based Text
Steganography for Preserving Message
Privacy. *Acta Informatica Pragensia*, 14(3),
445–459.
<https://doi.org/10.18267/j.aip.271>

Copyright:

© 2025 by the author(s). Licensee Prague
University of Economics and Business,
Czech Republic. This article is an open
access article distributed under the terms
and conditions of the [Creative Commons
Attribution License \(CC BY 4.0\)](https://creativecommons.org/licenses/by/4.0/).



Abstract

Background: Text steganography preserves the privacy of secret messages by hiding them in cover text. However, existing text steganography techniques embed messages by introducing distortions in text, reducing the similarity between the cover and stegotext.

Objective: The objective of this study was to design a method that increases the number of embedding choices and locations to hide more secret bits per distortion in the cover text. The goal is to enhance both embedding capacity and imperceptibility.

Methods: A text steganography method is proposed that uses eight zero-width characters (ZWCs) to embed secret messages in the cover text. The proposed method also treats every character in the cover text as a potential embedding location. With eight embedding choices and bit encoding based on embedding locations, more bits can be hidden with fewer insertions in cover text.

Results: Experimental results confirm that the proposed method embeds a greater number of bits per insertion of ZWC in the cover text. It also requires a smaller number of insertions to embed secret messages of comparable length. Consequently, the proposed method achieves higher embedding capacity and better imperceptibility compared to existing text steganography methods.

Conclusion: The proposed method presents a substantial improvement in text steganography by increasing embedding capacity per distortion and preserving high similarity between cover and stegotext, thus enabling more secure covert communication.

Index Terms

Information hiding; Zero-width character text steganography; Text watermarking; Text processing; Imperceptibility.

1 INTRODUCTION

The internet and modern communication devices have made communication faster and more convenient. However, they are also vulnerable to eavesdropping, a passive attack that intercepts information (Malik et al., 2017). Cryptography and information hiding are two security techniques used to guard against eavesdropping (Shearer & Gutmann, 1996). Cryptography ensures the confidentiality of secret messages by making them incomprehensible. Cryptography involves two main processes: encryption and decryption. During encryption, plain text is transformed into cipher text, which is not understandable (Ahmed et al., 2023), while decryption converts the cipher text into plain text. However, cryptography can raise suspicion due to the visible changes in the contents.

Unlike cryptography, information hiding involves concealing both the secret messages and their presence under a different medium, as explained by Liu et al. (2007). Common cover media include text, image, audio and video (Grothoff et al., 2005; Varghese & Sasikala, 2023). There are two main techniques for information hiding: watermarking and steganography. Watermarking embeds a watermark within the cover medium for purposes such as authentication, tamper detection and copyright protection (Iqbal et al., 2019; Khadam et al., 2021). Steganography, on the other hand, conceals the secret messages under a cover medium for covert communication (Askari et al., 2023; Johnson & Jajodia, 1998). Nowadays, steganography is also used to maintain the confidentiality of messages in communication between smart objects in IoT environments (Ray et al., 2021).

Our research is centred on text steganography, which involves using text as a cover to embed secret messages. Text is a commonly used mode of communication on the internet, making it an attractive option for steganography. Due to its low processing and memory requirements, text steganography can be used for security in IoT applications. However, compared to other cover media, text possesses less redundant data, which are typically exploited for steganography. As a result, concealing secret messages invisibly and imperceptibly within a text cover can be challenging (Al-Nofaie & Gutub, 2020).

Various techniques of text steganography are evaluated based on their capacity, invisibility, imperceptibility, robustness and security attributes. Capacity measures the amount of message concealed in a cover text (T_c). High capacity is desirable for text steganography methods. However, an increase in capacity may reduce invisibility, imperceptibility, robustness and security. Maintaining a balance between capacity and other attributes is challenging. Invisibility measures any visual artifacts generated in a cover text because of steganography. Imperceptibility refers to the degree to which the similarity between a cover text and a stegotext is maintained after embedding a secret message. Robustness is an ability that retains the hidden messages in a stegotext after attacks. Security ensures the undetectability of the hidden messages in a stegotext.

Most existing text steganography techniques conceal messages in the text by inserting extra characters, i.e., spaces (Por et al., 2012; Wu et al., 2022), zero-width characters (ZWC) (Ahvanooey et al., 2022; Al-Nofaie & Gutub, 2021), extension characters (Alanazi et al., 2022) etc. Insertion of extra characters increases the cover text length (L_c) and affects the cover text and stegotext similarity. The lack of similarity between the cover and stegotext compromises the imperceptibility and makes it suspicious. The aim of this research is to enhance imperceptibility by minimizing the number of insertions of extra characters in the cover text. In this paper, the cover text (T_c) is partitioned and one ZWC is inserted in each partition. The number of partitions required to hide a secret message depends on the cover text length and the secret message length. Along the type of ZWC, the location of insertion is also used to conceal secret message bits. Thus, more secret bits are embedded per distortion in the cover text and fewer insertions are required to embed a secret message. A smaller number of distortions maintains the similarity of cover text (T_c) with stegotext (T_s); hence, the imperceptibility is increased.

The key contributions of this paper are as follows.

- In the proposed method, the numbers of embedding choices and embedding locations are increased. The embedding locations in the cover text are also used to conceal the secret message bits. Due to the greater number of embedding choices and usage of the insertion location for steganography, more bits are embedded per insertion, increasing the capacity. Similarly, fewer insertions in the cover text generate highly imperceptible stegotext.
- In this method, the number of insertions required to conceal a secret message is not fixed but adapted based on the cover text length and the secret message length. Thus, a small message can be embedded with fewer insertions in the cover text than a large secret message.

The remainder of this paper is organized as follows: Section 2 reviews the existing text steganography literature. Section 3 discusses proposed algorithms. Section 4 reports on the experiments and their results. Finally, Section 5 provides the conclusions.

2 RELATED WORK

In this section, existing text steganography techniques are reviewed. Text steganography techniques are categorized into structural, random and statistical, linguistic and coverless text steganography.

2.1 Structural text steganography

Structural text steganography techniques alter the structure or format of the cover text for embedding a secret message. Some of these methods insert extra characters in cover text for embedding. Unicode has some width-less characters known as zero-width characters (ZWCs). These ZWCs are inserted in a cover text to embed the secret message (Ahvanooy et al., 2018; Ahvanooy et al., 2022; Al-Nofaie & Gutub, 2020; Al-Nofaie & Gutub, 2021).

Similarly, an extension character termed Kashida is inserted in Arabic text for steganography (Al-Nofaie et al., 2016; Alanazi et al., 2022). A secret message is embedded in space code steganography by inserting extra space codes (Gurunath & Samanta, 2023; Por et al., 2012). Null spaces are added in the cover text to authenticate the communication between lightweight IoT devices (Lee, 2019). Emotional icons are inserted in the cover text for steganography (Patiburn et al., 2017). Adding extra characters increases the cover text length and reduces imperceptibility. Inserting Kashida characters, emoticons and space codes may introduce visible changes in the cover text that reduce its invisibility. The text steganography methods that hide the secret message by inserting some extra characters in the cover text are empirically summarized in Table 1.

Some structural text steganography methods substitute one character with another character for steganography. The isolated and general Unicode of Arabic, Persian and Urdu alphabets are substituted for steganography (Alanazi et al., 2020). In the mixed-case font method, the English alphabet is substituted for steganography as depicted by Equation (1) (AminAli & Saad, 2013). The uppercase and lowercase English alphabets are C_i and c_i .

$$C_i | c_i = \begin{cases} c_i & \text{for } 0 \\ C_i & \text{for } 1 \end{cases} \quad (1)$$

Some characters have similar shapes but with different Unicode. These characters are known as homoglyphs. Homoglyphs are substituted for information hiding (Bertini et al., 2019; Rizzo et al., 2017). Due to the number of available embedding locations, substitution-based methods have low embedding capacity. The colour of the different cover text characters is changed intentionally to conceal a secret message (Malik et al., 2017; Thabit et al., 2022). However, such changes in the case and colour of a cover text for steganography are more detectable and reduce its invisibility.

Table 1. Empirical summary of insertion-based text steganography methods.

Method source	Characters	Capacity	Imperceptibility
(Al-Nofaie & Gutub, 2020)	Pseudo space	Insert up to 15 pseudo spaces to hide 4 bits	High
(Ahvanooy et al., 2018)	4 ZWCs	2 bits per insertion	High
(Por et al., 2012)	8 space codes	3 bits per insertion	Medium
(Al-Nofaie & Gutub, 2021)	Pseudo space	Insert up to 15 pseudo spaces to hide 4 bits	High
(Ahvanooy et al., 2022)	4 ZWCs	2 bits per insertion	High
(Alanazi, et al., 2022)	2 ZWCs and Kashida	1 bits per insertion	Medium
(Al-Nofaie et al., 2016)	Kashida	1 bits per insertion	Medium
(Gurunath & Samanta, 2023)	Space	Insert up to 3 spaces to hide 3 bits	Medium
(Patiburn et al., 2017)	36 emotions	1 character per insertion	Low

2.2 Random and statistical text steganography

Random and statistical text steganography conceals a secret message by generating random or natural-language text. The natural text is generated using the probability-based language model of a word sequence. The language model can be shown mathematically as follows.

$$p(S) = \prod_{j=1}^n p(w_j | w_1, \dots, w_{j-1}) \quad (2)$$

In Equation (2), $p(S)$ is the probability of the sentence S while $p(w_j | w_1, \dots, w_{j-1})$ is the probability of the j th word when $j-1$ words are given. Random characters are generated for the steganography (Elmahi et al., 2017). Long short-term memory (LSTM) generates a dialogue that carries a hidden message (Yang et al., 2018). The Markov chain model embeds the message by generating a natural language text (Shniperov et al., 2016; Wu et al., 2019; Wu et al., 2020). The text generated by these techniques may have low quality and feel unnatural. Fake email IDs are generated

that carry the secret message (Kumar et al., 2014; Kumar et al., 2016; Satir & Isik, 2012; Satir & Isik, 2014; Tutuncu & Abi Hassan, 2015). Insertion of fake email IDs raises suspicion.

2.3 Linguistic text steganography

Linguistic text steganography embeds a message by changing the syntax or semantics of a text. Abbreviations and complete forms of different words are substituted for embedding secret messages (Rafat, 2009; Shirali-Shahreza, 2007). Words with different spellings in US and UK English are utilized for steganography (Shirali-Shahreza, 2008). Synonym substitution is performed deliberately for embedding a secret message in a cover text (Huanhuan et al., 2017; Shirali-Shahreza, 2008). Synonym substitution text steganography is demonstrated in (3) below. W_i and S_i are the word and its synonym, respectively.

$$W_i = \begin{cases} W_i & \text{for } 0 \\ S_i & \text{for } 1 \end{cases} \quad (3)$$

These techniques are suitable for printed text. Due to the low number of embedding locations, these techniques have a low embedding capacity.

2.4 Coverless text steganography

Coverless text steganography techniques do not alter the original text to embed the secret message. Instead, these methods use inherent characteristics of the text for data hiding. For instance, the parity of strokes in Chinese characters is used to encode secret information (Wang & Gao, 2019). Similarly, English alphabets are categorized into four groups and for every two bits of the secret message, the position of a character from the corresponding group is appended to the stegokey (Kouser et al., 2016). The position of each secret character within the cover text is recorded in the stegokey, which is later used for message extraction at the receiver's end (Naqvi et al., 2018). An Arabic text having a first word with specific features is selected to represent the secret character (Rashid & Nasrawi, 2024). The location information of the secret message keyword in the Chinese text is encrypted using polynomial encryption and embedded within a forged URL (Guan et al., 2022). These methods often rely on large databases of cover text with appropriate features for embedding. Notably, these approaches face a significant challenge in securely transmitting the stegokey, as its length often exceeds that of the secret message itself.

3 DESCRIPTION OF PROPOSED METHOD

The proposed method embeds the secret message (M_s) in the cover text (T_c) by inserting different ZWCs in it. The proposed method consists of two phases, i.e., the embedding and extraction phases. Equations (4) and (5) depict the embedding function E and extraction function E^{-1} (Mansor et al., 2017).

$$E(T_c, M_s) = T_s \quad (4)$$

$$E^{-1}(T_s) = M_s \quad (5)$$

In Table 2, abbreviations used in this paper are given with their description.

3.1 Embedding phase

This phase embeds the secret message in the cover text by inserting the ZWCs. The embedding phase steps are given in Algorithm 1. Algorithm 1 takes three inputs, i.e., cover text (T_c), secret message (M_s) and key (K). The cover text (T_c), secret message (M_s) and key (K) are in English text form. Cover text (T_c) provides the cover to the secret message (M_s).

Table 2. Abbreviations along with their descriptions.

Abbreviation	Description
A_c	Average capacity
A_I	Average imperceptibility
A_R	Average modification ratio
C_t	Compression table

Abbreviation	Description
C_p	Capacity of one partition
C_n	Capacity of n partitions
K	Key
L_i	i^{th} embedding location
L_b	Compressed binary message length
L_c	Cover text length
L_s	Stegotext length
L_z	Number of elements in a set of zero width characters
m	Number of embedding locations in a partition
M_b	Compressed binary secret message
M_s	Secret message
n	Number of partitions
P_i	i^{th} partition
S_m	Secret message size in bits
S_s	Stegotext size in bits
T_c	Cover text
T_s	Stegotext
Z	Set of zero width characters

The key (K) is a pre-shared text between the communicating parties used to compress and decompress secret messages. The key enhances security as the eavesdropper cannot decompress the secret message without a key. The key is solely utilized for the compression and decompression processes and does not serve any role in the encryption or decryption of the secret message. Before embedding, a secret message is compressed, which reduces its size and enhances the embedding capacity. In the existing literature, Lempel–Ziv–Welch (LZW) (Satir & Isik, 2012), move to front (MTF), Burrows–Wheeler transform (BWT) (Kumar et al., 2014), arithmetic encoding (ARI), run length encoding (RLE) (Tutuncu & Abi Hassan, 2015) and Huffman encoding (Kumar et al., 2016; Satir & Isik, 2014; Thabit et al., 2022) are used for compression of a secret message. Kumar et al. (2016) mentioned that Huffman encoding has a better compression ratio. Thus, in the proposed method, a secret message is compressed using Huffman encoding. Huffman encoding generates the compression table (C_t) by using the key (K). This compression table is used for secret message (M_s) compression and to generate a compressed binary secret message (M_b). After secret message embedding, the cover text is transformed into stegotext (T_s) that carries a secret message. The embedding phase consists of two sub-phases.

3.1.1 Partitioning

In this phase, the cover text is divided into n non-overlapping partitions, i.e., $T_c = \{P_1, P_2, P_3, \dots, P_n\}$, each containing the same number of characters. Each character in a partition P_i is treated as an embedding location. The partition P_i consists of m embedding locations, i.e., $P_i = \{L_1, L_2, L_3, \dots, L_m\}$ and $m = \lfloor L_c/n \rfloor$. Thus, by using m locations in a partition P_i , $\log_2 m$ bits can be embedded. In each partition P_i , only one ZWC from the set Z is inserted, where Z is a set of zero width characters (ZWCs). ZWCs are width-less characters, so the insertion of these characters leaves no visual distortion in the cover text. In this method, Z consists of eight ZWCs listed along with their name and Unicode in Table 3.

Table 3. List of ZWCs used for steganography.

No.	Unicode	Name of ZWC
1	U+200B	Zero width space
2	U+200C	Zero width non-joiner
3	U+200D	Zero width joiner
4	U+200E	Left-to-right mark

No.	Unicode	Name of ZWC
5	U+202A	Left-to-right embedding
6	U+202C	Pop directional formatting
7	U+202D	Left-to-right override
8	U+180E	Mongolian vowel separator

If there are L_z ZWCs in the set Z , then $\log_2 L_z$ bits can be embedded by inserting a ZWC. The capacity of a partition (C_p) can be calculated by using Equation (6).

$$C_p = \log_2 m + \log_2 L_z \quad (6)$$

In Algorithm 1, the number of partitions n is decided in such a way that the capacity of the cover text by using n partitions (C_n) should be greater than or equal to the compressed binary secret message length (L_b). The compressed binary secret message is also divided into n partitions, i.e., $M_b = \{B_1, B_2, B_3, \dots, B_n\}$. The length of B_i is $\log_2 m + \log_2 L_z$.

3.1.2 Embedding

For each partition P_i of cover text, B_i is taken from the compressed binary secret message. The decimal representation of $\log_2 L_z$ most significant bits (MSB) of B_i , (Z_d) decides the ZWC ($z \in Z$) to be inserted in the P_i . Similarly, the decimal representation of $\log_2 m$ least significant bits (LSB) of B_i , (L_d) identifies the location of insertion of the ZWC in P_i . So a ZWC $Z[Z_d]$ is inserted at L_d location in P_i as mentioned in Equation (7).

$$P_i = \{L_1, L_2, L_3, \dots, L_{d-1}, Z[Z_d], L_{d+1}, \dots, L_m\} \quad (7)$$

In this way, $\log_2 m + \log_2 L_z$ bits are embedded by inserting only one ZWC in a partition.

Algorithm 1. Embedding algorithm.

Input: T_c, M_s, K

Output: T_s

1. $C_t \leftarrow \text{Huffman_Encoding}(K)$
2. $M_b \leftarrow \text{Compress}(M_s, C_t)$
3. $Z \leftarrow \{U+200B, U+200C, U+200D, U+200E, U+202A, U+202C, U+202D, U+180E\}$
4. $L_c \leftarrow \text{Length}(T_c)$
5. $L_z \leftarrow \text{Length}(Z)$
6. $L_b \leftarrow \text{Length}(M_b)$
7. $n \leftarrow 0$
8. *Do*
9. $n + +$
10. $m \leftarrow \lfloor L_c/n \rfloor$
11. $C_p \leftarrow \log_2 m + \log_2 L_z$
12. $C_n \leftarrow C_p \times n$
13. *While* $C_n < L_b$
14. Divide T_c into n partitions $\{P_1, P_2, P_3, \dots, P_n\}$ of size m
15. Divide M_b into n partitions $\{B_1, B_2, B_3, \dots, B_n\}$ of size C_p
16. For each P_i take B_i
17. $Z_b \leftarrow \text{Take MSB } \log_2 L_z \text{ bits from } B_i$
18. $Z_d \leftarrow \text{To_Decimal}(Z_b)$
19. $L_b \leftarrow \text{Take LSB } \log_2 m \text{ bits from } B_i$
20. $L_d \leftarrow \text{To_Decimal}(L_b)$
21. Insert $Z[Z_d]$ at L_d location i.e., $P_i = \{L_1, L_2, L_3, \dots, L_{d-1}, Z[Z_d], L_{d+1}, \dots, L_m\}$
22. *End For*

3.2 Extraction phase

Algorithm 2 retrieves the secret message (M_s) hidden in the stegotext (T_s) by tracking the occurrence of ZWCs. Algorithm 2 takes the stegotext (T_s) and the key (K) as input. The stegotext (T_s) is in the English language that carries

the secret message (M_s). The key (K) is the same text used to generate the compression table (C_t) in Algorithm 1. The pre-shared key (K) generates the compression table (C_t). The compressed binary secret message (M_b) is decompressed by using C_t . The output of this phase is a secret message (M_s) extracted from the stegotext (T_s). The extraction phase has the following sub phases.

3.2.1 Partitioning

In the embedding phase, one ZWC is inserted in each partition to hide $\log_2 m + \log_2 L_z$ secret bits. Thus, the number of partitions n is calculated by using Equation (8).

$$n = \sum_{i=1}^{L_z} F(z_i \in Z, T_s) \quad (8)$$

The function $F(.)$ counts the frequency of each ZWC z_i in the stegotext (T_s). As the embedding phase inserts n extra characters in the cover text (T_c), the cover text length (L_c) is obtained by taking the difference between the length of the stegotext (L_s) and the number of partitions (n). The partition length (m) of the cover text (T_c) is calculated by dividing L_c by n . Due to the insertion of ZWC, the length of each partition of the stegotext is increased by 1. The stegotext (T_s) is divided into n partitions, i.e., $T_c = \{P_1, P_2, P_3, \dots, P_n\}$. Each partition P_i consists of m embedding locations and one ZWC as mentioned in Equation (7).

3.2.2 Extraction

For a partition P_i , Z_d is a index of z in set Z , while L_d is a index of z in P_i . As Z_d and L_d hide $\log_2 L_z$ and $\log_2 m$ respectively, their binary representation Z_b and L_b are concatenated to B_i . Concatenation of $\{B_1, B_2, B_3, \dots, B_n\}$ generates the compressed binary secret message M_b , which is decompressed and the secret message M_s is obtained.

Algorithm 2. Extraction algorithm.

Input: T_s, K

Output: M_s

```

1.  $L_z \leftarrow \text{Length}(Z)$ 
2.  $n \leftarrow \sum_{i=1}^{L_z} F(z_i \in Z, T_s)$ 
3.  $L_s \leftarrow \text{Length}(T_s)$ 
4.  $L_c \leftarrow L_s - n$ 
5.  $m \leftarrow \lfloor L_c/n \rfloor$ 
6. Divide  $T_s$  into  $n$  partitions  $\{P_1, P_2, P_3, \dots, P_n\}$  of size  $m + 1$ 
7. For each partition  $P_i$ 
8.    $Z_d \leftarrow \text{Index\_of}(z \text{ in } Z)$ 
9.    $L_d \leftarrow \text{Index\_of}(z \in Z \text{ in } P_i \text{ partition})$ 
10.   $Z_b \leftarrow \text{To\_Binary}(Z_d)$ 
11.   $L_b \leftarrow \text{To\_Binary}(L_d)$ 
12.   $B_i \leftarrow \text{Concatenate}(Z_b, L_b)$ 
13. End For
14.  $M_b \leftarrow \text{Concatenate}(\{B_1, B_2, B_3, \dots, B_n\})$ 
15.  $C_t \leftarrow \text{Huffman\_Encoding}(K)$ 
16.  $M_s \leftarrow \text{Decompress}(M_b, C_t)$ 

```

4 EXPERIMENTS AND RESULTS

This section presents the experimental results. Python 3.6.5 and Spyder 3.2.8 IDE are used to implement the proposed method. The Dahuffman library is used to compress the secret message. Eight-bit binary representations of ASCII characters are used where applicable. The proposed method is demonstrated by using the benchmark cover text and secret message listed in Figure 1. These benchmark cover text and secret message are taken from Ahvanooey et al.(2022). For detailed experiments, a publicly available text dataset, "A Million News Headlines", is used (Kulkarni, 2022). This dataset consists of 1,226,259 headlines published by the Australian Broadcasting Corporation (ABC) in 18 years. On average, there are 6 to 7 words in each headline. For the experiments, 1000 random headlines with a length between 24 and 64 characters are selected. For each cover text, 40-bit and 80-bit secret messages are generated randomly. The proposed method performance is evaluated in terms of capacity and imperceptibility.

4.1 Capacity

Capacity measures the amount of message hidden in the cover text. Equation (9) is used to calculate the capacity of the proposed method by dividing the number of secret bits embedded in the cover text (S_m) by the size of the stegotext in bits (S_s) (Malik et al., 2017; Satir & Isik, 2014). Equation (10) measures the average capacity (A_c) of k cover text samples.

$$C = \frac{S_m}{S_s} \quad (9)$$

$$A_c = \frac{1}{k} \sum_{i=1}^k C_i \quad (10)$$

The capacity of the proposed method is compared with the state-of-the-art text steganography methods (Bertini et al., 2019; Ahvanooey et al., 2018; Ahvanooey et al., 2022; Al-Nofaie & Gutub, 2020; Al-Nofaie & Gutub, 2021). Average capacity is measured using Equation (10); results are reported in Table 4.

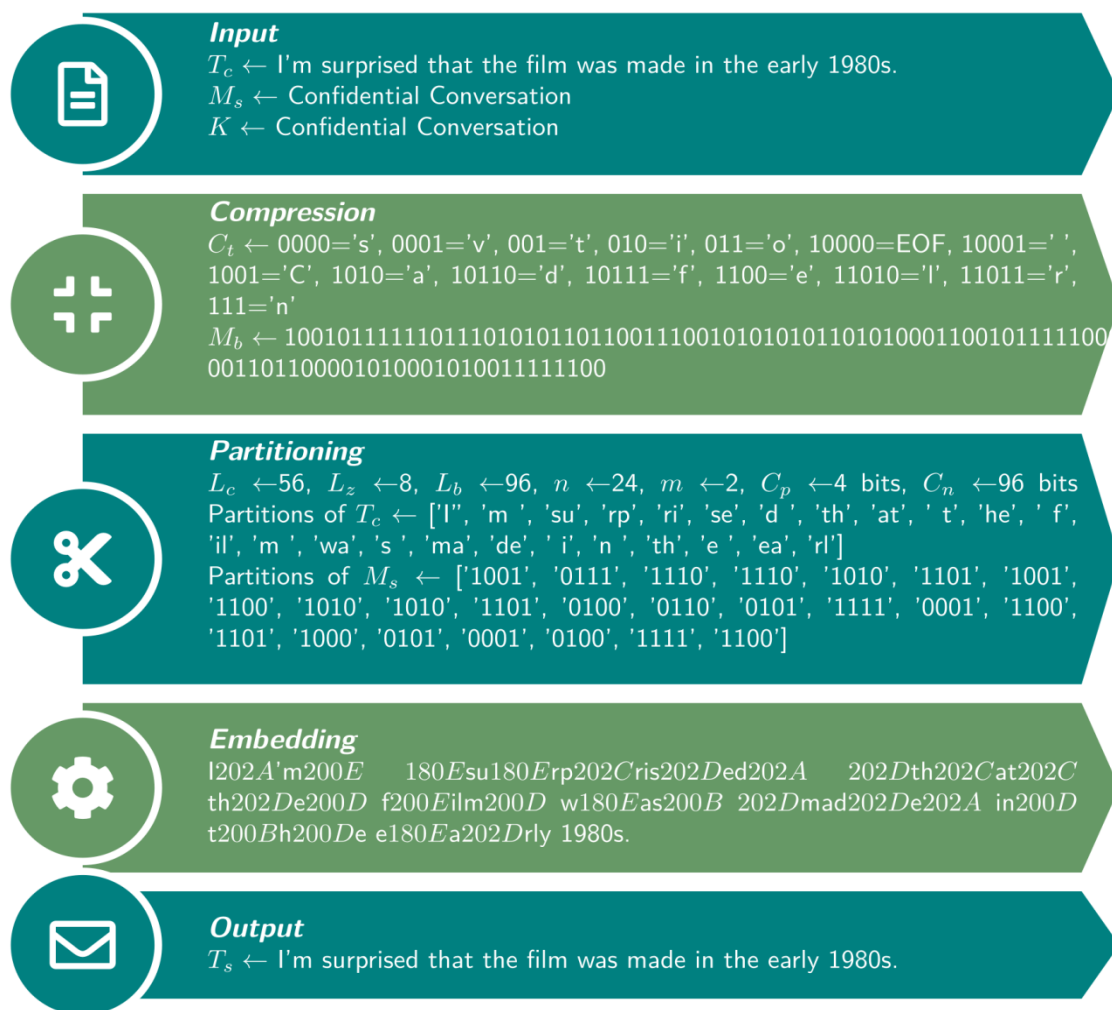
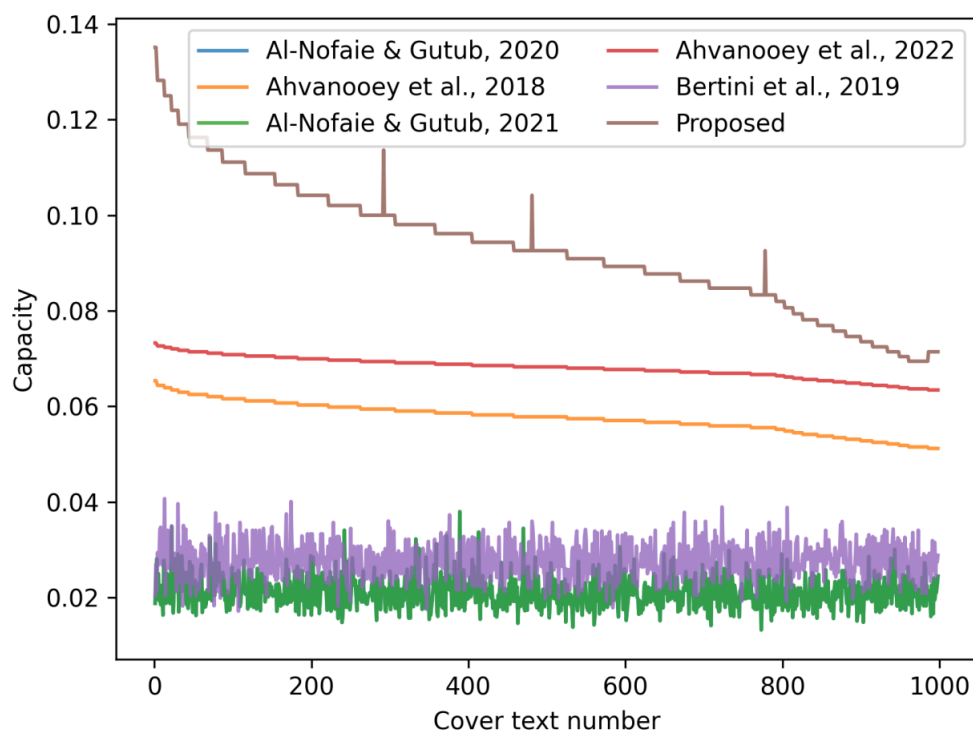


Figure 1. Example of embedding algorithm.

The capacity for each cover text for 40-bit and 80-bit secret messages is shown in Figures 2 and 3 respectively. The experimental results show that the proposed method has a higher embedding capacity than other methods. The proposed method achieved this improvement in capacity by compressing the secret message, using more embedding choices, i.e., 8 ZWCs, and utilizing the location of insertion for embedding.

Table 4. Comparison of capacity of proposed method and existing text steganography methods.

Message size	Al-Nofaie & Gutub (2020)	Ahvanooey et al. (2018)	Al-Nofaie & Gutub (2021)	Ahvanooey et al. (2022)	Bertini et al. (2019)	Proposed
40 bits	0.0211	0.0576	0.0211	0.0681	0.0276	0.0931
80 bits	0.0210	0.0673	0.0210	0.0710	0.0277	0.1466

**Figure 2.** Capacity comparison: proposed method vs. existing methods for 1000 cover text samples with a 40-bit secret message.

As claimed in the methodology section, the proposed method can hide the secret message with fewer insertions in the cover text, so an experiment is conducted to verify this statement. In this experiment, the average number of bits embedded per modification is calculated by using Equation (11). A_r is the average modification ratio, k is the total number samples of cover text, R_i is the modification ratio for the i^{th} sample; it is calculated by taking the ratio between the size of the secret message embedded (S_m) and the number of modifications (N_m) made in the text. The average modification ratio of the proposed and existing methods is reported in Table 5. The proposed method conceals the highest number of bits per insertion of ZWCs. Thus, fewer modifications are required to hide the secret message compared to the existing state-of-the-art text steganography methods.

$$A_r = \frac{1}{k} \sum_{i=1}^k R_i \quad (11)$$

$$R = \frac{S_m}{N_m} \quad (12)$$

Table 5. Comparison of average modification ratio of proposed method and existing text steganography methods.

Message size	Al-Nofaie & Gutub (2020)	Ahvanooey et al. (2018)	Al-Nofaie & Gutub (2021)	Ahvanooey et al. (2022)	Bertini et al. (2019)	Proposed
40 bits	1.205	1.176	1.205	0.5882	2.217	13.50
80 bits	2.399	1.250	2.399	0.9090	2.245	10.49

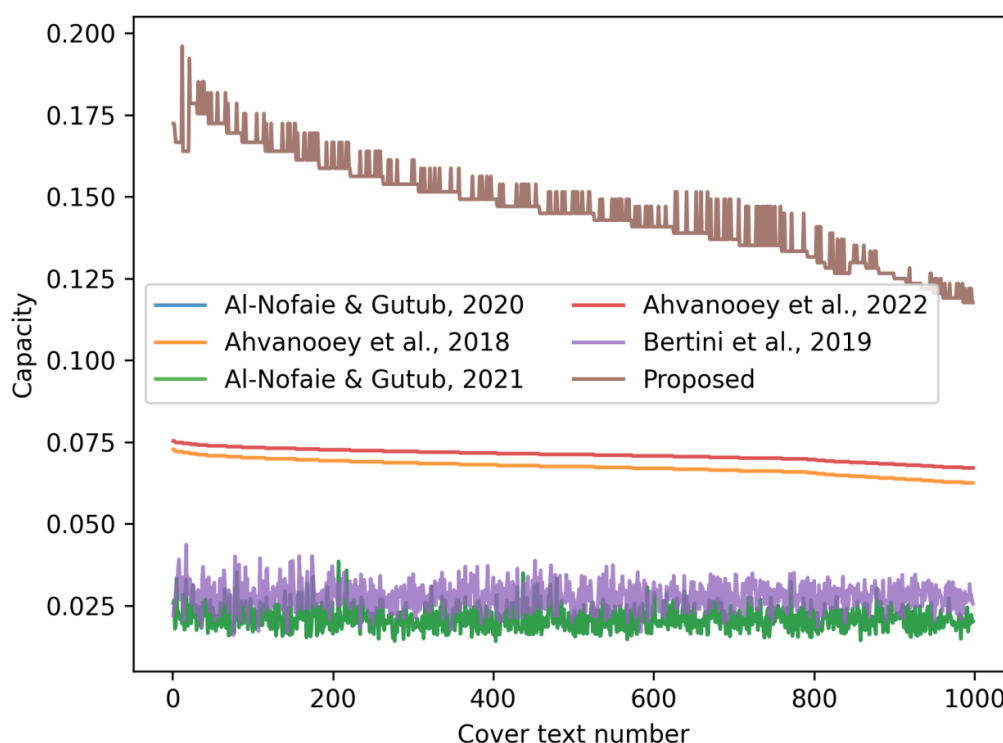


Figure 3. Capacity comparison: proposed method vs. existing methods for 1000 cover text samples with an 80-bit secret message.

4.2 Imperceptibility

Imperceptibility measures the similarity between the stegotext and the cover text. This section measures imperceptibility by calculating the Jaro-Winkler similarity of the stegotext (T_s) and the cover text (T_c). Jaro-Winkler similarity is calculated by using Equation (13) (Ahvanooy et al., 2022).

$$I = d_j + (p + l(1 - d_j)) \quad (13)$$

$$d_j = \frac{1}{3} \left[\frac{m}{|T_c|} + \frac{m}{|T_s|} + \frac{m-t}{|t|} \right] \quad (14)$$

$$A_I = \frac{1}{k} \sum_{i=1}^k I_i \quad (15)$$

I is the measure of imperceptibility determined using Jaro-Winkler similarity. The value of I ranges between 0 and 1. If the stegotext and cover text are identical, then I is 1 and 0 when the stegotext and cover text are dissimilar. Here, d_j is a Jaro distance, which measures the text difference by calculating the modifications generated by the steganography in the cover text; p is the scaling factor. The default value of p is 0.1; l is a common sub-string between the cover and stegotext; m is a number of the same characters, while t indicates the number of transpositions. The average imperceptibility (A_I) is calculated by using Equation (15); the results are reported in Table 6. The proposed method has the highest average imperceptibility (Bertini et al., 2019; Ahvanooy et al., 2018; Ahvanooy et al., 2022; Al-Nofaie & Gutub, 2020; Al-Nofaie & Gutub, 2021).

Table 6. Comparison of average imperceptibility of proposed method and existing text steganography methods.

Message size	Al-Nofaie & Gutub (2020)	Ahvanooy et al. (2018)	Al-Nofaie & Gutub (2021)	Ahvanooy et al. (2022)	Bertini et al. (2019)	Proposed
40 bits	0.8193	0.4091	0.8193	0.8799	0.8760	0.9848
80 bits	0.8189	0.1401	0.8189	0.8679	0.8758	0.9597

Imperceptibility for all the cover text samples for 40-bit and 80-bit secret messages is shown in Figures 4 and 5 respectively. For most of the samples, the proposed method has the highest imperceptibility. Only in the case of an

80-bit secret message does the method of Bertini et al., (2019) have a slightly higher imperceptibility for some cover text samples. However, the method of Bertini et al., (2019) achieves this improvement because it can hide fewer secret message bits than the proposed method.

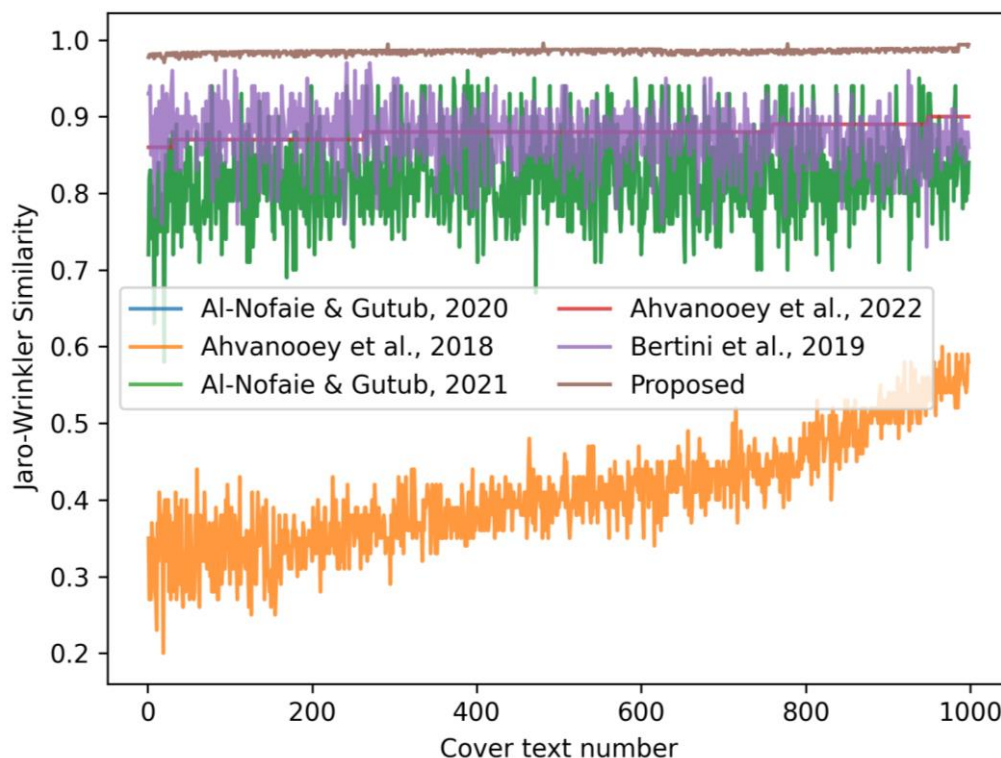


Figure 4. Imperceptibility comparison: proposed method vs. existing methods for 1000 cover text samples with a 40-bit secret message.

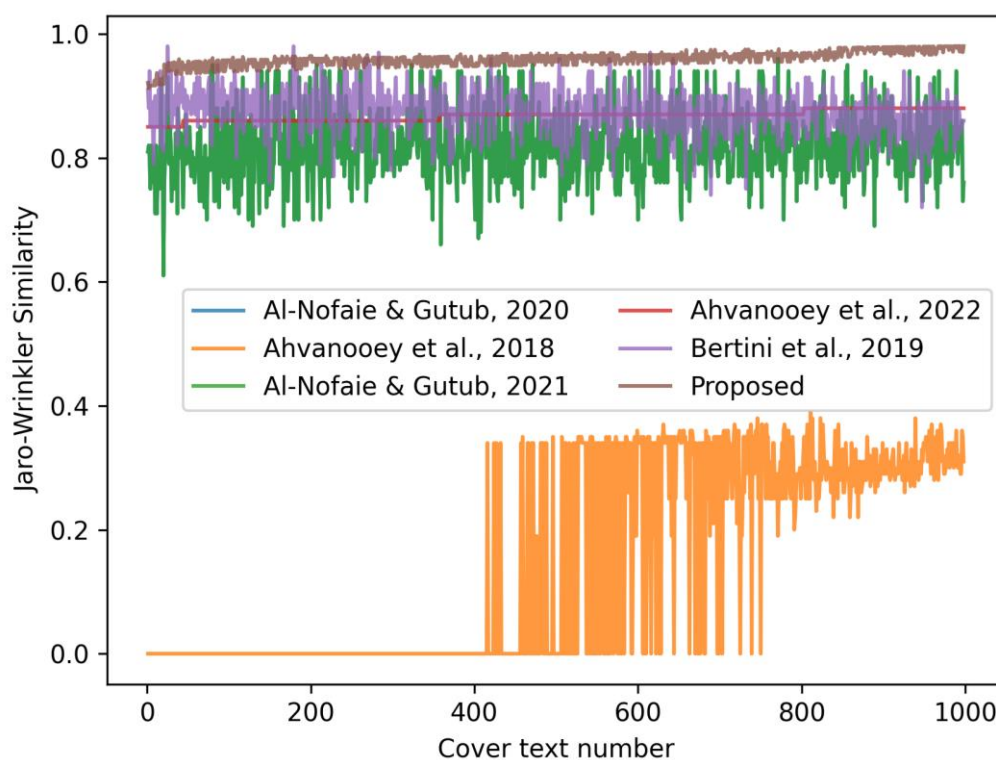


Figure 5. Imperceptibility comparison: proposed method vs. existing methods for 1000 cover text samples with an 80-bit secret message.

4.3 Adaptiveness verification

As discussed in the methodology section for the proposed method, the number of bits embedded per insertion of ZWC is adapted by using the cover text length and the secret message size. Thus, for a large cover text and a small secret message, fewer insertions are required for embedding than for a small cover text and a large secret message. To verify this statement, another experiment is conducted. In this experiment, randomly generated 40-bit and 80-bit secret messages are embedded in the cover text and the number of modifications done is counted. The results are shown in Figure 6. The results prove that when the cover text length is increased, fewer modifications are required compared to a short cover text for secret messages of the same size.

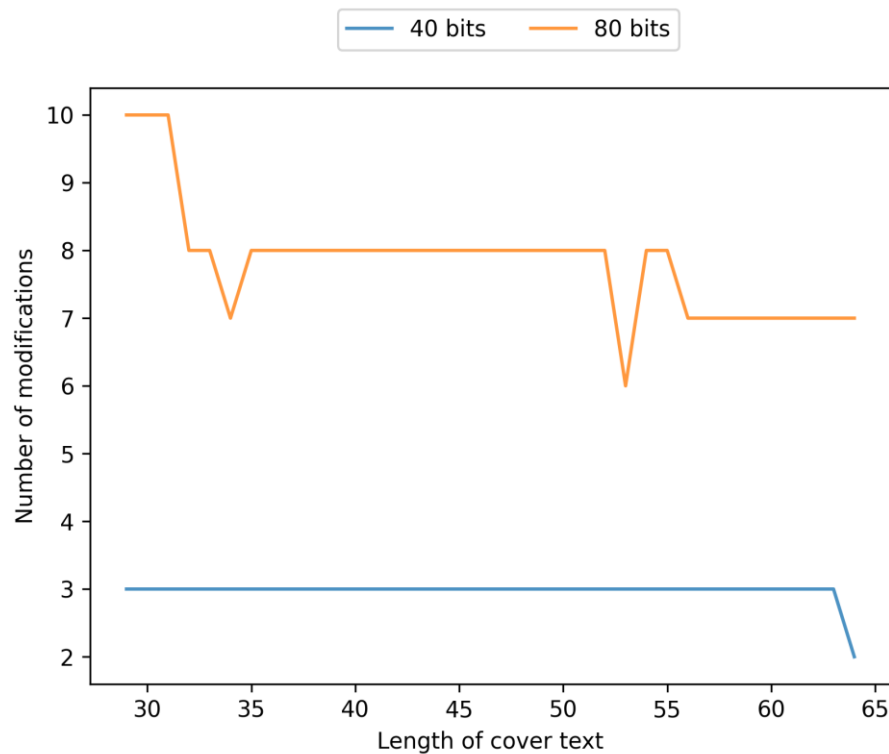


Figure 6. Adaptiveness verification of proposed method.

4.4 Analysis and discussion

In text steganography, the number of bits embedded per distortion depends on the number of embedding choices available. If the number of embedding choices is C , the capacity of the steganography method per distortion can be calculated by using Equation (16).

$$E_c = \log_2 C \quad (16)$$

The methods of Al-Nofaie & Gutub (2020) and Al-Nofaie & Gutub (2021) have only one embedding choice (C), so one bit is embedded per distortion. In these methods, to embed b bits at the same embedding location, up to 2^b insertions are made in the cover text. The methods of Ahvanooy et al. (2018) and Ahvanooy et al. (2022) have four embedding choices, so two secret message bits are concealed in the cover text per distortion. The method of Ahvanooy et al. (2018) expands the secret message characters to twelve bits, so six insertions of ZWCs are required to hide one character. The method of Bertini et al. (2019) has two embedding choices, so that it can hide one bit of secret message per homograph in the cover text. All these methods hide 1 or 2 bits of secret message per distortion in the cover text. These methods do not use embedding locations to carry the secret message. In the proposed method, we use the embedding locations to hide the secret message. If the number of embedding locations is E_l , the embedding capacity of the proposed method per distortion can be calculated by using Equation (17).

$$E_c = \log_2 C + \log_2 E_l \quad (17)$$

The proposed method uses eight embedding choices, so three bits can be embedded per insertion of ZWC. In the best case, when trying to hide a b -bit secret message, 2^{b-3} embedding locations (E_l) are available; then, the secret

message is embedded by inserting only one ZWC. In the worst case, when there is only one embedding location is available, then only one bit can be embedded by using this location. Therefore, the proposed method hides b bits per insertion in the best case and 4 bits per insertion (3 bits by using embedding choices + 1 bit by using location) in the worst case, which is higher than the existing methods. The experimental results also confirm the superiority of the proposed method in capacity and imperceptibility. As the proposed method hides more bits of the secret message per insertion of ZWC, fewer distortions are required to hide the same secret message compared to the existing methods. The major drawback of this method is that even minimal distortion can introduce detectable patterns or anomalies in the cover text, which may be exploited by eavesdroppers to reveal the secret message.

5 CONCLUSIONS

In this paper, a highly imperceptible text steganography method was proposed. This method hides the secret message via the insertion of ZWCs. An increase in embedding locations and choices enhances the capacity and reduces the required number of insertions of ZWCs. A simulator was implemented in Python to validate the proposed method and experiments were performed using benchmark secret messages and cover text samples. The experiments confirmed that the proposed method conceals the secret message with fewer distortions in cover text. Thus, the stegotext has high similarity in the cover text. Future work will study the efficiency of text steganography in the IoT environment to maintain confidentiality of messages between devices.

ADDITIONAL INFORMATION AND DECLARATIONS

Conflict of Interests: The authors declare no conflict of interest.

Author Contributions: S.I.: Conceptualization, Methodology, Software, Writing – Original draft, Writing – Reviewing and Editing. N.E.: Supervision, Validation. M.U.H.: Supervision, Validation, Writing – Reviewing and Editing. S.I.H.S.: Writing – Reviewing and Editing.

Statement on the Use of Artificial Intelligence Tools: The authors declare that they didn't use artificial intelligence tools for text or other media generation in this article.

Data Availability: The data that support the findings of this study are openly available in Kaggle at <https://www.kaggle.com/datasets/therohk/million-headlines>.

REFERENCES

- Ahmed, A., Iqbal, M. M., Jabbar, S., Ibrar, M., Erbad, A., & Song, H. (2023). Position-Based Emergency Message Dissemination Schemes in the Internet of Vehicles: A review. *IEEE Transactions on Intelligent Transportation Systems*, 24(12), 13548–13572. <https://doi.org/10.1109/tits.2023.3304127>
- Ahvanooy, M. T., Li, Q., Hou, J., Mazraeh, H. D., & Zhang, J. (2018). Aitsteg: An innovative text steganography technique for hidden transmission of text message via social media. *IEEE Access*, 6, 65981–65995. <https://doi.org/10.1109/access.2018.2866063>
- Ahvanooy, M. T., Zhu, M. X., Mazurczyk, W., Li, Q., Kilger, M., Choo, K. R., & Conti, M. (2022). CovertSYS: A systematic covert communication approach for providing secure end-to-end conversation via social networks. *Journal of Information Security and Applications*, 71, 103368. <https://doi.org/10.1016/j.jisa.2022.103368>
- Alanazi, N., Khan, E., & Gutub, A. (2020). Functionality-improved arabic text steganography based on unicode features. *Arabian Journal for Science and Engineering*, 45, 1037–11050. <https://doi.org/10.1007/s13369-020-04917-5>
- Alanazi, N., Khan, E., & Gutub, A. (2022). Inclusion of unicode standard seamless characters to expand arabic text steganography for secure individual uses. *Journal of King Saud University-Computer and Information Sciences*, 34(4), 1343–1356. <https://doi.org/10.1016/j.jksuci.2020.04.011>
- Al-Nofaie, S. M., Fattani, M. M., & Gutub, A. A. (2016). Merging two steganography techniques adjusted to improve Arabic text data security. *Journal of Computer Science & Computational Mathematics*, 59–65. <https://doi.org/10.20967/jcscm.2016.03.004>
- Al-Nofaie, S. M. A., & Gutub, A. A.-A. (2020). Utilizing pseudo-spaces to improve arabic text steganography for multimedia data communications. *Multimedia Tools and Applications*, 79, 19–67. <https://doi.org/10.1007/s11042-019-08025-x>
- Al-Nofaie, S., Gutub, A., & Al-Ghamdi, M. (2021). Enhancing arabic text steganography for personal usage utilizing pseudo-spaces. *Journal of King Saud University - Computer and Information Sciences*, 33(8), 963–974. <https://doi.org/10.1016/j.jksuci.2019.06.010>
- AminAli, A., & Saad, A. S. (2013). New Text Steganography Technique by using Mixed-Case Font. *International Journal of Computer Applications*, 62(3), 6–9. <https://doi.org/10.5120/10058-4650>
- Askari, M., Mahmood, A., & Iqbal, Z. (2023). A novel font color and compression text steganography technique. In *International Conference on Communication, Computing and Digital Systems (C-CODE)*, (pp. 1-6). IEEE. <https://doi.org/10.1109/c-code58145.2023.10139867>

- Bertini, F., Rizzo, S.G., & Montesi, D. (2019). Can information hiding in social media posts represent a threat? *Computer*, 52(10), 52–60. <https://doi.org/10.1109/mc.2019.2917199>
- Elmah, M.Y., & Sayed, M. (2017). Text steganography using compression and random number generators. *International Journal of Computer Applications Technology and Research*, 6(6), 259–263. <https://doi.org/10.7753/ijcatr0606.1005>
- Guan, B., Gong, L., & Shen, Y. (2022). A novel coverless text steganographic algorithm based on polynomial encryption. *Security and Communication Networks*, 2022(1), 1153704. <https://doi.org/10.1155/2022/1153704>
- Grothoff, C., Grothoff, K., Alkhutova, L., Stutsman, R., & Atallah, M. (2005). Translation based steganography. In *Information Hiding: 7th International Workshop*, (pp. 219–233). Springer. https://doi.org/10.1007/11558859_17
- Gurunath, R., & Samanta, D. (2023). A new 3-bit hiding covert channel algorithm for public data and medical data security using format-based text steganography. *Journal of Database Management*, 34(2), 1–22. <https://doi.org/10.4018/jdm.324076>
- Huanhuan, H., Xin, Z., Weiming, Z., & Nenghai, Y. (2017). Adaptive text steganography by exploring statistical and linguistical distortion. In *IEEE Second International Conference on Data Science in Cyberspace*, (pp. 145–150). IEEE. <https://doi.org/10.1109/dsc.2017.16>
- Iqbal, M.M., Khadam, U., Han, K.J., Han, J., & Jabbar, S. (2019). A robust digital watermarking algorithm for text document copyright protection based on feature coding. In *15th International Wireless Communications & Mobile Computing Conference*, (pp. 1940–1945). IEEE. <https://doi.org/10.1109/iwcmc.2019.8766644>
- Johnson, N.F., & Jajodia, S. (1998). Exploring steganography: Seeing the unseen. *Computer*, 31(2), 26–34. <https://doi.org/10.1109/mc.1998.4655281>
- Khadam, U., Iqbal, M.M., Jabbar, S., & Shah, S.A. (2021). Data aggregation and privacy preserving using computational intelligence. *IEEE Internet of Things Magazine*, 4(2), 60–64. <https://doi.org/10.1109/iotm.0001.2000010>
- Kouser, S. K., Khan, A., & Qamar, E. (2016). A Novel Content-Based Feature Extraction Approach: Text Steganography. *International Journal of Computer Science and Information Security*, 14(12), 916–922.
- Kulkarni, R. (2022). A million news headlines. Kaggle. Retrieved from <https://www.kaggle.com/datasets/therohk/million-headlines>
- Kumar, R., Chand, S., & Singh, S. (2014). An email based high capacity text steganography scheme using combinatorial compression. In *The 5th International Conference confluence the Next Generation Information Technology Summit*, (pp. 336–339). IEEE. <https://doi.org/10.1109/confluence.2014.6949231>
- Kumar, R., Malik, A., Singh, S., & Chand, S. (2016). A high capacity email based text steganography scheme using huffman compression. In *The 3rd International Conference on Signal Processing and Integrated Networks*, (pp. 53–56). IEEE. <https://doi.org/10.1109/spin.2016.7566661>
- Lee, W. (2019). Ultra-light mutual authentication scheme based on text steganography communication. *Journal of The Korea Society of Computer and Information*, 24(4), 11–18. <https://doi.org/10.15242/jie.e1014068>
- Liu, T.-Y., & Tsai, W.-H. (2007). A new steganographic method for data hiding in microsoft word documents by a change tracking technique. *IEEE Transactions on Information Forensics and Security*, 2(1), 24–30. <https://doi.org/10.1109/tifs.2006.890310>
- Malik, A., Sikka, G., & Verma, H.K. (2017). high capacity text steganography scheme based on lzw compression and color coding. *Engineering Science and Technology, an International Journal*, 20 (1), 72–79. <https://doi.org/10.1016/j.jestch.2016.06.005>
- Mansor, F.Z., Mustapha, A., & Samsudin, N.A. (2017). Researcher's perspective of substitution method on text steganography. *IOP Conference Series: Materials Science and Engineering*, 226, 012092. <https://doi.org/10.1088/1757-899x/226/1/012092>
- Naqvi, N., Abbasi, A. T., Hussain, R., Khan, M. A., & Ahmad, B. (2018). Multilayer partially homomorphic encryption text steganography (MLPHE-TS): A zero steganography approach. *Wireless Personal Communications*, 103, 1563–1585. <https://doi.org/10.1007/s11277-018-5868-1>
- Patiburn, S.A., Iranmanesh, V., & Teh, P.L. (2017). Text steganography using daily emotions monitoring. *International Journal of Education and Management Engineering*, 7(3), 1–14. <https://doi.org/10.5815/ijeme.2017.03.01>
- Por, L.Y., Wong, K., & Chee, K.O. (2012). Unispach: A text-based data hiding method using unicode space characters. *Journal of Systems and Software*, 85(5), 1075–1082. <https://doi.org/10.1016/j.jss.2011.12.023>
- Rafat, K. (2009). Enhanced text steganography in SMS. In *The 2nd International Conference on Computer, Control and Communication*, (pp. 1–6). IEEE. <https://doi.org/10.1109/ic4.2009.4909228>
- Rashid, S. H., & Nasrawi, D. A. (2024). Coverless Text Information Hiding Based on Built-in Features of Arabic Scripts. *Journal of Applied Data Sciences*, 5(2), 653–667. <https://doi.org/10.47738/jads.v5i2.243>
- Ray, A.M., Sarkar, A., Obaid, A.J., & Pandiaraj, S. (2021). IoT Security Using Steganography. In *Multidisciplinary Approach to Modern Digital Steganography*. IGI Global. <https://doi.org/10.4018/978-1-7998-7160-6.ch009>
- Rizzo, S.G., Bertini, F., Montesi, D., & Stomeo, C. (2017). Text watermarking in social media. In *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, (pp. 208–211). ACM. <https://doi.org/10.1145/3110025.3116203>
- Satir, E., & Isik, H. (2014). A huffman compression based text steganography method. *Multimedia tools and applications*, 70, 2085–2110. <https://doi.org/10.1007/s11042-012-1223-9>
- Satir, E., & Isik, H. (2012). A compression-based text steganography method. *Journal of Systems and Software*, 85(10), 2385–2394. <https://doi.org/10.1016/j.jss.2012.05.027>
- Shearer, J., & Gutmann, P. (1996). Government, cryptography, and the right to privacy. *Journal of Universal Computer Science*, 2 (3), 113–146. <https://doi.org/10.3217/jucs-002-03-0113>
- Shirali-Shahreza, M.H., & Shirali-Shahreza, M. (2008). A new synonym text steganography. In *International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, (pp. 1524–1526). <https://doi.org/10.1109/iih-msp.2008.6>

- Shirali-Shahreza, M., & Shirali-Shahreza, M.H.** (2007). Text steganography in SMS. In *International Conference on Convergence Information Technology*, (pp. 2260–2265). IEEE. <https://doi.org/10.1109/iccit.2007.100>
- Shirali-Shahreza, M.** (2008). Text steganography by changing words spelling. In *The 10th International Conference on Advanced Communication Technology*, (pp. 1912–1913). IEEE. <https://doi.org/10.1109/icact.2008.4494159>
- Shniperov, A. N., & Nikitina, K. A.** (2016). A text steganography method based on Markov chains. *Automatic Control and Computer Sciences*, 50(8), 802–808. <https://doi.org/10.3103/s0146411616080174>
- Thabit, R., Udzir, N.I., Yasin, S.M., Asmawi, A., & Gutub, A.A.-A.** (2022). CSNTSteg: Color spacing normalization text steganography model to improve capacity and invisibility of hidden data. *IEEE Access*, 10, 65439–65458. <https://doi.org/10.1109/access.2022.3182712>
- Tutuncu, K., & Abi Hassan, A.** (2015). New approach in e-mail based text steganography. *International Journal of Intelligent Systems and Applications in Engineering*, 3(2), 54–57. <https://doi.org/10.18201/ijisae.05687>
- Varghese, F., & Sasikala, P.** (2023). A detailed review based on secure data transmission using cryptography and steganography. *Wireless Personal Communications*, 129(4), 2291–2318. <https://doi.org/10.1007/s11277-023-10183-z>
- Wang, K., & Gao, Q.** (2019). A coverless plain text steganography based on character features. *IEEE Access*, 7, 95665–95676. <https://doi.org/10.1109/ACCESS.2019.2929123>
- Wu, D.-C., & Hsu, Y.-T.** (2022). Authentication of line chat history files by information hiding. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 18(1), 1–23. <https://doi.org/10.1145/3474225>
- Wu, N., Shang, P., Fan, J., Yang, Z., Ma, W., & Liu, Z.** (2019). Coverless text steganography based on maximum variable bit embedding rules. *Journal of Physics Conference Series*, 1237(2), 022078. <https://doi.org/10.1088/1742-6596/1237/2/022078>
- Wu, N., Yang, Z., Yang, Y., Li, L., Shang, P., Ma, W., & Liu, Z.** (2020). STBS-Stega: Coverless text steganography based on state transition-binary sequence. *International Journal of Distributed Sensor Networks*, 16(3), 155014772091425. <https://doi.org/10.1177/1550147720914257>
- Yang, Z., Zhang, P., Jiang, M., Huang, Y., & Zhang, Y.-J.** (2018). Rits: Real-time interactive text steganography based on automatic dialogue model. In *International Conference on Cloud Computing and Security*, (pp. 253–264). Springer. https://doi.org/10.1007/978-3-030-00012-7_24