applied sciences

MDPI

Review

A Comparative Analysis of Arabic Text Steganography

Reema Thabit [1,*], Nur Izura Udzir [1,*], Sharifah Md Yasin [1], Aziah Asmawi [1], Nuur Alifah Roslan [1] and Roshidi Din [2]

[1] Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, UPM, Serdang 43400, Selangor Darul Ehsan, Malaysia; ifah@upm.edu.my (S.M.Y.); a_aziah@upm.edu.my (A.A.); nuuralifahroslan@gmail.com (N.A.R.)

[2] School of Computing, College of Arts and Sciences, Universiti Utara Malaysia, Sintok 06010, Kedah, Malaysia; roshidi@uum.edu.my

* Correspondence: rmbinthabit@gmail.com (R.T.); izura@upm.edu.my (N.I.U.)

Abstract: Protecting sensitive information transmitted via public channels is a significant issue faced by governments, militaries, organizations, and individuals. Steganography protects the secret information by concealing it in a transferred object such as video, audio, image, text, network, or DNA. As text uses low bandwidth, it is commonly used by Internet users in their daily activities, resulting a vast amount of text messages sent daily as social media posts and documents. Accordingly, text is the ideal object to be used in steganography, since hiding a secret message in a text makes it difficult for the attacker to detect the hidden message among the massive text content on the Internet. Language's characteristics are utilized in text steganography. Despite the richness of the Arabic language in linguistic characteristics, only a few studies have been conducted in Arabic text steganography. To draw further attention to Arabic text steganography prospects, this paper reviews the classifications of these methods from its inception. For analysis, this paper presents a comprehensive study based on the key evaluation criteria (i.e., capacity, invisibility, robustness, and security). It opens new areas for further research based on the trends in this field.

Keywords: information hiding; covert communication; text hiding; Arabic script; Arabic characteristics

Citation: Thabit, R.; Udzir, N.I.; Yasin, S.M.; Asmawi, A.; Roslan, N.A.; Din, R. A Comparative Analysis of Arabic Text Steganography. *Appl. Sci.* **2021**, *11*, 6851. https://doi.org/10.3390/app11156851

Academic Editor: David Megías

Received: 28 March 2021
Accepted: 28 May 2021
Published: 26 July 2021

1. Introduction

The rapid expansion of Internet technologies enables flow of vast amounts of information across the public channel with risks of attacks. Under those circumstances, securing sensitive information has become a serious issue for by governments, organizations, and individuals due to the risk of attack (Different techniques for hiding the text information using text steganography [1,2]). To address this challenge, researchers have proposed various methods to protect secure messages transmitted via public and private communication channels.

The two essential methods that play significant roles in information security are data encryption and data hiding. Data encryption is an aspect of cryptography applied to protect the confidential message being transmitted across private and public channels by converting it to a scribbled enciphered form. Thus, the carrier object after encryption is meaningless. Meanwhile, information hiding conceals the secret message to make it unnoticed/invisible in the course of its transmission via the public (untrusted) communication channel [3]. Invisibility is the fundamental difference between cryptography and information hiding [4].

Information hiding can take one of two forms: Watermarking or steganography. Employing a watermarking to embed the secret information provides proof of ownership of the carrier object, so it is suitable for copyright protection [5]. Steganography conceals the existence of secret information in the cover carrier [6]. Steganography uses several classes of cover media (i.e., audio, video, image, text, network, and DNA).

A text is an "object" used by users of the public channel in their daily activities. It is an ideal cover item for the data travelling between a sender and a receiver because of its small size compared to other objects [7,8]. Moreover, text steganography improves the hidden capacity by exploiting language characteristics, grammatical or orthographic, which differs from one language to another [9–11]. Nevertheless, text stenography is one of the most challenged classes of stenography because of the lack of redundant data in text files [12]. In addition, text documents have an almost identical structure, which makes changes easily visible.

Social engagements are the most frequent activities of public channel users [13], with 93% of users visiting social networking platforms and 98.1% of users communicating by text. These online activities, which also involve confidential information, present the need for information hiding such as text steganography. At the same time, these activities offer convenient opportunities and advantages to hide information among the huge availability of online text. For example, social media posts, mail messages, and books in large libraries pose obstacles to eavesdroppers. This is attributed to the difficulty associated with examining, analyzing, and filtering the vast amount of text to determine which text may contain hidden information. In text steganography, the structure and language characteristics of texts are used to hide secret information. English script is mostly utilized compared to Arabic script in text hiding [14].

Although several surveys of information security for digital text have been published in recent years (see Table 1), these studies have not focused on utilizing the attributes of the Arabic language to hide secret data. Arabic script is one type of media in text steganography classes, and to the best of our knowledge, there is very limited, comprehensive information on Arabic text steganography methods. Thus, this review fills this gap by providing a comparative analysis of the existing Arabic text steganography methods.

Table 1. Prior surveys on text steganography.

Year	Reference	Highlights	Scope
2011	[15]	Exhibits the performance analysis of the text steganography classes by analyzing the strengths and weaknesses.	Text steganography
2016	[16]	Classifies text steganography methods into 2 groups based on changes in format and meaning. However, it summarizes the proposed methods without providing comprehensive analysis.	Text steganography
2016	[17]	Discusses the use of Genetic Algorithm (GA) in text steganography for avoiding suspicion. GA is widely used in image and video steganography compared to text steganography.	GA text steganography
2017	[18]	Presents a taxonomy of the protection and verifying methods (watermarking, steganography, and cryptography) for integrating the Arabic text using the online Qur'anic content as a case study.	Text preserving and verifying
2017	[19]	Classifies text steganography methods based on the embedding level into 3 levels: Bit-level, character-level, and mixed-level.	Text steganography
2018	[20]	This is a comparative study of structural methods in steganography and watermarking that are applied to copyright protection.	Text copyright protection
2018	[21]	Discusses, in general, the 3 categories in text steganography: Format-based methods, random and statistical generation, and linguistics.	Text steganography
2018	[22]	Provides the assessment of text steganography methods and discuss the current challenges.	Text steganography
2019	[4]	Presents an analysis of the security challenges and the pros and cons of structural text hiding methods.	Structural text hiding
2020	[23]	Addresses steganography methods' limitations and analyses their performance in each class, such as image, audio, and text video.	Steganography (image, audio, video, and text)
2021	[24]	Focuses on the comparative analysis of text steganography methods in feature-based category.	Feature-based text steganography

In this regard, this paper reviews text steganography, considering that it is a widely used steganography type. It focuses on analyzing Arabic text steganography methods in view of its propensity for information hiding, thus benefiting those who use Arabic text to embed confidential information on the public channel.

This effort is tailored at opening novel approaches useful in the exploitation of Arabic text as a cover for protecting sensitive information.

The contributions of this survey paper are summarized as follows:

- It presents a brief review of existing linguistic text steganography methods.
- It summarizes Arabic text steganography methods from their initiation while identifying their methodologies and analyzing their strengths and weaknesses.
- It provides a comparative analysis of Arabic text steganography based on the key evaluation criteria (i.e., capacity, invisibility, robustness, and security).
- It recommends future path work in Arabic text steganography.

The rest of this paper is organized as follows. Section 2 discusses steganography scenarios and types. Section 3 presents the background on text steganography, focusing on the language-based methods. The strengths and limitations of Arabic text steganography methodologies are discussed in Section 4, and Section 5 briefs the evaluation criteria for Arabic text steganography methods. Section 6 provides recommendations for future work. Finally, Section 7 concludes this paper.

2. Steganography

Steganography is the art of secret communication between confidential parties. It is the science in which the confidential message is embedded undetectably around the signal of the carrier so that no one except the sender and the intended recipient will be aware of the existence of the hidden data. The technical term steganography, derived from the Greek words *steganos* and *graphein*, means protected writing [25]. Therefore, a stenographic system facilitates data embedding in a discrete manner for easy access and data extraction, promotes a high capacity of embedding, and preferably includes some amount of resistance to removal [26]. Steganography allows the secret message to be exchanged without the knowledge or suspicions of the other parties. A successful attack on a stego object is the detection of the secret communication.

2.1. Steganography Scenario

We illustrate a typical steganography scenario in Figure 1. The entity responsible for sending the secret information (called the sender) applies a hiding technique to protect the secret message travelling through a public channel such as the Internet. The item type that contains the secret message could be as text, image, video, or audio. Similar to the secret message object, the cover could be an image, video, etc.

The embedding process needs a secret key (stego key) that protects the concealed message from being extracted by an attacker. After the embedding process, the stego object which represents the hidden message within the cover object is generated. Thereafter, the stego object is sent through the public channel to the receiver. If the stego key is designed as a private key, it will be sent to the receiver as a hidden key within the stego file.

Otherwise, a stego key will be produced as a public key, encrypted, and sent separately to the receiver via the public channel. The receiver can extract the secret message by exploiting the stego key and applying the extraction algorithm that corresponds to the embedding algorithm.

The risk appears when the hidden message is transmitted through the public channel where many attackers/eavesdroppers are ready to attack the stego object. The attacker extracts the hidden message by tracing the embedding algorithm and breaking the stego key. If an attacker is incapable of extracting the hidden message, he could tamper with the stego object to produce a tampered object by destroying the hidden message.

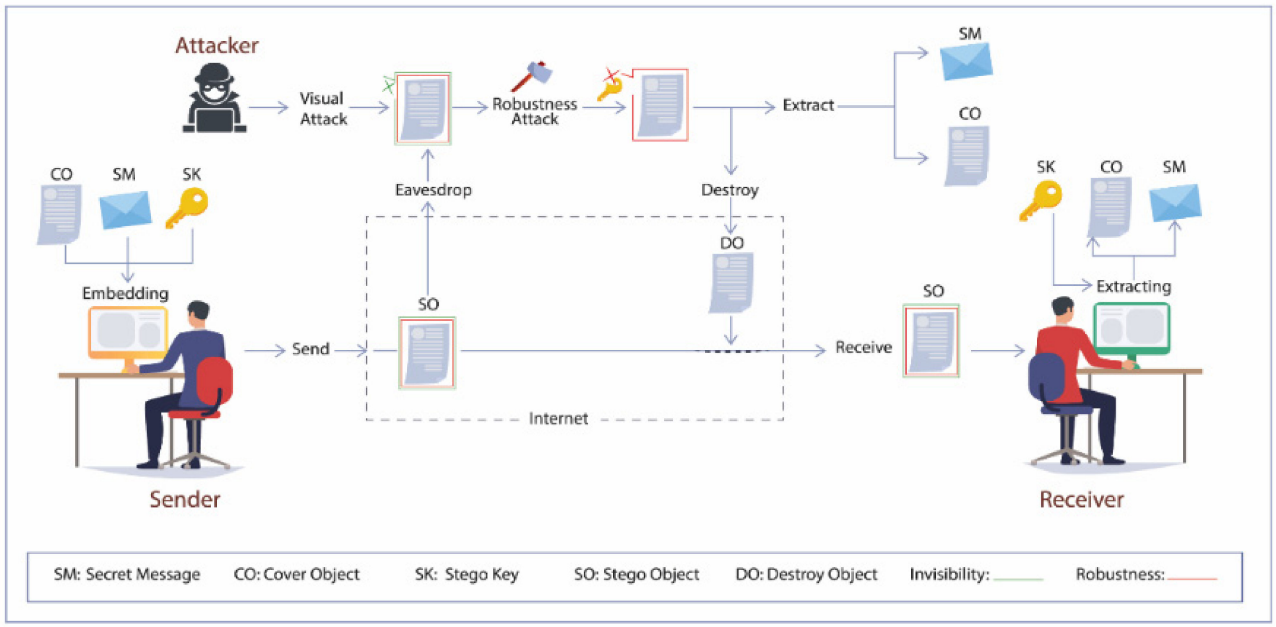

Figure 1. Scenario of steganography.

Therefore, it is imperative to build a steganography method that achieves a tradeoff between four evaluation criteria: Invisibility, robustness, capacity, and security. The proposed steganography techniques seek to hide as much information into a cover object as possible without affecting its invisibility. The essence of invisibility is to prevent distortion of the cover object's appearance to ensure it remains unnoticed by the eavesdropper. Robustness prevents the attacker from either extracting or destroying the hidden message. If an attacker notices altered cover object, then he could break into the first shield of the defense. Then, he could break the robustness, which is considered the second shield of the defense. A steganography technique with a high level of perceptual and robustness will achieve a high stage of protection [4,27]. Hence, it is important to study and analyze steganography techniques and assess their performance using the four evaluation criteria mentioned earlier.

2.2. Steganography Types

The strength of steganography security is connected to the inability of observers to distinguish the cover object from the stego object. Cover objects can be customized with varied media types such as image, video, audio, text, network, and DNA.

2.2.1. Image Steganography

When the carrier file is an image, the stenography type is referred to as image steganography. Here, the image files (e.g., JPEG, GIF, BMP, and PNG) are utilized to cover the sensitive message. Image steganography is achieved by employing the image format, spatial domain steganography and adaptive steganography [28].

Recently, an image steganography approach used a data mapping technique to minimize the number of bits changed per pixel. Four hidden data bits were mapped to the four most important bits of a cover pixel [29].

2.2.2. Video Steganography

Combining image and sound, a video file such as MPEG, AVI, or MP4 carries the capability of hiding a massive amount of sensitive information. A computed tomography (CT) scan, which is applied for image steganography, can likewise be implemented for video steganography by embedding sensitive information in each image of the video [11,30,31]. Other commonly applied techniques for video steganography include the Least Significant

Bit (LSB); Tri-way Pixel-Value Differencing (TPVD), which embeds the secret bits in the Inline frame (Iframe); and Bit Plane Complexity Segmentation (BPCS), which is also utilized for embedding secret bits within the MPEG video.

2.2.3. Audio Steganography

The file that saves digital sound (for example, MP3 or WAV) can be utilized to protect secret messages by shifting the binary sequence of that file. In various modern steganography methods, LSBs are changed with error diffusion. It is additionally conceivable to conceal secret messages using inaudible frequencies [30].

2.2.4. Text Steganography

Hiding sensitive information in a text file was the earliest means of transferring confidential messages. The intended receiver can only retrieve concealed data. Particularly, the text is ideal because it is a common object that is widely used in daily activities. This makes it difficult for the attacker to distinguish the hidden message [31]. Various methods have been introduced in this field. Section 3 provides further details on text steganography.

2.2.5. Network Steganography

In this type of steganography, a single network protocol is adjusted to embed the secret bits. Network protocols such as Transmission Control Protocol (TCP), Protocol Data Unit (PDU), User Datagram Protocol (UDP), Internet Control Message Protocol (ICMP), and Internet Protocol (IP) are used as cover objects. Network steganography is profoundly secure and robust [32].

2.2.6. DNA Steganography

DNA steganography is characterized by the shortest computation time because it has less storage and power requirements. Conventional storage media require 1012 cubic nanometers to store 1 bit of data, while DNA memory stores data at a density of about 1 bit per cubic nanometer. No power is needed during the DNA computation [33,34].

3. Text Steganography

Centered on the embedding method used to conceal the sensitive information in the cover text, text steganography can be divided into three categories [35]: Random and Statistical Generation, Linguistic, and Format-based.

3.1. Random and Statistical Generation

This class generates a cover item based on statistical properties by considering word and character sequences. Sometimes, the created stego text attracts a person who intercepts the message by appearing as a random sequence of words/characters.

As an example in this category, the structure of the Omega network integrated with part-of-speech (POS) in [36] by substituting "verb from cover" with "verb from secret" and "noun of cover" by "noun of secret." Besides, letter frequency and word length were two of the statistical properties used by the authors of [37] to create a stego word using the actual dictionary items and a codebook of mappings between bit sequences and lexical items. Table 2 shows an example of Random and Statistical Generation. The stego words consist of the repetitions of the three letters 'a,' 'r,' and 'd' in an indecipherable way. As a result, the generated file in this approach is incomprehensible and raise suspicions.

Table 2. Example of Random and Statical Generation (Data from [36]).

Secret Message	Cover Text	Stego Text
A	"abaca"	"aard aard aard aard aard aard aard aard aard aard aard aard"

3.2. Linguistic Steganography

Linguistic steganography entails concealing confidential information by utilizing the language of words or other linguistic features. Linguistic methods comprise two groups: Syntactic and synonym. The syntactic method depends on the use of punctuation [38–40]. The synonym method has been used in the dictionary in place of the interactive word (by some carrier file words) to pass the hidden bits [31,41]. Table 3 displays an example of linguistic steganography. In this example, the secret bits are hidden by substituting the words using a dictionary. For instance, the word "Trap" has replaced by the word "Gun" to hide one secret bit. (We show the replaced words as underlined.)

Table 3. Example of Linguistic Steganography (Data from [31]).

Secret Message	Cover Text	Stego Text
Keep the gun under the shed	"Today is the first day of summer which starts with light and cozy sunshine. But eventually the sun becomes scorching and heat goes up. All the rivers and ponds become dried up. People used to wear light clothes and eat less spicy foods. Several summer camps are organized for kids in hilly areas. Trap shooting, swimming, trekking, rock climbing, biking also included as sports. One such popular summer camp is in Shimla. Kids used to leave their belongings and bed beneath the tent. "	"Today is the first day of summer which starts with light and cozy sunshine. But eventually the sun becomes scorching and heat goes up. All the rivers and ponds become dried up. People used to wear light clothes and eat less spicy foods. Several summer camps are organized for kids in hilly areas. Gun shooting, swimming, trekking, rock climbing, biking also included as sports. One such popular summer camp is in Shimla. Kids used to keep their belongings and bed under the shed."

3.3. Format-Based Steganography

This group changes physical document formatting to cover secret information. Deliberate misspellings, font resizing, and space injection, among others, are examples of format-based methods used in text steganography. Although these format-based methods might trick the human eye, they cannot trick computer systems or extend the length of the stego text [42]. Furthermore, these methods are less robust against text retyping attacks [43]. The format-based category is divided into word-rule and feature-based methods as demonstrated in Figure 2.

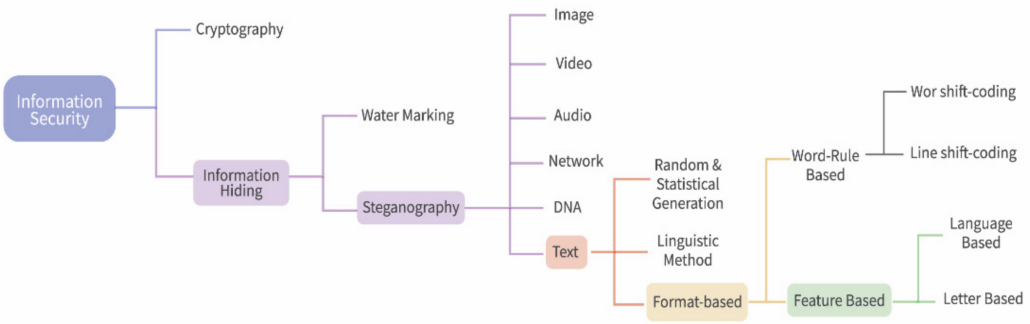

Figure 2. Classification of steganography.

The word-rule involves two branches: Word shift-coding and line-shift coding [44–46]. The feature-based method is divided into language-based and letter-based methods. Table 4 exhibits an example of format-based steganography. In this case, the produced stego text is identical in appearance to the cover text. The alphabets are grouped into two categories: Round shape and curve shape. In each class, the letters are divided into two groups. A letter can represent two secret bits based on its group.

Table 4. Example of format-based text steganography (Data from [37]).

Secret Message	Cover Text	Stego Text
110	"All birds can fly. This is a bird. Ostrich can also fly."	"All birds can fly. This is a bird. Ostrich can also fly."

A feature-based method manipulates the shape, size, and position that relates to the features and structures of the text font. This method prevents the reader from recognizing the secret message or information in the text [47]. Table 5 summarizes the differences between the three categories using examples described by the authors of [33,38,39].

Table 5. The main differences between the 3 categories.

Category	Description
Random and Statistical Generation	This method is not based on a specific text. However, the generated text is meaningless and raise suspicions. The computational time is also increased.
Linguistic steganography	The invisibility is improved, but the method still suffers from low capacity. Also, searching through a dictionary for a suitable word/letter to match the secret word/letter increases the computational time.
Format-based	This method improves invisibility and computational time. Nevertheless, it suffers from low embedding capacity.

The characteristics of the feature-based method have encouraged its use by researchers studying languages all over the world. For example, the letter-based method uses the alphabets A to Z, which can be adopted in many languages. The authors of [37,48–50] have studied the feature-based method, which can be operated in any language, either with figures or alphabets. Examples of feature-based embedded methods applied in several languages are reviewed below.

3.3.1. English-Based

The modification of a written status of the mark-up letter to hide the secret message was introduced by the authors of [51]. This was exploited to analyze the concealed secret information in hypertext. Mark-up letters determine the secret bits used to reveal the length of hidden messages. Machine translation was employed by the authors of [52] to hide a secret data. This embedded method translates the transmitted text and allows the source to be kept in its original form. A code representation method known as secret steganography code was proposed [53], which employs the positions of vowels and consonants according to the grammatical sequence.

The authors of [54] used right-to-left and left-to-right remark to conceal information. This embedded method hides the secret data/message without changing the file's information. It also avoids the retyping problem by converting the file into PDF format. Encryption with Cover text and Reordering (ECR) was proposed [55], which uses XOR operation. It merges two characters when enciphered in the original message. Because the suggested mechanism considers encryption and reordering processes, it is convenient to implement cloud computing.

The algorithm described by the authors of [40] utilizes several invisible character symbols for covering 4 bits between alphabets in word symbols such as left remark, right remark, and zero-width joiner. The algorithm can only be applied to specific languages, hence there is a need to extend its embedded method to be applied in any language. The concept of utilizing the font attributes and character frequency to embed the secret characters was presented by the authors of [56]. To accomplish the uniform appropriation in stego characters with the uniform hiding likelihood, this method integrates four models:

Frequency Normalization Set (FNS), Character String Mapping (CSM), embedding, and extracting.

Text justification was considered by the authors of [7] by justifying the cover text's host line based on the character's frequency in the confidential message. This method was deployed in both electronic and printed details. In the same line, the concealed method introduced by the authors of [57] changes the length per line in the text document to embed the secret bits, which are covered using white space between words and an extended line in the cover text.

3.3.2. Chinese-Based

The method suggested by the authors of [58] hides secret bits into characters by rearranging the sizes of the rectangular regions' components in the Chinese alphabets. In the same field, two embedding methods have been presented: The high efficient substitution embedded method (HESM) and the simple substitution embedded method (SSM). To hide a secret bit, SSM changes the traditional form of Chines characters, while HESM uses a substitution dictionary.

3.3.3. Indian-Based

In Hindi script, a specific matra is media vowel representation. This method was used by the authors of [59] to cover a hidden bit by shifting it to left or right. The authors of [60] integrated two hiding algorithms for the Hindi language. The first algorithm involved the existence of letters and their diacritics and compounds. The second proposes a numerical code for Hindi letters, which is based on a 4-bit binary.

Later, the vowels and consonants of the Hindi alphabets are encoded to a specific numerical code based on four binary bits representation. For the Indian language, the substitution hiding method introduced by the authors of [61] uses the longest common subsequence with minimal alteration of the alphabet features. The numerical code text steganography in Hindi character or other similar Indian languages was developed by the authors of [62].

In addition to numerical, a feature-based embedded method in Hindi text that uses grammar was developed by the authors of [63]. This method encodes a bit stream with the Finite State Machine (FSM) to define transition functions and transformable symbols in each category. Like the Hindi language, the use of feature scripts for Bangla text using chain code has been proposed [64]. The chain code is used to translate codes into several signified contour border pixel directions. This approach presents the use of 50 feature vectors of Bangla alphabets or characters.

3.3.4. Polish-Based

Utilizing Polish text to cover the hidden bits has been suggested [65]. This approach assigns points that are greater than the text alphabets' partial sizes. Polish extension characters are employed alongside the alphabets to join certain alphabets clutching the cloistered secret bits.

3.3.5. Thai-Based

The blind steganography strategy was proposed by the authors of [66] for the Thai text exploits redundancies in the way TIS-620 signifies compound alphabets, merging vowels and diacritical symbols.

3.3.6. Czech-Based

The authors of [67] utilized the dot (point) in the Czech language. Additionally, Czech extension characters were employed in the cover letter to indicate the positions of hidden bits.

Table 6 shows the main characteristics of format-based methods in each language. In the next section, Arabic text steganography is extensively reviewed.

Table 6. Fundamental characteristics of format-based text steganography.

Language-Based	Characteristics
English	Text justification, mark-up language in hypertext, font attributes, substitution, and invisible character are the main characteristics used to hide a secret message in English scripts. Most of these methods can be applied to other languages.
Chinese	The main characteristics employed to protect a secret information in Chinese scripts are rearranging the sizes of the rectangular regions' components in the Chinese alphabets and substitution. This method is language-specific, i.e., it is not applicable to other languages.
Indian	Matra, vowels, and substituting are characteristics that have been used to hide the secret message in Indian/Hindi/Bangla scripts.
Polish Czech	The dot and extension in the script have been exploited to hide the confidential message. It can be applied to the Polish, Czech, Arabic, Urdu, Jawi, and Persian language.
Thai	The redundancies of alphabet merging with vowel letter and diacritics in Thai scripts used to hide the secret message. This method is not applicable to other languages.

4. Arabic Text Steganography

Arabic, spoken by approximately 380 million people [68], is the fifth most spoken language in the world [69,70] and the sixth official language of the United Nations [71]. Arabic online content expands during daily activities on the Internet [72]. Arabic is composed of 28 characters that are written in a cursive style similar to Urdu and Farsi. Depending on its place in a word, an Arabic letter changes shape. It may come in the first, middle, or last position or may even be isolated. Each word usually comprises over two letters joined together. Some Arabic letters have one, two, or three dots placed either above or below the letter. In contrast to English, which has no multipoint letters, Arabic has 15 pointed letters, 5 of which are multipoint. The translation of Arabic letters is shown in Table 7.

Arabic words have diacritics called "Harakat" that are added to frame the vowel sounds. The eight Arabic content diacritics are Fathah (اَن), Kasrah (اِن), Damah (اُن), Sukun (اْن), Tanwin Fathah (اًا), Tanwin Kasrah (ن), Tanwin Damah (اٌ), and Shaddah (اّن). These diacritics are essential for understanding the Holy Quran, religious scripts, historical texts, and Arabic learning books. However, most other Arabic text does not contain diacritics. The Arabic text also contains an extension character called kashida, which is used to justify the words, as well as white spaces, which justify the texts. Kashida is inserted after a letter based on its location in a word [73]. Arabic letters can also be divided into 2 groups, i.e., the sun and moon letters, where each group contains 14 letters, as shown in Table 8. This grouping is based on how these letters affect the pronunciation of the definite article (ال) at the beginning of words. The sound of (ل) in the definite article appears in the moon letters and does not appear in the sun letters.

Table 7. Translation of Arabic letters.

Letter	Name of the Letter	Transliteration	Last	Middle	First
ا	alif	a, u, i,:	ـا	ـا	اـ
ب	ba	b	ـب	ـبـ	بـ
ت	ta	t	ـت	ـتـ	تـ
ث	tha	th	ـث	ـثـ	ثـ
ج	jim	j, g	ـج	ـجـ	جـ
ح	ha	h	ـح	ـحـ	حـ
خ	kha	kh	ـخ	ـخـ	خـ
د	dal	d	ـد	ـدـ	دـ
ذ	dhal	dh	ـذ	ـذـ	ذـ
ر	ra	r	ـر	ـرـ	رـ
ز	za	z	ـز	ـزـ	زـ
س	sin	s	ـس	ـسـ	سـ
ش	shin	sh	ـش	ـشـ	شـ
ص	sad	s	ـص	ـصـ	صـ
ض	da	d	ـض	ـضـ	ضـ
ط	ta	t	ـط	ـطـ	طـ
ظ	dha	dh	ـظ	ـظـ	ظـ
ع	'ain	'a, 'u, 'i, '	ـع	ـعـ	عـ
غ	ghain	gh	ـغ	ـغـ	غـ
ف	fa	f	ـف	ـفـ	فـ
ق	qaf	q	ـق	ـقـ	قـ
ك	kaf	k	ـك	ـكـ	كـ
ل	lam	l	ـل	ـلـ	لـ
م	mim	m	ـم	ـمـ	مـ
ن	nun	n	ـن	ـنـ	نـ
ه	ha	h	ـه	ـهـ	هـ
و	wau	w, u	ـو	ـوـ	وـ
ي	ya	y, i	ـي	ـيـ	يـ

Table 8. Moon and sun letters (Adapted from [74]).

Mon Latter				Sun Latter			
1	أ	8	خ	1	ت	8	ش
2	ب	9	ف	2	ث	9	ص
3	غ	10	ع	3	د	10	ض
4	ح	11	ق	4	ذ	11	ط
5	ج	12	ي	5	ر	12	ظ
6	ك	13	م	6	ز	13	ن
7	و	14	هـ	7	س	14	ل

The aforementioned features make the Arabic text more appropriate for hiding secret information. The various methods that have been employed in the literature are the dot method, diacritics, kashida, Unicode, sharp-edges, poetry, and hybrid methods. Each method is examined below.

4.1. Dot Method

Some early studies have used the points in the Arabic and Persian letters for hiding confidential information. For illustration, the authors of [75] hid one secret bit (0 or 1) within Arabic letters by shifting the dots. The secret message was converted to the bitstream, which was then compressed to reduce the bitstream's length. The cover text was scanned letter by letter to identify the pointed letter. Whenever a doted letter was identified, its dot was slightly shifted upward if the mystery bit was "1." Otherwise, the point was unchanged if it was "0." Figure 3 shows an example of the Arabic letter "Noon."

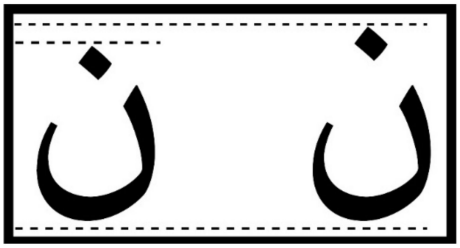

Figure 3. Vertical point shifting (Adapted from [75]).

Another study was carried out by the authors of [76] for Arabic letters with more than one point. Every multipoint alphabet was dealt with in two bits in the proposed study. The embedded process, combined with vertical point shifting, doubled the number of hidden bits. A challenging problem associated with this method is the retyping process, which destroys all the concealed bits. The authors suggested a solution to this issue that would restrict the number of new font format changes in the future. This was accomplished by merging all the data into a single file.

This approach assumes that the shifting point improves the capacity using the traditional points in Arabic letters and decreasing the hidden information's suspicion in the covert text. However, this approach is characterized by higher running time. Also, it has a fixed output format, and the secret message is vulnerable to retyping or scanning. Table 9 summarizes the dot method.

Table 9. A summary of the reviewed articles on the dot method.

Authors	Methodology	Pros	Cons
[75]	A pointed character moves its point to conceal "1" and remains untouched to hide "0."	Improves the robustness by changing the remaining characters randomly. Enhances the capacity by compression.	High computational time. Stego text is fixed for only 1 font type.
[76]	A pointed character shifts its point and increases the distance between its dots to hide 2 secret bits in one character.	Converts stego file (text) to image file to overcome the retyping challenge.	

4.2. Diacritic Method

The Arabic language uses varying marks or diacritics (Arabic redundant characters) known as harakats to represent vowel sounds. Using diacritics for security purposes is beneficial because diacritics exist naturally as a fundamental characteristic of Arabic language scripts [77,78]. Diacritics are used to differentiate between words with the same alphabets so that each word is pronounced differently, as explained in Table 10. Fathah is used to hide the bit '1' while the rest of the diacritics embed a 0 bit. This is because Fathah accounts for almost half of the diacritics' usage in Arabic texts. This approach has the flaw of attracting the reader's attention.

The early method presented by the authors of [79] exploits eight varying diacritical symbols to conceal mystery message. Fully diacritic Arabic texts are utilized as cover media. The first bit of a secret message is compared with the initial diacritic in the cover media. For instance, if the first secret bit is 1 and the initial diacritic is a Fathah, the diacritic remains on the cover media. Then, the index is incremented for both the cover media and the embedded text. If the first diacritic is not a Fathah, it is taken out of the cover media. A repetition of the approach is done until the next Fathah is realized. A secret 0 bit is used or embedded in a similar approach for the remaining seven diacritics (i.e., with the exemption of the Fathah).

In the same line, the uathors of [80] changed the diacritic's font style to cover the secret data. A new font style set was designed to embed "1" or unset to embed "0." The idea involved two approaches: The textual approach and the image approach. The textual approach chooses a font that hides extra (or all) diacritic marks completely. It then uses any encoding scenario to conceal secret bits in an arbitrary number of repeated but invisible diacritics. On the other hand, the image approach selects one of the fonts that slightly darkens multiple occurrences of diacritics. This approach needs to convert the document into a picture form to facilitate printing.

Two steganography algorithms for Arabic script were presented by the authors of [81]. The algorithms were designed based on the wasting/nonwasting property of the Arabic diacritics. In the first algorithm, a fixed-sized block parsing is used. A stream of binary bits is parsed into cover blocks. The second algorithm uses the variable size content-based approach. Here, binary data is parsed into an integer number of blocks irrespective of the number of bits they possess. These algorithms have different properties and are thereby suited for various application types as well as steganography requirements (i.e., robustness, file size, and capacity). In contrast to the content-based algorithm, the fixed size algorithm permits a straightforward computation of the required quantity of cover text. Still, it cannot instantly predict the output's file size.

Concealing Chinese text inside Arabic text was introduced by the authors of [82]. Characters of messages are automatically converted to capital letters of the English alphabet. Letters, numbers, or special characters can be hidden using two diacritics. In this case, the Unicode used warrants that each letter or diacritic is 16 bits in length. Thus, two tables (diacritics and elements tables) are used. The diacritic table has 64 inputs and

8 different diacritics, of which 2 diacritics carry 1 element. The element table is stored as a one-dimensional array and contains all the English alphabets and numbers 0 to 9.

The authors of [83] described an embedded method for hiding information in vocalized Arabic text. The method uses fully diacritic text, and if the secret bit is "1," then the diacritic is presented as it is. Otherwise (if it is "0"), the diacritic is removed.

On Arabic and Urdu text, the authors of [84] employed reversed Fathah to represent the document's concealed message. From the article written in the Arabic language, the hidden message was read and matched by character to the cover article. Then, the reversed Fathah was embedded in varied lines. The disadvantage of this method is that the text can be lost during retyping, and only one font possesses a static frame. However, this method can be applied to other similar scripts, such as Urdu. Perhaps its use can be considered in Asian scripts.

The shifting of harakat was considered by the authors of [85]. The authors applied vertical shifting by 1/200 inches to hide "1" and no change to hide "0."

Showing or omitting diacritics have been used as techniques used to hide the secret bit [86]. Three embedding algorithms were developed: The Basic algorithm, Switch algorithm, and Parity algorithm. The Basic algorithm shows a diacritic to hide the secret bit "1" while it omits a diacritic to hide "0." In the Switch algorithm, a diacritic is shown just when there is a change in the secret bits from "1" to "0" and vice versa in the secret bitstream sequence. The Parity algorithm sets a parity bit to every cover character in the text. If the cover character's position is an even number, then the parity bit of this character is "0." Otherwise, it is "1."

Two diacritics (Kasrah and Fathah) were utilized to design an embedding algorithm in [87] by fragmenting the hidden message into two arrays of binary values, forming odd and even lists. The general idea is that the odd array list is hidden in the Fathah diacritics while the even array list is concealed in the Kasrah diacritics. The first odd bit of the hidden message is read by the program and compared with the initial Fathah in the cover text. For instance, if the initial odd bit to be concealed is "1′," the initial Fathah will not be touched. Otherwise, if the initial odd bit to be concealed is not "1," the Fathah will be removed.

Recently, a modified Fathah in Arabic text steganography was presented by the authors of [88]. First, the secret message was encrypted with the AES algorithm. Then, text steganography with modified Fathah was used to hide the encrypted data. The modified Fathah lies in the same direction as the original Fathah, slightly oriented to be like the original to avoid suspicions.

The discussed diacritics method is summarized in Table 10. It can be concluded that most of the diacritics methods serve to enhance capacity. This is attributed to the benefits of diacritics' natural presence as historical characteristics of the Arabic language that originated for representing vowel sounds [78,86]. Nevertheless, diacritic methods increase suspicion since diacritics appear abnormally. Moreover, most of the Arabic scripts nowadays have no diacritics.

Table 10. A summary of the reviewed articles on the diacritics method.

Authors	Methodology	Pros	Cons
[80]	Multiple embedding scenarios are achieved by changing the font style of diacritics. It considers repeated but invisible diacritics.	Low computational time. Embedding is automated or manual. Improved invisibility. Improved security using RLE.	Stego text is fixed for the use of only 1 font type.
[82]	Hides each character in 2 diacritics.	Low computational time. Embedding is automated or manual. Stego file has a flexible format.	
[79]	The existence of Fathah hides "1," and the other diacritics hide "0."	Low computational time. Embedding is automated or manual. Stego file has a flexible format.	The stego text size is different from the cover text and raises suspicions.
[81]	Adds 1 diacritic to hide "1."		
[83]	Removes the diacritics to hide 0.		
[86]	Removes the diacritic to hide "1" or shows the diacritic when there is a switch between "0" and "1" in a sequence of bits.		
[84]	Reversed Fathah hides "1," and no change hides "0."	Embedding is automated or manual. Stego file has a flexible format.	High computational time.
[85]	Vertical shifting by 1/200 inches hides "1" and no change hides "0."		
[87]	Fathah hides "1" on the odd list, and Kasrah hides "1" on the even list.		
[88]	Change the direction of the original Fathah to embed 1 secret bit.	Improved security using AES.	

4.3. Kashida Method

Kashida refers to a type of justification, i.e., a stretch or extension of Arabic letters. It is used for various purposes such as emphasis, legibility, aesthetic, and justification [89]. In this steganography method, the extension (kashida) is added to words to represent the secret bit "1." When it is not added, it represents the secret bit "0." It is worth noting that alphabet extensions do not affect the writing content or the message content. Although the sentence in the output text still has the same meaning as the cover text, i.e., "It is from the excellence of (a believer's) Islam that he should shun that which is of no concern to him.", the appearance of the text changes and increases the file size. Thus, it may capture the reader's attention.

The Kashida method's established state, which protects the secret bit in any letter, was performed by the authors of [90]. It needs the pointed letters with extension to hide secret bit "1" and the unpointed letters with extension to cover secret bit "0." This method does not have any effect on the written content, as illustrated in Table 11. The improved work by the authors of [90], described in [91], involved injecting one kashida to hides "0" and employing two consecutive kashidas to conceal "1."

Table 11. A steganography example that adds extensions after letters (Adapted from [91]).

Watermarking Bits	110010
Cover-text	من حسن اسلام المرء تركه مالا يعنيه
Output text	من حسن اسلام المرء تركـه مـالا يـعنيه 1 1 0 0 1 0

Building on this method, a stego system for Arabic e-text, Maximising Steganography Capacity Using Kashida in Arabic Text (MSCUKAT), was developed [92]. At the same time, the algorithm proposed by the authors of [77] hides the secret message as numbers by inserting kashidas. Each extendable letter can hide a specific number based on the position and the number of kashida in a word. Later, the implementation of MSCUKAT was produced by the authors of [93].

The algorithm proposed by the authors of [73] considers four scenarios where kashida letters can be added. Techniques are employed at random for selecting one of the four scenarios in each round. Then, message segmentation principles enable the sender to select over one strategy for each message block.

Similar to the method described by the authors of [73], four embedding schemes were designed by the authors of [94] to hide two secret bits. The suggested design utilizes the existence of kashida after a pointed or unpointed letter to hide the secret bits.

Next, the authors of [95] compressed secret messages using Gzip and encrypted these compressed secret messages by deploying AES. The proposed embedding method involves four stego options: Pointed kashida (After Letters), pointed kashida (Before Letters), pointed kashida (Mixed Letters), and MSCUKAT.

Another work [96] hid a voice file into a text file using kashida and the word "La." The proposed embedding algorithm reduces the size of the secret voice using the Loss-Less compression algorithm. It then hides "1" by inserting kashida after the letters, while "0" is hidden by leaving the letters without kashida insertion.

Using the sun and moon letters, a technique proposed by the authors of [67] protects a secret bit in Arabic script. The technique considers four different scenarios. In the first, a kashida is placed next to a sun letter to conceal the confidential bits "00." The second scenario covers the sensitive bits "11" by inserting two kashidas after a sun letter. A kashida is added after a moon letter in the third scenario to embed the secret bits "01," and two kashidas are included in the fourth scenario to conceal the secret bits "10."

As kashida is frequently used in Arabic text, the utilization of kashida for steganography (Table 12) is one way of improving the embedding capacity of hidden information. Nevertheless, these studies still have drawbacks, such as high imperceptibility to suspicion and large output file size. Also, there are only a few attempts to reduce the algorithm's complexity for improving the extraction of hidden information.

4.4. Unicode Method

Unicode is an international character encoding format for displaying text for data processing. This standard is compatible with ISO/IEC 10646-1:2000 version 2. ISO/IEC 10646 has the same characters and codes. Unicode allows the encoding of all characters used in the world's writing systems. This standard employs 16-bit encoding, which allows for a total of 65,000 characters. This implies that it is possible to specify and define 65,000 characters in different modes such as numbers, letters, and symbols in various languages. Furthermore, due to the vast amount of space devoted to characters, this standard contains the majority of the symbols needed for high-quality typesetting. The languages whose writing systems can be supported by this standard are Latin (covering most of the European languages), Cyrillic (Russian and Serbian), Greek, Arabic (including Arabic, Persian, Urdu, Kurdish), Hebrew, Indian, Armenian, Assyrian, Chinese, Katakana, Hiragana (Japanese), and Hangeul (Korean). This standard also includes several mathematical and technical symbols, punctuation marks, arrows, and other marks. The Unicode standard consists of two groups of codes for the Arabic alphabets. The first is the representative code, and the second is the code of the letter's possible shapes. Separate characters are allocated for Persian letters with semantics or shapes that are significantly different from Arabic letters despite the unification of codes with common characters. This implies that separate places have been allocated to Persian special letters (پ،چ،ژ،گ) and two other Persian letters (ک, ی) that are different from their corresponding Arabic letters in terms of appearance. The Unicode approach utilizes the various possible Unicode values of the same alphabet to

conceal the bits. It is suitable for use on the public channel and modern devices such as smartphones.

Table 12. A summary of the reviewed articles on the kashida method.

Authors	Methodology	Pros	Cons
[90]	Pointed letters with kashida hide "1." Unpointed letters with kashida hide "0."	Embedding is automated or manual. No size increase of stego text.	Limited capacity since all letters cannot be extended.
[91]	Uses 1 kashida hide "0" and 2 consecutive kashidas to hide "1."		
[92]	Inserts kashida wherever applicable to hide "1."		Increase in the size of stego file.
[77]	Add kashida in a specific location to hide secret numbers.		
[93]	Inserts 1 kashida to hide "0" and 2 consecutive kashidas to hide "1."		
[74]	Uses 5 scenarios to hide 2 secret bits by inserting kashida after the moon or sun letters.		
[73]	Randomly applies kashida insertion in 4 scenarios to hide a secret bit.	Increases the algorithm complexity and reduces the likelihood of suspicions.	
[95]	Uses 4 choices to protect the secret bits based on kashida and dotted letter.	Improves security using AES. Improves the capacity using Gzip. Stego file has a flexible font and format. Low computational time.	
[96]	Compresses the secret message then inserts extra kashida after each letter and "La" word to hide "1." It leaves the letter with the original kashida to hide "0."	Reduces the size of secret bits using Loss-Less compression algorithm.	
[94]	Kashida-based insertion in 4 scenarios while considering pointed and unpointed letters for hiding 2 secret bits.	Improves the capacity using 1 character to hide 2 secret bits.	High computational time.

The authors of [97] proposed the usage of Unicode characters by inserting a normal space after pseudo-pace to embed "1" and no insertion to embed "0." In that same year, the authors of [98] presented a design that utilizes "La" to hide the secret message. The word has two forms in Arabic writing: Normal form and special form. The Unicode of the normal form is used to conceal "0," and the special form conceals "1."

Later, another Unicode technique was suggested where each Persian or Arabic letter has one unique code [99]. This code displays the letter in an isolated form and acts as a representative for the word. For each word in the text, it is possible to save a letter using the representative letter or the code of its correct shape (with respect to its position in the word). For hiding 0 bits in the word, the first option is used to save the word.

Similarly, for hiding 1 bit, the second option is used. The authors of [100] applied the similarity between Arabic and Persian characters « ک» and « ی» to hide the bit "0." The Arabic characters « ك» and « ي» were applied to hide bit "1." One approach used the isolated letters in Arabic text with Run Length Encoding (RLE) to embed the secret bits [101], where the secret bit streams are converted to groups of 0s and 1s by applying RLE. The Unicode characters related to the isolated letters are changed to embed the secret bit "1" or unchanged to embed the secret bit "0."

Likewise, the isolated letters in Urdu text [102] changed the Unicode character to hide the secret bits. The secret message is encrypted as an encipher text by applying RSA. The enciphered text is converted to bitstream and divided into even blocks. Randomization and swapped functions are then applied to these blocks. The Unicode equivalents of the isolated letters are changed to cover bit "1" in each block or unchanged to cover "0."

Another Unicode system that involves a modified RLE was presented by the authors of [78]. This system uses a coding method with an output that carries a sequence of 1s

and a few 0s. The modified RLE proposed in the system is suitable for compression. The outputs suit steganography purposes that use Unicode and unprinted characters to hide the secret message in an Arabic text.

Three scenarios were studied by the authors of [103]. The first reduces the character change by counting the number of 0s. If there are fewer 0s than 1s, the secret packet is complemented with 1s. The second scenario hides 0 bits by leaving a cover letter unchanged from the next word in the text, while the third hides 1 bit by identifying a cover letter from the text. According to the type of letter, this letter's Unicode must then be modified from the general letter's Unicode to the contextual Unicode. The cover letter's Unicode is changed to the isolated form if it belongs to an isolated group. If the cover letter is part of a series, the Unicode is reverted to the original form.

As seen in Table 13, high perceptual transparency and the unaffected format size and output file are the major benefits of Unicode methods. The Arabic letters take different shapes in different positions. For this reason, an inappropriate change of the letter's shape increases the reader's suspicion, hence limiting the cover letters.

Table 13. A summary of the reviewed articles on the Unicode method.

Authors	Methodology	Pros	Cons
[97]	Inserts normal space after pseudo space to hide "1." No insertion hides "0."	Stego file has a flexible font and format. Low computational time.	Increase the stego file size.
[99]	The shape's code hides "1" and representative letter's code hides "0."		Deficient capacity as the limited identical isolate letter between Arabic and Persian.
[100]	The Arabic character « ك» or « ي» hides "1" and the Persian character « ک» or « ی» hides "0."		Low capacity due to the limited use of « ك» and « ي»
[98]	The special form of the word "La" hides "1" and the normal form hides 0.	Embedding is automated or manual. Stego file has a flexible font and format. Low computational time.	Very low capacity because of the poor existence of La word.
[101]	Changes the Unicode of Arabic isolated letter to hide "1" and leaves it unchanged to hide 0.	Improves security using RLE. Stego file has a flexible font and format. Low computational time.	Low capacity due to the limited appearance of Arabic isolated letter.
[102]	Hides the secret bits based on Unicode and non-printed characters.	Improves security using RSA. Stego file has a flexible font and format. Low computational time.	Low capacity due to the limited appearance of Urdu isolated letter.
[78]	Changes the Unicode of isolated Urdu letter to hide "1" and leaves it unchanged to hide 0.	Improves the security using RLE. Stego file has a flexible font and format. Low computational time.	Limitation in capacity by considering only unpointed letters.
[103]	Uses 3 scenarios to hide secret bit by changing the Unicode of the letter.	Stego text size is not changed.	Limitation in capacity by considering only isolated and initial letters.

4.5. Sharp Edges Approach

The algorithms suggested by the authors of [104] involve using Arabic letters' sharp edges to hide the secret bits. Each letter hides secret bits based on the number of its sharp edges, as shown in Figure 4. For example, a letter with one sharp edge is probable to embed one secret bit 0 or 1. The authors of [104] designed a reference table to keep the locations of secret bits in the cover letter.

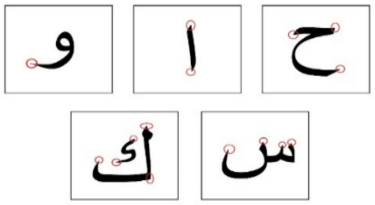

Figure 4. Arabic characters' sharp edges (Adapted from [105]).

The approach described by the authors of [105] operates on dotted and undotted alphabets. Random numbers are generated and used to assign alphabets sufficient for hiding 104 bits of secret message. This results in the following alphabets: ر ا ر ن ر ص و ت ب ج ي ا ق ا م أ و ن ش س خ ى ل ي ب ل ي ه ه ا ى ل ص ل ر ا ض ن ب خ ع س ل س ن ع ق

The number of sharp edges on the initial alphabet, as shown in in Table 14, determines the number of bits that will be hidden. The secret bit that corresponds to this number is included in the code sequence, and the process continues until all the bits are embedded. For instance, the character (ر) has two sharp edges. Hence, it can carry the first two binary bits (i.e., 01) and represent them in the corresponding decimal unit, which is 1.

Table 14. Number of sharp edges for each letter (Adapted from [105]).

Number of Sharp-Edges	1	2	3	4	5
Letter	ف ق ة ه م و	ا ب ت ث ذ ض ز ي ظ ن ل ط ر د ص ى	ع ء ح غ ج خ	س ش ؤ	ك أ ا ى

The sharp edges with dots and typographical proportion of Arabic letters were presented by the authors of [106]. The presented algorithm, called the Primitive Structural algorithm, gives each letter more than one potential position to embed the secret bits, as shown in Table 15. At the same time, each letter carries more hidden bits than the method in [106].

Table 15. Number of sharp edges for each letter (Adapted from [107]).

Char	Unicode	Primitive Structural Method Entities			Num. of Potential Hiding Places
		Sharp-Edges	Dot(s)	Typographical Proportion	
أ	\U0627	2	0	1	3
ب	\U0628	2	1	3	6
ت	\U062A	2	2	3	7
ث	\U062B	2	3	3	8
ج	\U062C	3	1	5	9

From Table 16 it can be summarized that the serious limitation of the sharp edges method is security. Thus, it requires further security layers to secure the indications of the concealed bits in the stego file. Nevertheless, the sharp edges method achieves better embedding capacity.

Table 16. A summary of the reviewed article on the sharp edges approach.

Authors	Methodology	Pros	Cons
[104]	Each sharp edge in character embeds one bit, "0" or "1." Generates reference table for secret bit's place.	High capacity because all characters can hide bits based on their sharp edges. Stego file has a flexible font and format.	Low security as additional security layers are needed to protect the reference table or code sequence.
[105]	The number of sharp edges in each character is utilized to protect the same number of secret bits by converting it to decimal. It generates a code sequence of decimal numbers.		
[106]	Each sharp edge, dot, and typographical proportion can hide "0" or "1."		

4.6. Poetry Approach

The Arabic poetry system is designed to be operated in text hiding [107]. Since there is a representation of binary units embedded in each Arabic poem, poems can be utilized to hide secret bits. The key idea here is to presume that the embedded binary bits position in poems contain secret bits. The real secret bit is either equivalent to the binary position or equivalent to its reverse. To increase the capacity of the introduced embedded technique, diacritics and kashida approaches have been utilized. Table 17 shows an example of Arabic poetry steganography.

Table 17. An example of a verse of an Arabic poem and how it is classified into poetry meters (Adapted from [107]).

The verse	وألا أ رى غيري له الدهرَ مالكا				ولي وطنٌ آليت ألا أبيعَهُ			
How it is pronounced	رمالكا	لهدده	أرى غيري	وألا	أبيعهو	ت ألا	طنن ءالي	ولي و
The corresponding feet	مفاعلن	فعولن	مفاعيلن	فعولن	مفاعلن	فعولن	مفاعيلن	فعول
Binary representation	011011	01011	0101011	01011	011011	01011	0101011	1011
Its classification	Al-Taweel meter							

Table 18 shows that the poetry method improved the embedding capacity. However, it is applicable only for Windows-1256.

Table 18. A summary of the reviewed article on the poetry approach.

Author	Methodology	Pros	Cons
[107]	This method represented the poetry meters into binary representation to hide the secret bits.	Improved the embedding capacity	Only used of Windows-1256 for the encoding.

4.7. Hybrid Approach

A combined or hybrid method involves the integration of two or more text steganography methods. The earliest proposal in this method [108] merges two methods: The Unicode method (whitespace) and the kashida method. The integrated technique embeds secret bit "1" by inserting whitespace. Before moving to the next word, it adds two consecutive whitespaces between words to hide "1." In the case of secret bit "0," there is no addition of kashida and whitespace.

Later, merging Unicode with diacritics was suggested [109] to hide the confidential message. This method employs RNA to encode the secret messages, while non-printed

characters are used to conceal these codes. Compression is applied by modifying the Run Length Encoding (RLE) compression algorithm to overcome its limitation.

Similarly, the authors of [110] compressed secret messages using Gzip and encrypted these compressed secret messages by deploying AES. The embedding method employs two stego options of "kashida" and changes the Unicode of the letter based on the proposed blood group algorithm's behavior.

Next, kashida and diacritics were combined [111] to cover the mystery message. The embedding algorithm conceals one part by adding Fathah, and the rest hides "0." The other part adds two consecutive kashidas to hide "1" and one kashida to hide "0."

Again, kashida and Unicode methods were used by the authors of [112] to cover the confidential information. For the Unicode method, three small spaces (thin, hair, and Six-PRE-EM) were utilized. The presented scheme grouped the bitstream into 4 bits each. The first bit indicates kashida, where it inserts kashida to hide "1" and considers an existing kashida to hide "0." The second bit indicates thin space, the third shows hair space, and the last bit indicates Six-PRE-EM. The existence of the three small spaces hides "1" while their absence hides "0."

The merger of counting-based secret sharing and kashida that was recently presented by the authors of [113] hides the secret sharing bits within Arabic text using kashida. Recently, Medium Mathematical Spaces (MSPs), ZWJ, JWJN and kashida were united [114] to protect the secret bits. The study hid one secret bit by changing format or including one whitespace or kashida.

It can be observed from Table 19 that hybrid Arabic text steganography methods, using kashida and several kinds of spaces to hide the secret bit, have recently received more attention. The main advantages of merging two or more steganography methods include a higher capacity for hidden information and a lower suspicion level. However, the hybrid scheme inherits drawbacks from its composite methods, such as the destruction of hidden information by retyping or deduction by Optical Character Recognition (OCR).

Table 19. A summary of the reviewed articles on the hybrid approach.

Authors	Methodology	Pros	Cons
[108]	Adds kashida and consecutive whitespace to hide "1" and single normal space to hide "0."	Slight improved in capacity using whitespace.	Stego file increases by inserting kashida and whitespace.
[109]	Hide 1 secret bit by changing the Unicode of the unpointed isolated letter and add diacritics.	Improves security using RNA. Stego file has a flexible font and format. Low computational time.	Used only the unpointed letters.
[110]	Different scenario merges kashida and Unicode methods based on blood group behavior.	Improves security using AES. Enhances the capacity by increasing the usable characters and using a compression algorithm (Gzip).	Stego file size increases by inserting kashida.
[111]	Inserts kashida or fathah to hide "1"; the rest to hide "0."	Improves the capacity by increasing the embedding characters.	Stego file size increases by inserting kashida and diacritics. Suspicion is raised because the included diacritics are not in the proper place.
[112]	Integrates kashida with 3 small spaces (Thin, Hair and Six-PRE-EM) to hide secret bits.	Inserting kashida and whitespaces are controlled, which enhance the capacity while maintaining invisibility.	Stego file size increases by inserting kashida and whitespace.
[113]	Hybrid kashida with secret sharing.	Improves security using secret sharing.	Stego file size increases by inserting kashida.
[114]	Multi types of whitespace are combined with kashida.	Improves the capacity by increasing the usable characters.	Stego file size increases by inserting kashida and whitespace.

The use of Arabic text steganography methods is illustrated in Figure 5, which demonstrates that the Unicode method has attracted the most significant interest among researchers due to its transparency. Besides, the diacritics methods suffered from several limitations, mainly increased suspicion (Low invisibility). The diacritics are exploited remarkably by either showing some and hiding the others or adding it in impropriate positions.

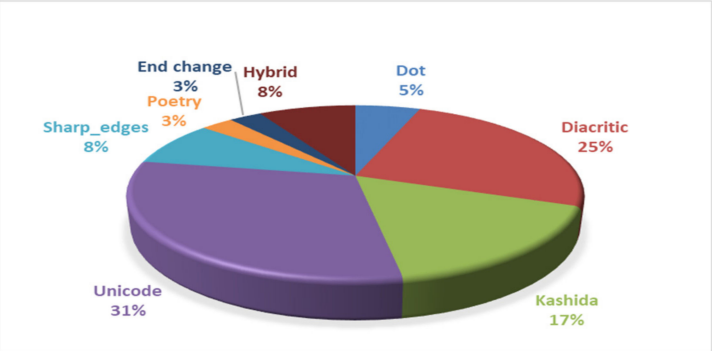

Figure 5. Usage of Arabic text steganography methods.

5. Evaluation Criteria

Four evaluation criteria must be considered when a researcher develops a text steganography [4], i.e., capacity, invisibility, robustness, and security, where some of these criteria can be evaluated by calculation while others can be visualized [115]. High embedding performance is achieved by making a tradeoff between these criteria [116]. Most text steganography methods 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 method's security, especially when its security depends on properties such as invisibility and robustness.

5.1. Embedding Capacity

The amount of hidden information in the cover text is called its embedding capacity [117]. This criterion is calculated by applying the following equation.

$$\text{Embedding Capacity} = \frac{\textit{Secret bits}}{\textit{Cover bits}} \times 100$$

Text steganography methods can improve the embedding capacity by increasing the embeddable/usable cover characters (embeddable positions) [56,118], bits per location [115], and compression techniques [43], as well as merging more than one text feature [114]. The embeddable character/location refers to a character/location with the ability to use in the embedding process. Bits per location indicates the hiding amount per location.

Arabic text has many features such as kashida, dots, diacritics, and so on, increasing the capacity by merging more than one feature. Besides, compressing the secret bits using compression algorithms reduces the hidden bits amount. This paper evaluates the capacity of Arabic text steganography methods by analyzing the bit per location and the compression techniques used.

5.2. Invisibility

Steganography hides the secret message in the cover media without making perceptible or visible distortions on the cover item [119]. Thus, the hidden message is not detectable by the attacker. Some researchers have not considered imperceptibility to be a basic requirement of steganography [120]. However, most other researchers have emphasized imperceptibility as one of the primary goals to protect the hidden message [17,43,56,121–126]

("protection by invisibility"). We are inclined to take the latter perspective—invisibility is the key properties to prevent the attacker from detecting and hence deducting the hidden message [127–130].

According to the authors of [131], a stego file can be attacked in two ways: A visual attack and statistical attack. The visual attack uses the human vision to detect any abnormal appearance on the object or distinguish the differences between the original object from the stego object, whereas the statistical attack analyzes the item using steganalysis algorithms based on mathematical theories [132,133].

Some researchers view perceptible modification on the cover media as a disadvantage. The original cover media is not secret and is available for the public. One way to detect the hidden message is to check the similarity between the original cover and the stego file. The similarity depends on the type of the cover. Accordingly, the unseen change in the cover text cannot be evaluated by calculations or numeric computation. This is because the human vision is different from one human to another. Nevertheless, the difference between two texts can be measured using Jaro–Winkler distance [134,135] by checking the size, semantic, and lexical of the texts. Thus, any imperceptible modification can be detected using this parameter. Therefore, a successful text steganography should achieve a high similarity between the original and the cover file. However, Jaro–Winkler distance has a limitation in font attributes.

Invisibility can be divided into two types: Similarity and ambiguity. Similarity is achieved when the two texts being compared are identical in size, format, semantic, and lexical. It can only be measured in the availability of both the cover and stego text. It is assessed in three levels (low, medium, high) that illustrate how close the two documents are. Ambiguity arises when a word/diacritic is not in the text's context and when inserting multiple kashidas in irregular manner. It can be evaluated when the attack is found only the stego text and not in the cover text. Ambiguity can be assessed in three levels (low, medium, high) that demonstrate the extent of attracting the eavesdropper's attention. This study analyzes invisibility by studying the examples of cover and stego texts, which are given in the mentioned methods in terms of similarity and ambiguity.

5.3. Robustness

Robustness in steganography is the ability of the hidden data to withstand attacks, such as rotation, cropping, added noise, compression, and so on [120]. Considering that a vast amount of text messages and contents are transmitted over and posted on the Internet, robustness is now becoming more relevant. This has also been noted by many researchers [17,119,122,136–139]. Moreover, tampering attack is the most common type of text attack [2,4,18,44]. It can take many forms, such as insertion/deletion, copy/paste, font format, printing, and retyping [8,115]. Besides, attackers can use OCR to identify different characters from a record picture. Additionally, a tampering attack provides full alphanumeric recognition of printed or handwritten characters, numerical letters, and symbols into a computer processable layout that includes ASCII and Unicode [140].

5.4. Security

Security in steganography conceals a high amount of secret information whilst maintaining the invisibility and robustness [4]. The proposed text steganography algorithm must prevent the attacker from visually detecting the hidden information, destroying it by tampering or extracting it by breaking (understanding) the embedding algorithm. Security criterion is influenced by invisibility and robustness criteria. Invisibility prevents the eavesdropper from distinguishing the hidden information in the stego text. At the same time, robustness prevents the attacker from tampering the hidden message. The security of the modern text steganography method can be defined as its ability and strength to resist any attack to remove or destroy the hidden data [116,141]. It is achieved by increasing the level of algorithm complexity, such as random or non-sequence embedding positions [106,142], randomly selecting secret bits [143–145], and generating a strong stego

key [116,121,144–148]. As a result, using one or more of these complexity techniques makes the hidden data extremely difficult to extract. Besides, the aforementioned, researchers found that the importance of reinforcing text steganography and cryptography methods lies in protecting the secret keys, which are considered the most critical element in information security technologies [35,37,149]. However, only a few researchers have used an encryption to improve the efficiency of their methods [150,151].

5.5. Evaluation of Arabic Text Steganography Methods

This paper presents the evaluation of Arabic text steganography methods based on the capacity, security, robustness, and invisibility criteria in Tables 20–26 and summarized below:

- **Dot method:** Although this method enhances invisibility, it is less robust, as the hidden message may be lost if the font format is changed. In addition, the method does not consider the encryption or non-sequence embedding positions to prevent the hidden bits from extraction. Despite using compression, the capacity is still low, with the maximum embedded bits per location are two.
- **Diacritics method:** Low invisibility is the major drawback of this method. The cover text and stego text are not identical, and the stego text has many ambivalences in using diacritics. Besides that, capacity is still low, where the bits per location ranges from one to four. Whereas this method partially enhances the robustness of the stego text, the secret message is not encrypted and not embedded in non-sequence positions.
- **Kashida method:** This method is resistant against the copy-paste action but has downsides in terms of capacity, invisibility, and robustness.
- **Unicode method:** The embedding capacity of this method is decreased even though compression is used in some techniques. However, this method accomplishes high invisibility. The robustness is improved in copy-paste action, font format, and OCR, but encryption and non-sequence embedding are not considered.
- **Sharp edges method:** This method achieved high invisibility, capacity, and robustness. Adding non-sequence embedding adds more complexity to protect it from the attacker.
- **Poetry system method:** The method has high invisibility and higher robustness. However, its hiding capacity is limited.
- **Integrated method:** The primary goal of merging methods is to improve performance and overcome the previous methods' limitations. Nevertheless, if the methods are not integrated properly, then these drawbacks are inherited.

Table 20. Evaluation of dot method based on invisibility and robustness and capacity.

Authors	Capacity		Security		Robustness						Invisibility	
	Bits/Location	Compression	Encryption	Sequence Embedding	Insertion/ Deletion	Copy/Paste	Font Format	OCR	Printing	Retyping	Similarity	Ambiguity
[75]	1	✓	✗	✓	✗	✗	✗	✗	✓	✗	High	Low
[76]	2	✗	✗	✓	✗	✗	✗	✗	✓	✗	High	Medium

Table 21. Evaluation of diacritics method based on invisibility and robustness and capacity.

Authors	Capacity		Security		Robustness						Invisibility	
	Bits/Location	Compression	Encryption	Sequence Embedding	Insertion/ Deletion	Copy/Paste	Font Format	OCR	Printing	Retyping	Similarity	Ambiguity
[80]	1	✗	✗	✓	✗	✓	✗	✓	✓	✗	low	High
[82]	1	✗	✗	✓	✓	✓	✓	✓	✓	✓	low	High
[79]	4	✓	✗	✓	✓	✓	✓	✓	✓	✓	low	high
[81]	4	✗	✗	✓	✓	✓	✓	✗	✗	✗	Low	High
[83]	1	✗	✗	✓	✓	✓	✓	✗	✗	✗	High	Medium
[86]	1	✗	✗	✓	✓	✓	✓	✓	✓	✓	low	high
[84]	1	✗	✗	✓	✓	✓	✓	✓	✓	✓	Medium	Medium
[85]	1	✗	✗	✓	✓	✓	✗	✗	✗	✗	Medium	High
[87]	1	✗	✗	✓	✓	✓	✓	✓	✓	✗	Medium	Medium
[88]	1	✗	✓	✓	✓	✓	✓	✗	✗	✗	High	Low

Table 22. Evaluation of kashida method based on invisibility and robustness and capacity.

Authors	Capacity	Security			Robustness						Invisibility	
	Bits/Location	Compression	Encryption	Sequence Embedding	Insertion/ eletion	Copy/Paste	Font Format	OCR	Printing	Retyping	Similarity	Ambiguity
[90]	1	✗	✗	✓	✗	✗	✓	✓	✓	✗	High	Low
[91]	1	✗	✗	✓	✗	✗	✓	✗	✗	✗	Low	Medium
[92]	1	✗	✗	✓	✗	✗	✓	✗	✗	✗	Low	High
[77]	1	✗	✗	✓	✗	✗	✓	✗	✗	✗	Low	High
[93]	1	✗	✗	✓	✗	✗	✓	✗	✗	✗	Medium	High
[73]	1	✗	✗	✓	✗	✗	✓	✗	✗	✗	Medium	High
[95]	1	✓	✓	✓	✗	✗	✓	✗	✓	✗	Medium	Medium
[96]	1	✓	✗	✓	✗	✗	✓	✗	✗	✗	Medium	High
[74]	2	✗	✗	✓	✗	✗	✓	✗	✗	✗	Medium	Medium
[94]	2	✗	✗	✓	✗	✗	✓	✓	✓	✗	High	Low

Table 23. Evaluation of Unicode method based on invisibility and robustness and capacity.

Authors	Capacity	Security			Robustness						Invisibility	
	Bits/Location	Compression	Encryption	Sequence Embedding	Insertion/ Deletion	Copy/Paste	Font Format	OCR	Printing	Retyping	Similarity	Ambiguity
[97]	1	✗	✗	✓	✓	✓	✓	✓	✗	✗	Medium	Low
[98]	1	✗	✗	✓	✓	✓	✓	✓	✗	✗	High	Low
[99]	1	✗	✗	✓	✓	✓	✓	✓	✗	✗	High	Low
[100]	1	✗	✗	✓	✓	✓	✓	✓	✗	✗	Medium	Low
[101]	1	✓	✗	✓	✓	✓	✓	✓	✗	✗	High	Low
[102]	1	✗	✓	✓	✓	✓	✓	✓	✗	✗	High	Low
[78]	1	✓	✗	✓	✓	✓	✓	✓	✗	✗	High	Low
[103]	1	✗	✗	✓	✓	✓	✓	✓	✗	✗	High	Low

Table 24. Evaluation of sharp edges method based on invisibility and robustness and capacity.

Authors	Capacity	Security			Robustness						Invisibility	
	Bits/Location	Compression	Encryption	Sequence Embedding	Insertion/ Deletion	Copy/Paste	Font Format	OCR	Printing	Retyping	Similarity	Ambiguity
[104]	3	✗	✗	✓	✓	✓	✓	✓	✓	✓	High	Low
[105]	6	✗	✗	✓	✓	✓	✓	✓	✓	✓	High	Low
[106]	6	✗	✗	✓	✓	✓	✓	✓	✓	✓	High	Low

Table 25. Evaluation of poetry system method based on invisibility and robustness and capacity.

Authors	Capacity	Security			Robustness						Invisibility	
	Bits/Location	Compression	Encryption	Sequence Embedding	Insertion/ Deletion	Copy/Paste	Font Format	OCR	Printing	Retyping	Similarity	Ambiguity
[107]	3	✗	✗	✓	✗	✓	✓	✓	✓	✓	High	Low

Table 26. Evaluation of hybrid method based on invisibility and robustness and capacity.

Authors	Capacity	Security			Robustness						Invisibility	
	Bits/Location	Compression	Encryption	Sequence Embedding	Insertion/ Deletion	Copy/Paste	Font Format	OCR	Printing	Retyping	Similarity	Ambiguity
[108]	1	✗	✗	✓	✗	✓	✓	✗	✗	✗	Medium	Medium
[109]	1	✓	✗	✓	✗	✓	✓	✓	✗	✗	Low	High
[110]	1	✓	✓	✓	✓	✓	✓	✓	✗	✗	low	low
[111]	1	✗	✗	✓	✗	✓	✓	✗	✗	✗	Low	High
[112]	1	✗	✗	✓	✗	✓	✓	✗	✗	✗	Medium	High
[113]	1	✗	✗	✓	✗	✓	✓	✓	✓	✓	Medium	Medium
[114]	1	✗	✗	✓	✓	✗	✗	✓	✗	✗	✓	✓

6. Recommendations for Future Works

This study provides recommendations and opens new paths, which are highlighted as follows:

- Although some researchers have considered Arabic characters, most of them have not applied their suggested methods to social media. Meanwhile, such media are fertile environments for information hiding, as a large volume of texts is pumped on social networks every day. This volume of texts makes it difficult for the eavesdropper to specifically select any of them that may contain hidden information. Researchers can thus apply some of these methods to social media while facilitating the support for Arabic characters.
- The integration of text steganography methods improves the capacity and increases the difficulty experienced by the eavesdropper in an attempt to trace the embedding algorithm. However, these methods inherit the disadvantages of the methods that make them up. This is especially obvious in the kashida approach, which increases the stego file size which also raising the level of suspicion in specific cases. Therefore, the integration should be well studied to identify which methods achieve the desired objectives while minimizing the constituent methods' drawbacks.
- A compression method reduces the amount of hidden information, thereby increasing the capacity. It also increases the complexity of extracting the secret message from the cover text. Despite this, only a few researchers have used compression to improve the efficiency of their methods.
- A few of the proposed studies have provided solutions to enhance the protection of secret messages prior to the embedding process by combining both cryptography and steganography methods, especially for protecting the stego key. This combined method constitutes another layer of protection if the embedding algorithm is detected.
- During this survey, it was observed that the use of full diacritics text is lacking, which is the obstacle preventing the exploitation of such diacritics. It is worth noting that students of religious studies or linguistic sciences at all stages adopt this type of text. Similarly, the Quran and Hadiths scripts are omnipresent on the web and social media and widely used as references and an inference.
- Regardless of kashida's weaknesses, such as increasing the stego file size and thus increasing the suspicion to a reader, utilizing kashida in text watermarking, especially in Quranic scripts, is ideal than text steganography. In text watermarking, kashida is used to protect or copyright the text without affecting the meaning of the text, unlike the diacritics.
- Most Arabic text steganography methods suffer from low capacity because of the limited number of bits per location and usable characters. Integrated Arabic features with font attributes are used to enhance the capacity.
- The selection of the embedding positions sequentially tells the attacker the order of the secret bits. Therefore, it is imperative to propose embedding methods with non-sequence position as the additional security layer.

7. Conclusions

The importance of using the Arabic text as a cover for hiding sensitive information via public channels by governments, companies, and individuals in Arabic-speaking countries cannot be overemphasized. This is because these countries use Arabic text in their daily activities. This paper presents the research landscape of Arabic text steganography methods from its inception to date and discusses seven Arabic text steganography methods: Dot, diacritics, kashida, Unicode, sharp edges, poetry system, and hybrid. We analyzed these methods, categorized them, summarized their methodologies, and determinedtheir strengths and weaknesses. We also evaluated these methods based on the four existing objectives in any steganography method (i.e., capacity, invisibility, robustness, and security).

We found that most of the existing Arabic steganography methods suffer from low capacity because of the low bit per location and less usable characters. In terms of security, several proposed techniques integrated steganography with cryptography to provide prior protection for a confidential message. Converting the selection of embedding position from sequence to non-sequence will add a security layer to the embedding techniques. Although the Arabic language is rich in linguistic characteristics, the previous studies and existing methods have not utilized most of them to achieve high embedding performance. Consequently, this paper opens new paths in Arabic text steganography by providing recommendations for future work.

Author Contributions: Conceptualization: R.T.; methodology: R.T. and N.I.U.; validation: R.T., S.M.Y., and A.A.; formal analysis: R.T.; resources: R.T., N.I.U., R.D., and N.A.R.; writing original draft preparation: R.T.; writing—review and editing: R.T., N.I.U., S.M.Y., and A.A.; supervision: N.I.U., S.M.Y., and A.A.; funding acquisition: N.I.U. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by Universiti Putra Malaysia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

Acknowledgments: The publication of this work is funded by the UPM Publication Fund.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Azeem, M.; He, J.; Rana, K.G.; Akhtar, F. A cryptographic data hiding algorithm with high cover text capacity. *Int. J. Electron. Secur. Digit. Forens.* **2019**, *11*, 225–244. [CrossRef]
2. Malik, A.; Sikka, G.; Verma, H.K. A high capacity text steganography scheme based on LZW compression and color coding. *Eng. Sci. Technol. Int. J.* **2017**, *20*, 72–79. [CrossRef]
3. Din, R.; Utama, S.; Hanizan, S.H.; Hilal, M.M.; Hanif, M.A.M.; Zulhazlin, A.; Fazali, G.M. Evaluating the Feature-Based Technique of Text Steganography Based on Capacity and Time Processing Parameters. *Adv. Sci. Lett.* **2018**, *24*, 7355–7359. [CrossRef]
4. Ahvanooey, M.T.; Li, Q.; Hou, J.; Rajput, A.R.; Yini, C. Modern Text Hiding, Text Steganalysis, and Applications: A Comparative Analysis. *Entropy* **2019**, *21*, 355. [CrossRef] [PubMed]
5. Cohen, A.; Holmgren, J.; Nishimaki, R.; Vaikuntanathan, V.; Wichs, D. Watermarking Cryptographic Capabilities. *SIAM J. Comput.* **2018**, *47*, 2157–2202. [CrossRef]
6. Artz, D. Digital steganography: Hiding data within data. *IEEE Internet Comput.* **2001**, *5*, 75–80. [CrossRef]
7. Khosravi, B.; Khosravi, B.; Khosravi, B.; Nazarkardeh, K. A new method for pdf steganography in justified texts. *J. Inf. Secur. Appl.* **2019**, *45*, 61–70. [CrossRef]
8. Rahman, S.A.E.; Nourah, P.; Abdulrahman, B.; Arabia, S. *Text Steganography Approaches Using Similarity of English Font Styles*; IGI Global: Hershey, PA, USA, 2019; Volume 7, pp. 29–50. [CrossRef]
9. Aljawarneh, S.A.; Vangipuram, R.; Puligadda, V.K.; Vinjamuri, J. G-SPAMINE: An approach to discover temporal association patterns and trends in internet of things. *Futur. Gener. Comput. Syst.* **2017**, *74*, 430–443. [CrossRef]
10. Alsaadi, H.I.; Al-Anni, M.K.; Almuttairi, R.M.; Bayat, O.; Ucan, O.N. Text steganography in font color of MS excel sheet. In Proceedings of the International Conference on Data Science, E-learning and Information Systems, Madrid, Spain, 1–2 October 2018; pp. 1–7.
11. Kang, Y.; Liu, F.; Yang, C.; Luo, X.; Zhang, T. Color Image Steganalysis Based on Residuals of Channel Differences. *Comput. Mater. Contin.* **2019**, *59*, 315–329. [CrossRef]
12. Ditta, A.; Yongquan, C.; Azeem, M.; Rana, K.G.; Yu, H.; Memon, M.Q. Information hiding: Arabic text steganography by using Unicode characters to hide secret data. *Int. J. Electr. Secur. Digit. Forens.* **2018**, *10*, 61–78. [CrossRef]
13. Müller, J. Online Activities of Internet Users in Malaysia as of May 2020, by Activity. Available online: https://www.statista.com/statistics/788504/online-activities-of-internet-users-by-activity-malaysia/ (accessed on 24 April 2021).
14. Din, R.; Thabit, R.A.; Udzir, N.I.; Utama, S. Traid-bit embedding process on Arabic text steganography method. *Bull. Electr. Eng. Inform.* **2021**, *10*, 493–500. [CrossRef]
15. Gupta, S.; Gupta, D. Text-Steganography: Review Study & Comparative Analysis. *Int. J. Comput. Sci. Inf. Technol.* **2011**, *2*, 2060–2062.

16. Sharma, S.; Gupta, A.; Trivedi, M.C.; Yadav, V.K. Analysis of Different Text Steganography Techniques: A Survey. In Proceedings of the 2016 Second International Conference on Computational Intelligence & Communication Technology (CICT), Ghaziabad, India, 12–13 February 2016; pp. 130–133.
17. Osman, B.; Yasin, A.; Omar, M.N. An analysis of alphabet-based techniques in text steganography. *J. Telecommun. Electron. Comput. Eng.* **2016**, *8*, 109–115.
18. Hakak, S.; Kamsin, A.; Tayan, O.; Idris, M.Y.I.; Gilkar, G.A. Approaches for preserving content integrity of sensitive online Arabic content: A survey and research challenges. *Inf. Process. Manag.* **2017**, *56*, 367–380. [CrossRef]
19. Krishnan, R.B.; Thandra, P.K.; Baba, M.S. An Overview of Text Steganography. In Proceedings of the 2017 Fourth International Conference on Signal Processing, Communication and Networking (ICSCN), Chennai, India, 16–18 March 2017; pp. 1–6.
20. Ahvanooey, M.T.; Li, Q.; Shim, H.J.; Huang, Y. A Comparative Analysis of Information Hiding Techniques for Copyright Protection of Text Documents. *Secur. Commun. Netw.* **2018**, *2018*, 1–22. [CrossRef]
21. Baawi, S.S.; Mokhtar, M.R.; Sulaiman, R. A comparative study on the advancement of text steganography techniques in digital media. *ARPN J. Eng. Appl. Sci.* **2018**, *13*, 1854–1863.
22. Narayana, V.L.; Gopi, A.P.; Kumar, N.A. Different techniques for hiding the text information using text steganography techniques: A survey. *Ing. Sys. Inf.* **2018**, *23*, 115–125. [CrossRef]
23. Dhawan, S.; Gupta, R. Analysis of various data security techniques of steganography: A survey. *Inf. Secur. J. Glob. Perspect.* **2020**, *30*, 1–25. [CrossRef]
24. Muhammad, M.H.; Hussain, H.S.; Din, R.; Samad, H.; Utama, S. Review on feature-based method performance in text steganography. *Bull. Electr. Eng. Inform.* **2021**, *10*, 427–433. [CrossRef]
25. Mollin, R.A. *An Introduction to Cryptography*; CRC Press: Boca Raton, FL, USA, 2000.
26. Kawaguchi, E.; Eason, R.O. *Principles and Applications of BPCS Steganography Multimedia Systems and Applications*; International Society for Optics and Photonics: Boston, MA, USA, 1999; Volume 3528, pp. 464–474.
27. Xiang, L.; Wang, X.; Yang, C.; Liu, P. A Novel Linguistic Steganography Based on Synonym Run-Length Encoding. *IEICE Trans. Inf. Syst.* **2017**, *E100D*, 313–322. [CrossRef]
28. Girdhar, A.; Kumar, V. Comprehensive survey of 3D image steganography techniques. *IET Image Process.* **2017**, *12*, 1–10. [CrossRef]
29. Zakaria, A.A.; Hussain, M.; Wahab, A.W.A.; Idris, M.Y.I.; Abdullah, N.A.; Jung, K.-H. High-Capacity Image Steganography with Minimum Modified Bits Based on Data Mapping and LSB Substitution. *Appl. Sci.* **2018**, *8*, 2199. [CrossRef]
30. Singh, T.; Verma, S.; Parashar, V. Securing Internet of Things in 5G Using Audio Steganography. In Proceedings of the International Conference on Smart Trends for Information Technology and Computer Communications, Jaipur, India, 6–7 August 2016; pp. 365–372.
31. Banik, B.G.; Bandyopadhyay, B.; Kumar, S. Novel Text Steganography Using Natural Language Processing and Part-of-Speech Tagging. *IETE J. Res.* **2020**, *66*, 384–395. [CrossRef]
32. Arya, A.; Soni, S. A literature review on various recent steganography techniques. *Int. J. Future Revolut. Comput. Sci. Commun. Eng.* **2018**, *4*, 143–149.
33. Vijayakumar, P.; Vijayalakshmi, V.; Rajashree, R. Increased level of security using DNA steganography. *Int. J. Adv. Intell. Paradig.* **2018**, *10*, 74–82. [CrossRef]
34. Malathi, P.; Manoaj, M.; Manoj, R.; Raghavan, V.; Vinodhini, R.E. Highly improved DNA based steganography. *Procedia Comput. Sci.* **2017**, *115*, 651–659.
35. Yang, Z.-L.; Guo, X.-Q.; Chen, Z.-M.; Huang, Y.-F.; Zhang, Y.-J. RNN-Stega: Linguistic Steganography Based on Recurrent Neural Networks. *IEEE Trans. Inf. Forens. Secur.* **2019**, *14*, 1280–1295. [CrossRef]
36. Hamdan, A.M.; Hamarsheh, A. AH4S: An algorithm of text in text steganography using the structure of omega network. *Secur. Commun. Netw.* **2016**, *9*, 6004–6016. [CrossRef]
37. Dulera, S.; Jinwala, D.; Dasgupta, A. Experimenting with the Novel Approaches in Text Steganography. *Int. J. Netw. Secur. Appl.* **2012**, *3*, 213–225. [CrossRef]
38. Gaur, M.; Sharma, M. 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.* **2015**, *3*, 1344–1352. [CrossRef]
39. Wilson, A.; Blunsom, P.; Ker, A.D. Linguistic steganography on Twitter: Hierarchical language modeling with manual interaction. *Media Watermarking Secur. Forens.* **2014**, 9028. [CrossRef]
40. Odeh, A.; Elleithy, K.; Faezipour, M. Steganography in Text by Using MS Word Symbols. In Proceedings of the Conference of the American Society for Engineering Education, Bridgeport, CT, USA, 3–5 April 2014; pp. 1–5.
41. Rafat, K.F. Enhanced Text Steganography in SMS. In Proceedings of the 2009 2nd International Conference on Computer, Control and Communication, Karachi, Pakistan, 17–18 February 2009; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2009; pp. 1–6.
42. Bhat, D.; Krithi, V.; Manjunath, K.N.; Prabhu, S.; Renuka, A. Information Hiding through Dynamic Text Steganography and Cryptography: Computing and Informatics. In Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Manipal, India, 13–16 September 2017; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2017; pp. 1826–1831.

43. Naqvi, N.; Abbasi, A.T.; Hussain, R.; Khan, M.A.; Ahmad, B. Multilayer Partially Homomorphic Encryption Text Steganography (MLPHE-TS): A Zero Steganography Approach. *Wirel. Pers. Commun.* **2018**, *103*, 1563–1585. [CrossRef]
44. Low, S.H.; Maxemchuk, N.F.; Brassil, J.T.; O'Gorman, L. Document Marking and Identification Using both Line and Word Shifting. In Proceedings of the INFOCOM'95, Boston, MA, USA, 2–6 April 1995; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2002; pp. 853–860.
45. Huang, D.; Yan, H. Interword distance changes represented by sine waves for watermarking text images. *IEEE Trans. Circuits Syst. Video Technol.* **2001**, *11*, 1237–1245. [CrossRef]
46. Yang, H.; Kot, A.C. Text Document Authentication by Integrating Inter Character and Word Spaces Watermarking. In Proceedings of the 2004 IEEE International Conference on Multimedia and Expo (ICME), Taipei, Taiwan, 27–30 June 2004; IEEE: Piscataway, NJ, USA, 2004; pp. 955–958. [CrossRef]
47. Roy, S.; Manasmita, M. A novel approach to format-based text steganography. In Proceedings of the 2011 International Conference on Communication, Computing & Security ICCCS'11, Rourkela, India, 12–14 February; Association for Computing Machinery (ACM): New York, USA, 2011; pp. 511–516.
48. Kumar, P.; Sharma, V.K. Information Security Based on Steganography & Cryptography Techniques: A Review. *Int. J. Adv. Res. Comput. Sci. Softw. Eng.* **2014**, *4*, 246–250.
49. Mahato, S.; Yadav, D.K.; Khan, D.A. A Modified Approach to Text Steganography Using HyperText Markup Language. In Proceedings of the 2013 Third International Conference on Advanced Computing and Communication Technologies (ACCT), Rohtak, India, 6–7 April 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 40–44.
50. Bhattacharyya, S.; Indu, P.; Dutta, S.; Biswas, A.; Sanyal, G. Hiding Data in Text Through Changing in Alphabet Letter Patterns (CALP). *J. Glob. Res. Comput. Sci.* **2011**, *2*, 33–39.
51. Sui, X.-G.; Luo, H. A new steganography method based on hypertext. In Proceedings of the 2004 Asia-Pacific Radio Science Conference, Qingdao, China, 24–27 August 2004; pp. 181–184. [CrossRef]
52. Stutsman, R.; Grothoff, C.; Atallah, M.; Grothoff, K. Lost in just the translation. In Proceedings of the 2006 ACM Symposium on Applied Computing (SAC'06), Dijon, France, 23 April 2006; ACM: New York, NY, USA, 2006; pp. 338–345.
53. Banerjee, I.; Bhattacharyya, S.; Sanyal, G. Novel text steganography through special code generation. In Proceedings of the International Conference on Systemics, Cybernetics and Informatics (ICSCI-2011), Hyderabad, India, 5–8 January 2011.
54. Odeh, A.; Elleithy, K.M.; Faezipour, M. Text Steganography Using Language Remarks. In Proceedings of the 2013 ASEE Northeast Section Conference, Northfield, VT, USA, 14–16 March 2013.
55. Kataria, S.; Kumar, T.; Singh, K.; Nehra, M.S. ECR (encryption with cover text and reordering) based text steganography. In Proceedings of the 2013 IEEE Second International Conference on Image Information Processing (ICIIP-2013), Waknaghat, India, 9–11 December 2013; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2013; pp. 612–616.
56. Ramakrishnan, B.K.; Thandra, P.K.; Srinivasula, A.V.S.M. Text steganography: A novel character-level embedding algorithm using font attribute. *Secur. Commun. Netw.* **2016**, *9*, 6066–6079. [CrossRef]
57. Shiu, H.-J.; Lin, B.-S.; Huang, P.-Y.; Huang, C.-H.; Lei, C.-L. Data hiding on social media communications using text steganography. In Risks and Security of Internet and Systems; Springer: 2018; Volume 10694, pp. 217–224. In *Risks and Security of Internet and Systems*; Springer: Berlin/Heidelberg, Germany, 2018; Volume 10694, pp. 217–224.
58. Zhang, W.Z.; Zeng, Z.Z.; Pu, G.P.; Zhu, Z.H. Chinese Text Watermarking Based on Occlusive Components. In Proceedings of the 2006 2nd International Conference on Information & Communication Technologies, Damascus, Syria, 24–28 April 2006; pp. 1850–1854.
59. Changder, S.; Debnath, N.C.; Ghosh, D. A New Approach to Hindi Text Steganography by Shifting Matra. In Proceedings of the 2009 International Conference on Advances in Recent Technologies in Communication and Computing, Kottayam, India, 27–28 October 2009; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2009; pp. 199–202.
60. Alla, K.; Prasad, R.S.R. An Evolution of Hindi Text Steganography. In Proceedings of the 2009 Sixth International Conference on Information Technology: New Generations, Las Vegas, NV, USA, 27–29 April 2009; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2009; pp. 1577–1578.
61. Changder, S.; Ghosh, D.; Debnath, N.C. LCS based text steganography through Indian Languages. In Proceedings of the 2010 3rd International Conference on Computer Science and Information Technology, ICCSIT 2010, Chengdu, China, 9–11 July 2010; pp. 53–57.
62. Pathak, M. A new approach for text steganography using Hindi numerical code. *Int. J. Comput. Appl.* **2010**, *1*, 199–202.
63. Changder, S.; Debnath, N.C.; Ghosh, D. A Greedy Approach to Text Steganography Using Properties of Sentences. In Proceedings of the 2011 Eighth International Conference on Information Technology: New Generations, Las Vegas, NV, USA, 11–13 April 2011; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2011; pp. 30–35.
64. Alam, M.N.; Naser, M.A. Re-evaluating chain-code as features for Bangla script. In Proceedings of the 2013 International Conference on Electrical Information and Communication Technology (EICT), Khulna, Bangladesh, 13–15 February 2014; pp. 1–5.
65. Khan, S.; Abhijitha, B.; Sankineni, R.; Sunil, B.; Sungkrityayan, K. Polish text steganography method using letter points and extension. In Proceedings of the 2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), Tamil Nadu, India, 5–7 March 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 1–5.

66. Samphaiboon, N.; Dailey, M.N. Steganography in Thai text. In Proceedings of the 2008 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, Krabi, Thailand, 14–17 May 2008; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2008; Volume 1, pp. 133–136.
67. Khan, S.; Sankineni, R.; Balagurunathan, P.; Shree, N.S.D.; Balasubramanian, A. Czech Text Steganography Method by Selective Hiding Technique. In Proceedings of the World Congress on Engineering, London, UK, 1–3 July 2015.
68. Elnagar, A.; Al-Debsi, R.; Einea, O. Arabic text classification using deep learning models. *Inf. Process. Manag.* **2020**, *57*, 102121. [CrossRef]
69. Soufan, A. Deep Learning for Sentiment Analysis of Arabic Text. In Proceedings of the ArabWIC 6th Annual International Conference Research Track on ArabWIC 2019, Rabat, Morocco, 7 March 2019; ACM: New York, NY, USA, 2019; pp. 1–8.
70. Guellil, I.; Saâdane, H.; Azouaou, F.; Gueni, B.; Nouvel, D. Arabic natural language processing: An overview. *J. King Saud. Univ. Comput. Inf. Sci.* **2019**, *33*. [CrossRef]
71. Eldos, T. Arabic Text Data Mining: A Root-Based Hierarchical Indexing Model. *Int. J. Model. Simul.* **2003**, *23*, 158–166. [CrossRef]
72. Al-Salemi, B.; Ayob, M.; Kendall, G.; Noah, S.A.M. Multi-label Arabic text categorization: A benchmark and baseline comparison of multi-label learning algorithms. *Inf. Process. Manag.* **2019**, *56*, 212–227. [CrossRef]
73. Odeh, A.; Elleithy, K.; Faezipour, M. Steganography in Arabic text using Kashida variation algorithm (KVA). In Proceedings of the 2013 IEEE Long Island Systems, Applications and Technology Conference (LISAT), Farmingdale, NY, USA, 3 May 2013; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2013; pp. 1–6.
74. Shaker, A.A.; Ridzuan, F.; Pitchay, S.A. Text Steganography using Extensions Kashida based on the Moon and Sun Letters Concept. *Int. J. Adv. Comput. Sci. Appl.* **2017**, *8*, 286–290.
75. Shirali-Shahreza, M.H.; Shirali-Shahreza, M. A New Approach to Persian/Arabic Text Steganography. In Proceedings of the 5th IEEE/ACIS International Conference on Computer and Information Science and 1st IEEE/ACIS International Workshop on Component-Based Software Engineering, Software Architecture and Reuse (ICIS-COMSAR'06), Honolulu, HI, USA, 10–12 July; pp. 310–315.
76. Odeh, A.; Alzubi, A.; Hani, Q.B.; Elleithy, K. Steganography by multipoint Arabic letters. In Proceedings of the 2012 IEEE Long Island Systems, Applications and Technology Conference (LISAT), Farmingdale, NY, USA, 4 May 2012; pp. 1–7.
77. Al-Haidari, F.; Gutub, A.; Al-Kahsah, K.; Hamodi, J.; Hamodi, J.M. Improving security and capacity for Arabic text steganography using Kashida extensions. In Proceedings of the 2009 IEEE/ACS International Conference on Computer Systems and Applications, Rabat, Morocco, 10–13 May 2009; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2009; pp. 396–399.
78. Kadhem, S.M.; Wameedh, D. Proposed Arabic Text Steganography Method Based on New Coding Technique. *Int. J. Eng. Res. Appl.* **2016**, *6*, 38–46.
79. Aabed, M.A.; Awaideh, S.M.; Elshafei, A.-R.M.; Gutub, A.A. Arabic Diacritics based Steganography. In Proceedings of the 2007 IEEE International Conference on Signal Processing and Communications, Dubai, United Arab Emirates, 24–27 November 2007; pp. 756–759.
80. Gutub, A.; Elarian, Y.; Awaideh, S.; Alvi, A. Arabic text steganography using multiple diacritics. In Proceedings of the WoSPA 2008 International Workshop on Signal Processing and its Applications, Sharjah, United Arab Emirates, 18–20 March 2008.
81. Gutub, A.; Ghouti, L.; Elarian, Y.; Awaideh, S.; Alvi, A. Utilizing diacritic marks for Arabic text steganography. *Kuwait J. Sci. Eng.* **2010**, *37*, 89–109.
82. Shakir, A.C.; Xuemai, G.; Min, J. Chinese Language Steganography using the Arabic Diacritics as a Covered Media. *Int. J. Comput. Appl.* **2010**, *11*, 43–46. [CrossRef]
83. Bensaad, M.L.; Yagoubi, M.B. High capacity diacritics-based method for information hiding in Arabic text. In Proceedings of the 2011 International Conference on Innovations in Information Technology, Abu Dhabi, United Arab Emirates, 25–27 April 2011; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2011; pp. 433–436.
84. Memon, M.S.; Shah, A. A Novel Text Steganography Technique to Arabic Language Using Reverse Fatha (الفتحة). *Pak. J. Eng. Technol. Sci.* **2011**, *1*, 106–113.
85. Odeh, A.; Elleithy, K.M. Steganography in Arabic Text Using Full Diacritics Text. In Proceedings of the International Society for Computers and Their Applications, Inc., Bridgeport, CT, USA, 14 November 2012.
86. Bensaad, M.L.; Yagoubi, M.B. Boosting the Capacity of Diacritics-Based Methods for Information Hiding in Arabic Text. *Arab. J. Sci. Eng.* **2013**, *38*, 2035–2041. [CrossRef]
87. Ahmadoh, E.M.; Gutub, A.A.-A. Utilization of Two Diacritics for Arabic Text Steganography to Enhance Performance. *Lect. Notes Inf. Theory* **2015**, *3*. [CrossRef]
88. Malalla, S.; Shareef, F.R. A New Modified Fatha Method for Arabic Text Steganography Hybrid with Aes Encryption. *IOSR J. Comput. Eng.* **2016**, *18*, 37–45. [CrossRef]
89. Azmi, A.; Alsaiari, A. Arabic typography: A survey. *Int. J. Electr. Comput. Sci.* **2010**, *9*, 16–22.
90. Gutub, A.; Fattani, M. A Novel Arabic Text Steganography Method Using Letter Points and Extensions. In Proceedings of the International Conference on Computer, Information and Systems Science and Engineering (ICCISSE), Vienna, Austria, 25–27 May 2007.
91. Al-Alwani, W.; Mahfooz, A.B.; Gutub, A.A. *A Novel Arabic Text Steganography Method Using Extensions*; World Academy of Science, Engineering and Technology: Barcelona, Spain, 2007; pp. 502–505.

92. Al-Nazer, A.; Gutub, A. Exploit Kashida Adding to Arabic e-Text for High Capacity Steganography. In Proceedings of the 2009 Third International Conference on Network and System Security, Golf Coast, QLD, Australia, 19–21 October 2009; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2009; pp. 447–451.
93. Gutub, A.A.; Al-Nazer, A.A. High Capacity Steganography Tool for Arabic Text Using 'Kashida'. *ISeCure* **2010**, *2*, 107–118.
94. Alhusban, A.M.; Alnihoud, O. A meliorated kashida-based approach for Arabic text steganography. *Int. J. Comput. Sci. Inf. Technol.* **2017**, *9*, 99–112. [CrossRef]
95. Malalla, S.; Shareef, F.R. Improving Hiding Security of Arabic Text Steganography by Hybrid AES Cryptography and Text Steganography. *J. Eng. Res. Appl.* **2016**, *6*, 60–69.
96. Al-Oun, S.M.; Alnihoud, J.Q.O. An Efficient Approach to Hide Compressed Voice Data in Arabic Text using Kashida and "La". *J. Comput. Sci.* **2017**, *13*, 48–54. [CrossRef]
97. Shirali-Shahreza, M. Pseudo-space Persian/Arabic text steganography. In Proceedings of the 2008 IEEE Symposium on Computers and Communications, Marrakech, Morocco, 6–9 July 2008; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2008; pp. 864–868.
98. Shirali-Shahreza, M.; Shirali-Shahreza, M.H. An Improved Version of Persian/Arabic Text Steganography Using "La" Word. In Proceedings of the 2008 6th National Conference on Telecommunication Technologies and 2008 2nd Malaysia Conference on Photonics, Kuala Lampur, Malaysia, 26–28 August 2008; pp. 372–376. [CrossRef]
99. Shirali-Shahreza, M.; Shirali-Shahreza, S. Persian/Arabic Unicode Text Steganography. In Proceedings of the 2008 The Fourth International Conference on Information Assurance and Security, Napoli, Italy, 8–10 September 2008; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2008; pp. 62–66.
100. Shirali-Shahreza, M.H.; Shirali-Shahreza, M. Arabic/Persian text steganography utilizing similar letters with different codes. *Arab. J. Sci. Eng.* **2010**, *35*, 213–222.
101. Mohamed, A.A. An improved algorithm for information hiding based on features of Arabic text: A Unicode approach. *Egypt. Inform. J.* **2014**, *15*, 79–87. [CrossRef]
102. Abbasi, A.T.; Naqvi, S.N.S.; Khan, A.; Ahmad, B. Urdu Text Steganography: Utilizing Isolated Letters. In Proceedings of the 13th Australian Information Security Management Conference, Perth, Western Australia, 30 November–2 December 2015; pp. 37–46.
103. Obeidat, A.A. Arabic Text Steganography Using Unicode of Non-Joined to Right Side Letters. *J. Comput. Sci.* **2017**, *13*, 184–191. [CrossRef]
104. Roslan, N.A.; Mahmod, R.; Udzir, N.I. Sharp-edges Method in Arabic Text Steganography. *J. Theor. Appl. Inf. Technol.* **2011**, *33*, 32–41.
105. Mersal, S.; Alhazmi, S.; Alamoudi, R.; Almuzaini, N. Arabic Text Steganography in Smartphone. *Int. J. Comput. Inf. Technol.* **2014**, *3*, 2279–2764.
106. Roslan, N.A.; Mahmod, R.; Udzir, N.I.; Zurkarnain, Z.A. Primitive Structural Method for High Capacity Text Steganography. *J. Theor. Appl. Inf. Technol.* **2014**, *67*, 373–383.
107. Khan, E. Using Arabic Poetry System for Steganography. *Asian J. Comput. Sci. Inf. Technol.* **2014**, *4*, 55–61. [CrossRef]
108. Al-Nofaie, S.M.; Fattani, M.M.; Gutub, A.A. Merging Two Steganography Techniques Adjusted to Improve Arabic Text Data Security. *J. Comput. Sci. Comput. Math.* **2016**, *6*, 59–65. [CrossRef]
109. Kadhem, S.M.; Ali, D.W.M. Proposed Hiding Text in Text Based on RNA for Encoding Secret Information. *Iraqi J. Sci.* **2017**, *58*, 562–573.
110. Malalla, S.; Shareef, F.R. A Novel Approach for Arabic Text Steganography Based on the "BloodGroup" Text Hiding Method. *Eng. Technol. Appl. Sci. Res.* **2017**, *7*, 1482–1485. [CrossRef]
111. Alshahrani, H.M.S.; Weir, G. Hybrid Arabic text steganography. *Int. J. Comput. Inf. Technol.* **2017**, *6*, 329–338.
112. Taha, A.; Hammad, A.S.; Selim, M.M. A high capacity algorithm for information hiding in Arabic text. *J. King Saud. Univ. Comput. Inf. Sci.* **2020**, *32*, 658–665. [CrossRef]
113. Gutub, A.A.; Alaseri, K. Hiding Shares of Counting-Based Secret Sharing via Arabic Text Steganography for Personal Usage. *Arab. J. Sci. Eng.* **2019**, *45*, 1–26. [CrossRef]
114. Alanazi, N.; Khan, E.; Gutub, A. Inclusion of Unicode Standard seamless characters to expand Arabic text steganography for secure individual uses. *J. King Saud. Univ. Comput. Inf. Sci.* **2020**, (in press). [CrossRef]
115. Aman, M.; Khan, A.; Ahmad, B.; Kouser, S. A Hybrid Text Steganography Approach Utilizing Unicode Space Characters and Zero-Width Character. *Int. J. Inf. Technol. Secur.* **2017**, *9*, 85–100.
116. Ahvanooey, M.T.; Li, Q.; Hou, J.; Mazraeh, H.D.; Zhang, J. AITSteg: An Innovative Text Steganography Technique for Hidden Transmission of Text Message via Social Media. *IEEE Access* **2018**, *6*, 65981–65995. [CrossRef]
117. Malik, A.; Sikka, G.; Verma, H.K. A high capacity text steganography scheme based on Huffman compression and color coding. *J. Inf. Optim. Sci.* **2017**, *38*, 647–664. [CrossRef]
118. Al-Azzawi, A.F. A Multi-Layer Hybrid Text Steganography for Secret Communication Using Word Tagging and RGB Color Coding. *Int. J. Netw. Secur. Its Appl.* **2018**, *10*, 01–12. [CrossRef]
119. Altaay, A.A.J.; Sahib, S.B.; Zamani, M. An Introduction to Image Steganography Techniques. In Proceedings of the 2012 International Conference on Advanced Computer Science Applications and Technologies (ACSAT), Kuala Lumpur, Malaysia, 26–28 November 2012; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2012; pp. 122–126.

120. Cox, I.; Miller, M.L.; Bloom, J.; Fridrich, J.; Kalker, T. *Digital Watermarking and Steganography*, 2nd ed.; Morgan Kaufmann Publishers: Burlington, MA, USA, 2007.
121. Korzhik, V.; Fedyanin, I.; Cuong, N.D. Detection of stegosystems using block ciphers for encryption of the embedded messages. In Proceedings of the 2017 20th Conference of Open Innovations Association (FRUCT), Saint Petersburg, Russia, 3–7 April 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 181–186.
122. Zamani, M.; Manaf, A.B.A.; Ahmad, R.B.; Jaryani, F.; Taherdoost, H.; Zeki, A.M. A Secure Audio Steganography Approach. In Proceedings of the 2009 International Conference for Internet Technology and Secured Transactions (ICITST), London, UK, 9–12 November 2009; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2009; pp. 1–6.
123. Roslan, N.A.; Udzir, N.I.; Mahmod, R.; Zukarnain, Z.A.; Ninggal, M.I.H.; Thabit, R. Character Property Method for Arabic Text Steganography with Biometric Multifactor Authentication Using Liveness Detection. *J. Theor. Appl. Inf. Technol.* **2020**, *98*, 4140–4157.
124. Morkel, T.; Eloff, J.H.; Olivier, M.S. An overview of image steganography. In Proceedings of the Fifth Annual Information Security South Africa Conference, Sandton, South Africa, 29 June–1 July 2005.
125. Kouser, S.; Khan, A. A Novel Feature Extraction Approach: Capacity Based Zero-Text Steganography. *Int. J. Inf. Technol. Secur.* **2017**, *3*, 85–99.
126. Zaheer, M. *Secure Communication Using Steganography in Image Processing*; Air University: Islamabad, Pakistan, 2018.
127. Ekodeck, S.G.R.; Ndoundam, R. PDF steganography based on Chinese Remainder Theorem. *J. Inf. Secur. Appl.* **2016**, *29*, 1–15. [CrossRef]
128. Afanasyeva, O. Analysis of Aspects of Messages Hiding in Text Environments. *J. Konbin* **2015**, *34*, 5–16. [CrossRef]
129. Chao, M.-W.; Lin, C.-H.; Yu, C.-W.; Lee, T.-Y. A High Capacity 3D Steganography Algorithm. *IEEE Trans. Vis. Comput. Graph.* **2008**, *15*, 274–284. [CrossRef] [PubMed]
130. Kour, J.; Verma, D. Steganography techniques–A review paper. *Int. J. Emerg. Res. Manag. Technol.* **2014**, *3*, 132–135.
131. Westfeld, A.; Pfitzmann, A. Attacks on Steganographic Systems. In Proceedings of the 6th European Conference on Computer Vision, ECCV 2000, Dublin, Ireland, 26 June–1 July 2000; pp. 61–76.
132. Khairullah, M. A novel steganography method using transliteration of Bengali text. *J. King Saud. Univ. Comput. Inf. Sci.* **2019**, *31*, 348–366. [CrossRef]
133. Reinel, T.-S.; Raul, R.-P.; Gustavo, I. Deep Learning Applied to Steganalysis of Digital Images: A Systematic Review. *IEEE Access* **2019**, *7*, 68970–68990. [CrossRef]
134. Jaro, M.A. Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. *J. Am. Stat. Assoc.* **1989**, *84*, 414–420. [CrossRef]
135. Winkler, W.E. *String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage Report*; Educational Resources Information Center (ERIC): Washington, DC, USA, 1990; pp. 354–359.
136. Devi, K.R.; Prabakaran, S. An Enhanced Bilateral Information Security towards a Conventional Cryptographic System using DNA Sequences. *Indian J. Sci. Technol.* **2016**, *9*. [CrossRef]
137. Cheddad, A.; Condell, J.; Curran, K.; Mc Kevitt, P. Digital image steganography: Survey and analysis of current methods. *Signal Process.* **2010**, *90*, 727–752. [CrossRef]
138. Xue, Y.; Zhou, J.; Zeng, H.; Zhong, P.; Wen, J. An adaptive steganographic scheme for H.264/AVC video with distortion optimization. *Signal Process. Image Commun.* **2019**, *76*, 22–30. [CrossRef]
139. Nazari, M.; Ahmadi, I.D. A novel chaotic steganography method with three approaches for color and grayscale images based on FIS and DCT with flexible capacity. *Multimed. Tools Appl.* **2020**, *79*, 13693–13724. [CrossRef]
140. Xiang, L.; Wu, W.; Li, X.; Yang, C. A linguistic steganography based on word indexing compression and candidate selection. *Multimed. Tools Appl.* **2018**, *77*, 28969–28989. [CrossRef]
141. Prabakaran, G.; Bhavani, R.; Rajeswari, P. Multi secure and robustness for medical image based steganography scheme. In Proceedings of the 2013 International Conference on Circuits, Power and Computing Technologies (ICCPCT), Nagercoil, India, 20–21 March 2013; pp. 1188–1193.
142. Osman, B.B. Message Hiding Technique in Text Steganography using RGB Colour Approach and Random Location. Ph.D. Thesis, Universiti Utara Malaysia, Changlun, Malaysia, 2020.
143. Xiang, L.; Sun, X.; Luo, G.; Xia, B. Linguistic steganalysis using the features derived from synonym frequency. *Multimed. Tools Appl.* **2014**, *71*, 1893–1911. [CrossRef]
144. Gutub, A.A.; Al-Ghamdi, M. Image Based Steganography to Facilitate Improving Counting-Based Secret Sharing. *3D Res.* **2019**, *10*. [CrossRef]
145. Hamdani, H.; Ismanto, H.; Munir, A.Q.; Rahmani, B.; Syafrianto, A.; Suprihanto, D.; Septiarini, A. The Proposed Development of Prototype with Secret Messages Model in Whatsapp Chat. *Int. J. Electr. Comput. Eng.* **2018**, *8*, 3841–3850. [CrossRef]
146. Arunkumar, S.; Subramaniyaswamy, V.; Vijayakumar, V.; Chilamkurti, N.; Logesh, R. SVD-based robust image steganographic scheme using RIWT and DCT for secure transmission of medical images. *Measurement* **2019**, *139*, 426–437. [CrossRef]
147. Saravanan, K.; Purusothaman, T.; Velmurugan, T.; Kavitha, K.V.N. Design and Performance Analysis of Diverse Generic Data Hiding Algorithms in Cryptography. *ARPN J. Eng. Appl. Sci.* **2017**, *12*, 6423–6429.

148. Pandya, I.; Jhajj, S.; Pawar, R. A steganographic approach to mitigate password attacks. In Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2017, Manipal, India, 13–16 September 2017; pp. 248–253.
149. Liu, J.; Lu, T.; Zhao, Q. Improving the performance of lossless reversible steganography via data sharing. In Proceedings of the IEEE 8th International Conference on Awareness Science and Technology (iCAST 2017), Taichung, Taiwan, 8–10 November 2017; pp. 7–12.
150. Luo, Z.; Xie, W.; Wang, B.; Tang, Y.; Xing, Q. EasyStego: Robust Steganography Based on Quick-Response Barcodes for Crossing Domains. *Symmetry* **2019**, *11*, 222. [CrossRef]
151. Wijayanto, E.F.; Zarlis, M.; Situmorang, Z. Increase the PSNR of image using LZW and AES algorithm with MLSB on steganography. *Int. J. Eng. Technol.* **2018**, *7*, 119–121. [CrossRef]