SEMARAK ILMU PUBLISHING

Semarak International Journal of Machine Learning

Journal homepage:
https://semarakilmu.my/index.php/sijml/index
ISSN: 3030-5241

Feature-Based Method in Text Steganography: Evaluating Single-Bit and Dual-Bit Techniques for Secure Hidden Message

Sunariya Utama[1], Roshidi Din[1,*], Fazli Azzali[1], Azizi Abas[1], Hrudaya Kumar Tripathy[2]

[1] School of Computing UUM College Arts and Sciences, Universiti Utara Malaysia, 06010, Sintok, Kedah, Malaysia
[2] Kalinga Institute of Industrial Technology, Bhubaneswar, Odisha, India

ARTICLE INFO

Article history:
Received 23 September 2025
Received in revised form 23 October 2025
Accepted 20 November 2025
Available online 8 December 2025

Keywords:
Text steganography; feature-based technique; single-bit technique; dual-bit technique; embedding capacity size bit

ABSTRACT

Steganography, the art of concealing information within other non-secret data, has gained prominence as a method for secure communication. Among its various forms, text steganography particularly feature-based techniques offers a discreet approach by embedding messages within the structural characteristics of text. However, challenges persist in balancing data capacity, security, and imperceptibility. This paper aims to analyze and compare four feature-based text steganography techniques: two single-bit methods Change Alphabet Letter Pattern (CALP) and One-Flow-1-bit and two dual-bit methods One-Flow-2-bit and QUAD. Each technique utilizes distinct visual and structural features of English letters. This paper employs a comparative methodology, evaluating the techniques based on letter usage efficiency and embedding capacity. Results indicate that while single-bit methods like CALP offer simplicity and precise embedding for short texts, they are limited in capacity. In contrast, dual-bit techniques, particularly QUAD, demonstrate superior data embedding capabilities, making the technique suitable for scenarios requiring higher security and larger data concealment. However, increased complexity in dual-bit methods may impact processing efficiency. The study concludes that selecting an appropriate technique depends on the specific requirements of the application, such as the volume of data to be hidden and the desired level of covertness.

1. Introduction

The Steganography is the practice of concealing messages within various types of data, making the hidden information invisible to both human and machine detection [1], [2]. This implementation has applications in personal communication, security systems, and the protection of confidential information, making it valuable for businesses, the military, governments, and other entities [3], [4]. Steganography can be broadly categorized into two main types: natural language steganography and digital steganography. Natural language steganography itself is divided into linguistic steganography, which conceals hidden messages by manipulating linguistic rules [5], [6], and text steganography,

* *Corresponding author.*
E-mail address: roshidi@uum.edu.my

https://doi.org/10.37934/simjl.7.1.1830

which hides messages by altering text elements such as lines, words, spaces, and other text features [7], [8], [9].

Text steganography, a subfield of information security, is the practice of concealing information within text such that its presence is undetectable [10]. Unlike cryptography, which obfuscates the content of the message, steganography hides the existence of the message itself [11], [12], [13]. The growing need for secure communication in various domains ranging from military to private sector communications has led to the development of numerous steganographic techniques. This paper focuses on a specific subset of these methods, namely single-bit and dual-bit feature-based text steganography techniques, which manipulate the visual or structural features of alphabetic characters to encode binary data.

The motivation in implementation of the method is the continuous evolution of data hiding techniques in response to the increasing complexity of detection and extraction methods. As challengers develop more advanced tools to detect steganographic content, it becomes crucial to innovate and refine the methods used to embed hidden messages within cover texts [14], [15], [16]. This paper examines four techniques that represent different approaches to single-bit and dual-bit of feature-based text steganography: the Change Alphabet Letter Pattern (CALP) and One-Flow-1-bit as the single-bit technique, then One-Flow-2-bit, and QUAD as the dual bit-bit technique techniques [17], [18], [19], [20]. Each technique uses unique characteristics of English letters, such as their curvature, stroke continuity, and presence of lines, to embed binary data [21][22].

In single-bit technique that embeds one binary bits in cover text that using only 0 or 1 bit. CALP in single-bit technique that suited for scenarios where the cover text is short, and the need for concealment is moderate. For instance, CALP might be used in short, informal communications where a small amount of data needs to be hidden quickly and with minimal effort [17]. Then, the One-Flow-1-bit technique extends the principles of CALP by categorizing letters based on their writability in a single continuous sentence. One of the key advantages of this technique is its intuitive use of visual characteristics, which makes it more versatile in diverse text scenarios. Unlike CALP, which relies on specific letters, the One-Flow-1-bit technique can be applied to a broader range of texts, as it is less dependent on the occurrence of specific characters [18]. This versatility enhances the technique's applicability across different types of cover texts, from formal documents to casual messages.

However, like CALP, the One-Flow-1-bit technique is still somewhat limited by its single-bit encoding approach. While it offers better data capacity than CALP, it cannot match the efficiency of dual-bit techniques like One-Flow-2-bit or QUAD. Therefore, it is best suited for medium-level security needs where the embedding capacity is not the primary concern, but the technique still requires a certain degree of stealth [4], [23]. Building on the One-Flow-1-bit technique, the One-Flow-2-bit method introduces dual-bit encoding, significantly increasing the amount of data that can be concealed within the text. By categorizing letters into four groups based on their writability in one flow and the presence of vertical or horizontal lines, the One-Flow-2-bit technique achieves a more sophisticated level of data embedding [18]. This dual-bit approach allows for the representation of binary data in two bits per character, effectively doubling the data capacity compared to single-bit techniques.

The use of additional categorization criteria, such as the presence of lines, adds complexity to the encoding process but also increases the robustness of the steganographic technique. This added complexity makes it harder for adversaries to detect the hidden data, as the patterns are less obvious and more diverse [24]. The One-Flow-2-bit technique is particularly well-suited for scenarios where both high data capacity and moderate to high security are required. For example, it could be used in the secure transmission of longer messages or documents where the presence of hidden data must remain undetected.

The QUAD technique that categorizes letters based on a combination of structural features, such as the presence of lines and curves, to represent dual-bit binary data. This technique's nuanced approach to letter categorization enables it to achieve the highest data capacity among the methods analysed, making it ideal for applications that require the concealment of large amounts of data within a text. By utilizing a broad range of letters and structural features, it can be applied to a wide variety of texts without compromising the natural appearance of the cover text [19].

This paper aims to provide a comprehensive analysis of these four techniques, comparing their strengths and weaknesses in terms of letter used and embedding capacity ratio. The discussion of this paper explore how these methods can be applied in various contexts, their potential vulnerabilities, and areas where further research is needed. By understanding the intricacies of each technique, researchers and practitioners can make more informed decisions about which method to use in specific scenarios, ultimately contributing to the development of more secure and effective steganographic systems. Feature-based methods in text steganography involve concealing information by embedding it within various attributes or qualities of a text document [25]. The primary goal of these methods is to hide the existence of the hidden message in such a way that it remains inconspicuous to unintended recipients [26], [27]. Several researcher effort about feature-based techniques, ranging from single-bit to multi-bit embedding methods, each with its unique approach to hiding messages in text [21],[28], [29].

One of the simplest forms of feature-based steganography is the single-bit technique. Bajaj and Aggarwal [30] utilized this approach in their work on text steganography within HTML web pages. They proposed embedding a hidden message by converting it into binary bits, which are then encoded using HTML tags. Specifically, a binary '1' bit is embedded using the sequence (u, i, /i, /u), and a '0' bit using (i, u, /u, /i). This method capitalizes on the inconspicuous nature of HTML tags, which are rarely scrutinized in the source code of web pages, thus providing a simple yet effective means of concealing information.

Moving beyond single-bit techniques, researchers have also explored dual-bit and other multi-bit methods to enhance the capacity and robustness of steganography. Alsaadi *et al.* [31] developed a dual-bit technique that leverages the font color in Microsoft Excel cells. In this method, hidden messages are first converted into binary bits, which are then mapped to decimal numbers corresponding to RGB color values. The technique allows for a higher capacity of hidden data due to the wide range of colors available in Excel, making it a robust solution for data hiding in spreadsheets.

Wu *et al.* [22] introduced a coverless steganography technique that further advanced multi-bit methods. This approach converts the hidden message into binary bits and searches for corresponding tags in a binary sequence. The method ensures high security and is particularly resistant to attacks since it does not rely on traditional cover text but rather on the inherent structure of the English language, making it difficult to detect.

Naharuddin [28] proposed a feature-based technique that involves mapping binary bits using the ASCII table to conceal hidden messages. In this method, the binary bits of the hidden message are mapped by adding numbers from 1 to 7, which are then used to determine the specific row and column positions for embedding the text. This approach is particularly effective because it focuses on the positions within the text (based on rows and columns) rather than relying solely on the length of the cover text. As a result, this technique offers high-capacity performance, enabling a larger amount of data to be embedded discreetly within the text/

Kouser *et al.* [32] presented another innovative approach using dual category system based on binary bits. The technique divides hidden messages into two categories: the first embeds '0' bits in certain letters and '1' bits in others, while the second category uses consonants and vowels to

represent different bit values. This multi-bit technique enhances the embedding capacity by utilizing the structural properties of letters, allowing for more complex and robust hidden messages.

Additionally, more exploring techniques on multi-bit feature-based methods have been explored in different contexts. Reddy *et al.* [33] explored the use of Deoxyribonucleic Acid (DNA) steganography, where four DNA nucleotides are used to represent binary numbers (00, 01, 10, and 11). This method involves converting the hidden message into a DNA sequence using a lookup table, which is then translated into readable text. To decrypt the concealed message, a rule sequence table search is performed alongside the DNA sequence. This technique effectively embeds the hidden message in binary form within the DNA sequence, offering a unique and secure approach to data hiding.

Saad and Algamdi [34] introduced a feature-based method that categorizes binary bits by mapping them to specific letters (from A to G) using ASCII characters. This technique embeds the hidden message within the cover text by determining the row and column positions based on the ASCII values of the characters. The method is designed to provide robust performance and a large capacity for hiding messages, making it an effective solution for secure text communication.

This paper focuses on single-bit and dual-bit techniques for several reasons. Firstly, these methods strike a balance between simplicity and effectiveness. Single-bit techniques are relatively straightforward to implement and can be highly effective in scenarios where only a small amount of data needs to be concealed. They also offer the advantage of minimal impact on the cover text, making the hidden message less detectable. Dual-bit techniques, on the other hand, provide a higher capacity for data embedding while maintaining a reasonable level of simplicity. It particularly useful in applications where there is a need to hide more substantial amounts of data without compromising the integrity of the cover text. By concentrating on these techniques, this paper aims to explore the foundational methods of feature-based steganography and their potential applications in secure communication.

2. Methodology

For The methods section of this paper describes the processes used to encode and decode hidden messages using the four text steganography techniques discussed: CALP, One-Flow-1-bit, One-Flow-2-bit, and QUAD. Each technique involves a unique approach to manipulating the visual or structural features of alphabetic characters to represent binary data. The following subsections provide detailed descriptions of how each technique works, including the specific steps involved in the encoding and decoding processes.

2.1 Single-Bit Technique

Single-bit text steganography techniques embed one bit of binary data per character by altering specific features of letters, such as the presence of dots or the ability to be written in one continuous stroke. These methods are straightforward and suitable for scenarios where the cover text is short and moderate data concealment is needed that is shown in Table 1.

Table 1
Single-bit technique in feature-based method

ID	Bit	One-Flow-2-bit		QUAD	
		Group	**Letter used**	**Group**	**Letter used**
A	0	Letter with dot	“i, j”	Letters not writable in one flow	“A, B, D, E, F, H, K, Q, R, T, X”
B	1	Chosen letter	“a, A, c”	Letters writable in one flow	“C, G, I, J, L, H, M, N, O, P, R, S, U, V, W, Y, Z”

In Table 1 shows the CALP technique that utilizes a small subset of the English alphabet to represent binary data, focusing on letters with dots (such as "i" and "j") to embed 0 bit and specific other letters ("a", "A", "c") to encode a 1 bit. This technique, while simple and straightforward, is limited by the small number of letters it uses, which constrains the amount of data that can be embedded in a given text. Meanwhile, the One-Flow-1-bit technique, on the other hand, classifies letters based on their ability to be written in a single continuous stroke. Letters that cannot be written in one flow (like "A", "B", "D") are used to hide a 0 bit, while letters that can be written in one flow (like "C", "G", "I") hide a 1 bit. This technique is more versatile than CALP but is still limited by its reliance on capital letters, which restricts the diversity of cover texts that can be used.

In the embedding process, the hidden message is first converted into binary form based on technique chosen, each character of the hidden message is represented by its value, which is then converted into an 8-bit binary number. Then, Letters with dots are used to represent a binary 0 bit and some selected letters are used to represent a binary 1 bit. The mapped letters are then inserted into the cover text at appropriate positions. The positioning is done in such a way that the cover text appears natural and unaltered to the reader. In the extracting process, the embedded text is analysed, and the positions of the CALP or One-Flow-1-bit letters are identified. The identified letters are converted back into binary form, based on the mapping established during the embedding process. The identified letters are converted back into binary form, based on the mapping established during the embedding process and the binary data is grouped and converted back into characters to reconstruct the hidden message.

2.2 Table Style and Format

Dual-bit text steganography techniques embed two bits of binary data per character by categorizing letters based on more complex features, such as curvature and the presence of vertical or horizontal lines. These methods significantly increase data capacity while maintaining the natural appearance of the text, making them ideal for scenarios requiring higher data concealment. The two techniques of dual-bit feature-based method are shown in Table 2

Table 2
Dual-bit technique in feature-based method

ID	Bit	One-Flow-2-bit		QUAD	
		Group	Letter used	Group	Letter used
A	00	Letters not writable in one flow and has no vertical or horizontal line	"Q, X"	Curved letter	"C, D, G, O, Q, S, U"
B	01	Letters not writable in one flow and has vertical or horizontal line	"A, B, D, E, F, H, K, T"	Letter with middle horizontal line	"A, B, E, F, H, P, R"
C	10	Letters writable in one flow and has no vertical or horizontal line	"C, G, O, S, V, W"	Letter with one vertical straight line	"I, J, K, L, T, Y"
D	11	Letters writable in one flow and has no vertical or horizontal line	"I, J, L, M, N, P, U, Y, Z"	Letter with diagonal line	"M, N, V, W, X, Z"

Table 2 shows the two techniques of dual-bit of feature-based method which are One-Flow-2-bit and QUAD technique. The hidden message is converted into binary form. For One-Flow-2-bit technique, letters are categorized into four groups based on their flow characteristics and the presence of vertical or horizontal lines. The letters not writable in one flow and without vertical or horizontal lines (e.g., "Q", "X") represent binary 00 and letters not writable in one flow but with vertical or horizontal lines (e.g., "A", "B", "D") represent binary 01. Then, letters writable in one flow and without vertical or horizontal lines (e.g., "C", "G", "O") represent binary 10 and letters writable in one flow and with vertical or horizontal lines (e.g., "I", "J", "L") represent binary 11. For QUAD technique, letters are categorized based on their structural features, Letters without lines or curves (e.g., "I", "T") represent binary 00 and Letters without lines but with curves (e.g., "C", "O") represent binary 01. Then, letters with lines but without curves (e.g., "L", "Z") represent binary 10. Letters with both lines and curves (e.g., "B", "D", "Q") represent binary 11. Moreover, the binary data is embedded by selecting appropriate letters from the cover text based on their categorized characteristics.

In the extracting process, the embedded text is analysed to identify the categorized characteristics of the letters. The identified letters are converted back into binary form based on their categorization One-Flow-2-bit or QUAD technique. The binary data is grouped and converted back into characters to reconstruct the hidden message.

2.3 Table Style and Format

This paper utilizes two main performances which are letter used and embedding capacity size bit. The choice of letters used performance plays crucial roles in text steganography. According to Tyagi et al. [35], the selection of letters for embedding binary data depends on their visual or structural characteristics. For instance, certain letters might be used to represent specific binary values based on features such as the presence of dots or continuous stroke patterns. The dataset's composition, including the range of letters and their frequencies, impacts the overall embedding capacity and the effectiveness of the steganographic method.

The capacity size of a dataset, which includes the total number of characters and their usability for embedding, directly affects the embedding capacity. The ratio is determined by calculating the number of bits of data that can be hidden relative to the size of the cover text. This calculation is typically performed by analyzing the system's output after loading the dataset, which provides an interface showing the total bits in kilobyte available for embedding [7], [35]. This approach assists in assessing how effectively the steganographic method utilizes the cover text and manages the concealed information.

3. Results

The results of this paper provide a detailed comparison of the four text steganography techniques in term of single-bit and dual-bit techniques on their performance in terms of letter used and embedding capacity size bit. By analyzing how each method utilizes the visual and structural characteristics of letters to encode binary data, we can understand the strengths and limitations of each technique. The following subsections present the findings for letter usage performance and embedding capacity, highlighting the effectiveness of single-bit versus dual-bit approaches.

3.1 Performance of Letter Used

The performance of letter used of each technique was analysed in terms of the total number of bits that could be hidden within a given stego text. Figure 1 show how the letter used that utilize on four techniques based on 20 stego text achieved.

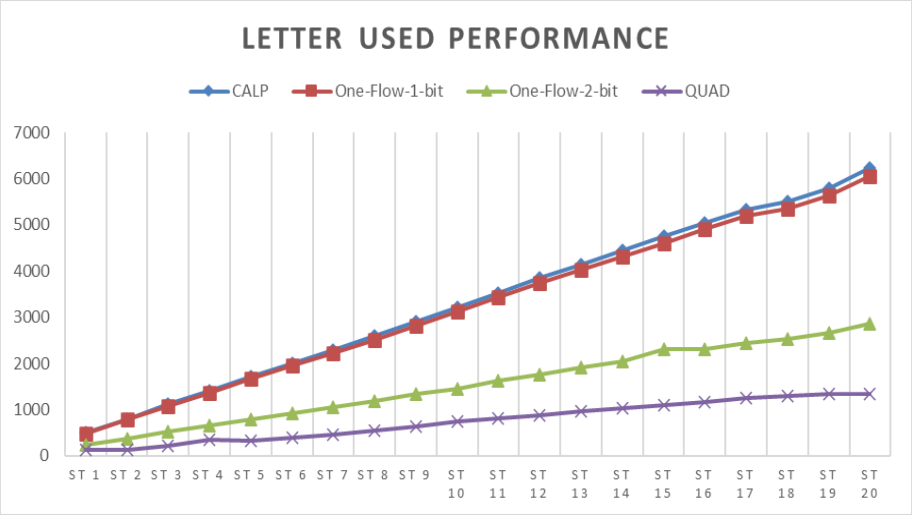

Fig. 1. Comparison of letter used performance of feature-based methods

The performance of letter used of each technique was analysed in terms of the total number of bits that could be hidden within a given stego text. Figure 1 show how the letter used that utilize on four techniques based on 20 stego text achieved.

Figure 1 presents a visual comparison of how each method utilizes letters for hiding data. The methods assessed include CALP, One-Flow (both single-bit and dual-bit versions), and QUAD. The graph highlights that the CALP technique consistently achieves the expected embedding capacity, meaning it can hide the exact number of bits anticipated. On the other hand, other techniques, particularly the dual-bit versions, tend to lose some letters during the embedding process, resulting in a discrepancy between the expected and actual embedding capacity.

Table 3
Result of letter used single-bit and dual-bit techniques

HM	Single-bit techniques		
	Embed Expected	***CALP***	***One-Flow-1-bit***
1	496	496	487
2	800	800	780
3	1112	1112	1077
4	1416	1416	1373
5	1712	1712	1664
6	2008	2008	1952
7	2296	2296	2231
8	2592	2592	2518
9	2896	2896	2818
10	3208	3208	3122
11	3520	3520	3427
12	3840	3840	3737
13	4144	4144	4030
14	4448	4448	4321
15	4744	4744	4606
16	5040	5040	4898
17	5336	5336	5189
18	5496	5496	5341
19	5784	5784	5626
20	6224	6224	6053

HM	Dual-bit techniques		
	Embed Expected	***One-Flow-2-bit***	***QUAD***
1	248	235	132
2	400	373	130
3	556	516	223
4	708	652	342
5	856	787	323
6	1004	925	397
7	1148	1060	469
8	1296	1193	544
9	1448	1335	646
10	1604	1447	753
11	1760	1620	803
12	1920	1767	879
13	2072	1906	960
14	2224	2047	1041
15	2372	2315	1090
16	2520	2315	1168
17	2668	2451	1247
18	2748	2520	1301
19	2892	2652	1333
20	3112	2858	1342

*Note= Total letter embedded in the technique

Table 3 presents the comparison of letter used for single-bit and dual-bit text steganography techniques. The CALP method demonstrates a precise match between the expected and actual number of embedded bits in preserving the integrity of the stego text. In contrast, the One-Flow-1-bit and One-Flow-2-bit techniques exhibit a slight reduction in the number of letter used compared

to the expected values, with the efficiency notably decreasing in the dual-bit approach as more bits are embedded. The QUAD technique shows the lowest efficiency among the techniques, with a significant reduction in the number of usable letters for embedding, particularly in dual-bit scenarios. The difference between single-bit and dual-bit techniques is evident, with single-bit embedding one binary bit per letter and dual-bit embedding two binary bits per letter. CALP stands out as particularly effective due to its precision in achieving the expected embed count, while the other techniques experience some loss of letters during the embedding process.

3.2 Performance of Embedding Capacity Size Bit

The embedding capacity size bit determines based on calculation in the system that displayed the total number of bits in the dataset after the dataset is loaded into the system. Figure 2 compares the performance of the four techniques based on their embedding capacity.

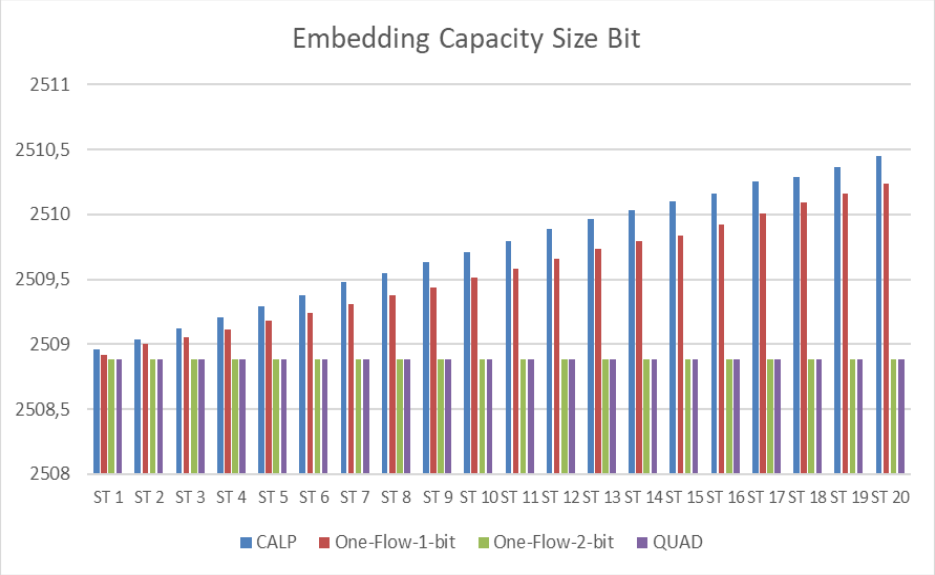

Fig. 2. Comparison of embedding capacity size bit performance of feature-based methods

Figure 2 presents a comparison of the embedding capacity size bit performance across different feature-based methods. The results reveal that the CALP and One-Flow (1-bit) techniques exhibit variations in their embedding capacity sizes across different stego texts. This variability indicates that these single-bit methods sometimes require more space to embed the same amount of information, which can lead to an increase in the overall size of the text and potentially make it more detectable.

In contrast, the One-Flow (2-bit) and QUAD techniques show a stable embedding capacity size bit across different texts, meaning that the amount of embedded data does not cause the overall size of the text to fluctuate. This stability is particularly beneficial because it ensures that the stego text remains consistent in size, making it less likely to attract attention or appear suspicious. These dual-bit techniques are designed to embed two bits of data per letter, which increases the amount of information that can be hidden without expanding the text size. The further elaboration of this embedding capacity size bit performance in kilobyte (Kb) shows in Table 4.

Table 4
Result of letter used single-bit and dual-bit techniques

HM	Single-bit techniques	
	CALP	***One-Flow-1-bit***
1	2508.96	2508.92
2	2509.04	2509.00
3	2509.12	2509.05
4	2509.21	2509.11
5	2509.29	2509.18
6	2509.38	2509.24
7	2509.48	2509.31
8	2509.55	2509.38
9	2509.63	2509.44
10	2509.71	2509.51
11	2509.79	2509.58
12	2509.89	2509.66
13	2509.96	2509.73
14	2510.03	2509.79
15	2510.10	2509.84
16	2510.16	2509.92
17	2510.25	2510.01
18	2510.29	2510.09
19	2510.36	2510.16
20	2510.45	2510.24

HM	Dual-bit techniques	
	One-Flow-2-bit	***QUAD***
1	2508.88	2508.88
2	2508.88	2508.88
3	2508.88	2508.88
4	2508.88	2508.88
5	2508.88	2508.88
6	2508.88	2508.88
7	2508.88	2508.88
8	2508.88	2508.88
9	2508.88	2508.88
10	2508.88	2508.88
11	2508.88	2508.88
12	2508.88	2508.88
13	2508.88	2508.88
14	2508.88	2508.88
15	2508.88	2508.88
16	2508.88	2508.88
17	2508.88	2508.88
18	2508.88	2508.88
19	2508.88	2508.88
20	2508.88	2508.88

*Note= in kilobyte (Kb)

Table 4 provides detailed data that backs up the observations from Figure 2. It shows that single-bit techniques like CALP and One-Flow (1-bit) tend to slightly increase the capacity size as they embed more bits, which suggests they aren't using the available space as efficiently. On the other hand, the One-Flow (2-bit) and QUAD methods keep the size bit consistent, meaning they can embed more

data without increasing the overall size of the text. This makes them more efficient in terms of space usage while keeping the text size stable. The main strength of the One-Flow (2-bit) and QUAD techniques is their efficiency and consistency based on embedding size bit performance. These techniques embed two bits per letter while keeping the text size unchanged, which helps them avoid the issues that single-bit techniques might face, such as increasing the size of the stego text and making it more noticeable. Because of this, the One-Flow (2-bit) and QUAD techniques are especially useful in situations where it's important to preserve the original text size and reduce the chances of detection.

4. Conclusions

In this paper, it compares four text steganography techniques that consist of single-bit techniques which are CALP and One-Flow-1-bit, then dual-bit technique One-Flow-2-bit, and QUAD through a comparative analysis focused on letter usage and embedding capacity. The single-bit techniques, CALP and One-Flow-1-bit, demonstrated efficiency in scenarios with moderate data concealment needs, particularly when working with shorter texts. These methods, while simple and straightforward, are limited by the small number of letters they use, which constrains their embedding capacity. On the other hand, dual-bit techniques like One-Flow-2-bit and QUAD significantly enhance the embedding capacity by utilizing more complex categorization of letters based on structural features. These techniques are better suited for applications requiring the concealment of larger data volumes, offering higher data capacity while maintaining the natural appearance of the cover text.

The comparative analysis revealed that while single-bit techniques are useful for low to medium-security scenarios, dual-bit techniques provide a more robust solution for higher security and data capacity needs. Specifically, the QUAD technique stood out for its ability to achieve the highest data capacity among the methods analysed, making it ideal for applications requiring extensive data concealment. However, this increased capacity comes with added complexity in the encoding process, which may require more computational resources. Thus, the paper highlights the importance of selecting the appropriate technique based on the specific requirements of the steganographic task, such as the size of the cover text, the desired level of data concealment, and the operational context. For future work, It expected to optimize these techniques for different types of text or develop hybrid methods that combine the strengths of both single-bit and dual-bit approaches.

Acknowledgement

This research was not funded by any grant. Then, we would like to thank for members of School of Computing, Universiti Utara Malaysia and for their moral support for the realization of this work.

References

[1] Prasetyo, K. A. D. I. “Text Steganography Based on Unicode Characters as Marker in Indonesian Excel File.” *Journal of Theoretical and Applied Information Technology (JATIT), Vol. 102, no. 12* (2024): 4972–4988.

[2] Utama, S., and R. Din. “Performance Review of Feature-Based Implementation Text Steganography Approach Method.” *Journal of Advanced Research in Applied Sciences and Engineering Technology, Vol. 2, no. 2* (2022): 325–333.

[3] Nechta, I. V. “New Steganalysis Method for Text Data Produced by Synonym Run-Length Encoding.” In *Proceedings of the 14th International Scientific Conference on Actual Problems of Electronic Instrument Engineering (APEIE 2018)*, 188–190. 2018. https://doi.org/10.1109/APEIE.2018.8545230.

[4] Din, R., R. Bakar, S. Utama, J. Jasmis, and S. J. Elias. "The Evaluation Performance of Letter-Based Technique on Text Steganography System." *Bulletin of Electrical Engineering and Informatics* 8, no. 1 (2019): 291–297. https://doi.org/10.11591/eei.v8i1.1440.

[5] Saad, R. "The Impact of International Professional Accreditation to Enhance Quality at Higher Education." *Academy of Accounting and Financial Studies Journal* 26, no. 2 (2022): 1–20.

[6] Mansor, F. Z., A. Mustapha, R. Din, A. Abas, and S. Utama. "An Antonym Substitution-Based Model on Linguistic Steganography Method." *Indonesian Journal of Electrical Engineering and Computer Science* 12, no. 1 (2018): 225–232. https://doi.org/10.11591/ijeecs.v12.i1.pp225-232.

[7] Alghamdi, N., and L. Berriche. "Capacity Investigation of Markov Chain-Based Statistical Text Steganography: Arabic Language Case." *ACM International Conference Proceeding Series* (2019): 37–43. https://doi.org/10.1145/3314527.3314532.

[8] Chaudhary, S., M. Dave, and A. Sanghi. "Text Steganography Based on Feature Coding Method." *ACM International Conference Proceeding Series* (2016): 5–8. https://doi.org/10.1145/2979779.2979786.

[9] Muhammad, M. H., H. S. Hussain, R. Din, H. Samad, and S. Utama. "Review on Feature-Based Method Performance in Text Steganography." *Bulletin of Electrical Engineering and Informatics* 10, no. 1 (2021): 427–433. https://doi.org/10.11591/eei.v10i1.2508.

[10] Gutub, A., and K. Alaseri. "Hiding Shares of Counting-Based Secret Sharing via Arabic Text Steganography for Personal Usage." *Arabian Journal for Science and Engineering* (2019): 1–26. https://doi.org/10.1007/s13369-019-04010-6.

[11] Shafi, I., et al. "An Adaptive Hybrid Fuzzy-Wavelet Approach for Image Steganography Using Bit Reduction and Pixel Adjustment." *Soft Computing* (2017). https://doi.org/10.1007/s00500-017-2944-5.

[12] Din, R., R. Bakar, A. Ismail, A. Mustapha, and S. Utama. "Evaluation Review of Effectiveness and Security Metrics Performance on Information Technology Domain." *Indonesian Journal of Electrical Engineering and Computer Science* 16, no. 2 (2019): 1059–1064. https://doi.org/10.11591/ijeecs.v16.i2.pp1059-1064.

[13] Jabbar, A., Q. Almaliki, S. M. Abd, I. A. Lafta, and R. Din. "Application of the Canny Filter in Digital Steganography." *Vol. 1, no. 1* (2024): 21–30.

[14] Tian, H., Y. Wu, C. C. H. Chang, C. Yongfeng, W. Yonghong, T. Cai, L. Yiqiao, and J. Liu. "Steganalysis of Adaptive Multi-Rate Speech Using Statistical Characteristics of Pulse Pairs." *Signal Processing* 134 (2017): 9–22. https://doi.org/10.1016/j.sigpro.2016.11.013.

[15] Din, R., R. A. Thabit, N. I. Udzir, and S. Utama. "Triad-Bit Embedding Process on Arabic Text Steganography Method." *Bulletin of Electrical Engineering and Informatics* 10, no. 1 (2021): 493–500. https://doi.org/10.11591/eei.v10i1.2518.

[16] Ditta, A., M. Azeem, S. Naseem, K. G. Rana, M. A. Khan, and Z. Iqbal. "A Secure and Size Efficient Algorithm to Enhance Data Hiding Capacity and Security of Cover Text by Using Unicode." *Journal of King Saud University – Computer and Information Sciences* (2020). https://doi.org/10.1016/j.jksuci.2020.07.010.

[17] Bhattacharyya, S., P. Indu, S. Dutta, A. Biswas, and G. Sanyal. "Hiding Data in Text Through Changing in Alphabet Letter Patterns (CALP)." *Journal of Global Research in Computer Science* 2, no. 3 (2011).

[18] Kouser, S., A. Khan, and E. Qamar. "A Novel Content-Based Feature Extraction Approach: Text Steganography." *International Journal of Computer Science and Information Security* 14, no. 12 (2016): 916–923.

[19] Dulera, S., D. Jinwala, and A. Dasgupta. "Experimenting with the Novel Approaches in Text Steganography." *International Journal of Network Security & Its Applications* 3, no. 6 (2011): 213–225.

[20] Wu, N., P. Shang, J. Fan, Z. Yang, W. Ma, and Z. Liu. "Coverless Text Steganography Based on Maximum Variable Bit Embedding Rules." *Journal of Physics: Conference Series* (2019): 022078. https://doi.org/10.1088/1742-6596/1237/2/022078.

[21] Din, R., and S. Utama. "The Design Review of Feature-Based Method in Embedding the Hidden Message in Text as the Implementation of Steganography." *Borneo International Journal* 6, no. 3 (2023): 88–95. Available at www.majmuah.com.

[22] Wu, Y., Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, and Y. Fu. "Large Scale Incremental Learning." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 374–382. 2019. https://doi.org/10.1109/CVPR.2019.00046.

[23] Torvi, S. D., K. B. S. Kumar, and R. Das. "An Unique Data Security Using Text Steganography." In *IEEE 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, 3834–3838. 2016.

[24] Zhao, H., et al. "Compressive Sensing-Based Secret Signals Recovery for Effective Image Steganalysis in Secure Communications." *Multimedia Tools and Applications* 78, no. 20 (2019): 29381–29394. https://doi.org/10.1007/s11042-018-6065-7.

[25] Akotoye, F. X. K., Y. E. Yakavor, J. Kwofie, and F. La Tirogo. "Character Pair Text Steganography Based on the Enhanced Paragraph Approach." In *IEEE 7th International Conference on Adaptive Science and Technology (ICAST)*, 1–5. 2018.

[26] Utama, S., R. Din, and M. Mahmuddin. "The Performance Evaluation of Feature-Based Technique in Text Steganography." *Journal of Engineering Science and Technology* 12 (2017): 169–180.

[27] Iqbal, M. M., U. Khadam, K. J. Han, J. Han, and S. Jabbar. "A Robust Digital Watermarking Algorithm for Text Document Copyright Protection Based on Feature Coding." In *15th International Wireless Communications and Mobile Computing Conference (IWCMC)*, 1940–1945. 2019.

[28] Naharuddin, A., A. D. Wibawa, and S. Sumpeno. "A High Capacity and Imperceptible Text Steganography Using Binary Digit Mapping on ASCII Characters." In *2018 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, 287–292. 2019. https://doi.org/10.1109/ISITIA.2018.8711087.

[29] Kamath, K. M., and R. S. Kunte. "Framework for Reversible Data Hiding Using Cost-Effective Encoding System for Video Steganography." *International Journal of Electrical and Computer Engineering* 10, no. 5 (2020): 5487–5496. https://doi.org/10.11591/IJECE.V10I5.PP5487-5496.

[30] Bajaj, I., and R. K. Aggarwal. "Steganography Using HTML Web Pages as a Carrier: A Survey." *SSRN Electronic Journal* (2019). https://doi.org/10.2139/ssrn.3351033.

[31] Alsaadi, H. I., M. K. Al-Anni, R. M. Almuttairi, O. Bayat, and O. N. Ucan. "Text Steganography in Font Color of MS Excel Sheet." *ACM International Conference Proceeding Series* (2018). https://doi.org/10.1145/3279996.3280006.

[32] Kouser, S., and A. Khan. "A Novel Feature Extraction Approach: Capacity Based Zero-Text Steganography." (2017): 85–98.

[33] Reddy, R. P. K., C. Nagaraju, and N. Subramanyam. "Text Encryption Through Level-Based Privacy Using DNA Steganography." *Vol. 3, no. 3* (2014).

[34] Saad, E., N. Al, and A. Algamdi. "Survey of Steganography Applications." *Al-Salam Journal for Engineering and Technology*, (2023): 69–75. https://doi.org/10.55145/ajest.2023.01.01.008

[35] Tyagi, S., R. K. Dwivedi, and A. K. Saxena. "A High Capacity PDF Text Steganography Technique Based on Hashing Using Quadratic Probing." *International Journal of Intelligent Engineering and Systems* 12, no. 3 (2019): 192–202. https://doi.org/10.22266/IJIES2019.0630.20