ISSN 2723-6471

Coverless Text Information Hiding Based on Built-in Features of Arabic Scripts

Sabaa Hamid Rashid[1,*], Dhamyaa A. Nasrawi[2]

[1,2]*Department of Computer Science, College of Computer Science and Information Technology, University of Kerbala, Karbala, 56001, Iraq*

(Received: February 12, 2024; Revised: March 18, 2024; Accepted: April 22, 2024; Available online: May 31, 2024)

Abstract

Text steganography is crucial in information security due to the limited redundancy in text. The Arabic language features offer a new method for data concealment. In this paper, the researchers propose a new coverless text information hiding method based on built-in features of Arabic scripts. The first word of each row in the dataset is tested based on eight features to get one byte containing 1 or 0. That is a result of the presence or absence of the following features: mahmoze, diacritics, isolated, two sharp edges, vowels, dotted, looping, and high frequency. Then, each byte is converted to a decimal number (ASCII code) to implement a dynamic mapping protocol with the most frequent letter. In the hiding process, each character in the secret message is converted to ASCII code and successfully matched in the dataset. Thus, after matching, the candidate text is sent to the receiver. In contrast, the pre-agreed dynamic mapping protocol was implemented in a receiver to extract secret messages. Three Arabic datasets are used in this paper SANAD (Single-Label Arabic News Articles Dataset) includes 45500 articles, Arabic Poem Comprehensive Dataset (APCD) contains 1,831,770 poetic verses in total, Arabic Poetry Dataset contains more than 58000 poems). The suggested approach withstands existing detecting methods because of no modification or generation. Moreover, there is an enhancement in hiding capacity, which can conceal a (character per word). Hence, all the messages are embedded successfully using dynamic mapping.

Keywords: Arabic script, Built-in Features, Coverless Information Hiding, Dynamic Mapping

1. Introduction

This Information hiding refers to a technology used to manipulate data that needs protection, allowing the secure transmission of confidential information through public channels. This concept involves combining various fields and technological aspects. The main purpose of information hiding is to hide confidential information from the unauthorized user and third parties [1].

Various kinds of multimedia can be used to hide messages, including text, images, audio, and video. Compared to others, text steganography is entirely difficult to design as a result of the absence of redundancy. Nevertheless, it offers simpler communication and consumes less memory. Working with text has benefits despite this challenge, including easy encoding, a large amount of data, effective use of space, and widespread use [2].

Text steganography is a branch of steganography, sometimes also referred to as linguistic steganography. It relies on hiding information in textual messages and textual documents as cover data, including those in magazines, newspapers, word processing documents, personal notes, etc. [3]. Text steganography is mainly divided into two categories: format-based methods and content-based ones [4]. Format-based methods use special characters, line-shifting, and word-shifting to code the text [5]. It is based on a space character [6], [7], altering the space between words or lines [8], or deals with font attributes [9], making the same glyphs for the multiple codes by designing new fonts with special properties [10]. These format-based techniques can fool humans, but they are unable to fool computer programs or make the stego text longer. Moreover, these techniques exhibit reduced resilience to text retyping attacks which makes it simple to lose the hidden data [11], [12].

***Corresponding author: Sabaa Hamid Rashid (saba.h@s.uokerbala.edu.iq)**
DOI: https://doi.org/10.47738/jads.v5i2.243

Meanwhile, content-based methods are usually based on lexical, syntactic, or semantic manipulation, such as using a synonym replacement strategy [10], changing the structures of sentences [5], ligature characters Unicode [13], Unicode of characters in multilingual [14], etc., or composing a new text that mimics some aspects of the typical text such as generating a random sequence [15] or considering context-free grammars [3]. These content-based techniques must guarantee that the modified text meets the necessary standards for grammatical rationality and semantic accuracy, as well as that the resulting text's quality, which contains concealed information, is sufficient [11].

However, the conventional methods of information hiding above are difficult to resist the corresponding steganalysis methods because the modification always causes some distinguishable features. Additionally, altering a carrier will unavoidably result in some distortion. Coverless text steganography, which belongs to the genre of steganography without a cover alteration, is therefore receiving greater attention. Text is a popular form of mobile communication due to its small size, high transmission speed, and ease of reading. Current instant messaging applications such as Facebook Messenger, WhatsApp, or Telegram prioritize mobile computing security while enabling real-time communication. Hence the importance of hiding in maintaining the security of communication [16].

A newly introduced concept known as “coverless information hiding” has been devised by researchers to tackle issues associated with conventional information hiding techniques. It is crucial to differentiate between the term “coverless” and the absence of a carrier signal. In this context, “coverless information hiding” denotes that the hidden information does not require embedding within a predetermined carrier [17]. Typically, coverless information hiding techniques exhibit three primary characteristics: they involve no modification, no embedding, and demonstrate resistance to steganalysis attacks [18], [19].

Xiang et al. [20] categorize coverless text information hiding methods into two general groups. The first is the search method (or called carrier retrieval) [21], which involves seeking available carriers that align with the statistical characteristics of the secret message, or the message from the cover text must be retrieved while adhering to certain guidelines. In order to conceal information without changing the carrier text, Chen et al. [22] presented the notion of "tag + keyword," in which the tag is determined and connected to a Chinese mathematical expression. The second group is the generation method [20], [23], which involves the automatic generation of paraphrases, serving as a novel and valuable resource for linguistic steganography transformations. Fang et al. [16] employed a new linguistic stego-system based on a Long Short-Term Memory (LSTM) neural network to train language generating models and obtain effective concealment effects,

In order to address the issue of the inadequate security and low concealment of traditional data concealing methods, this research suggests a new coverless text information hiding method within the search method group. The suggested method is predicated on Arabic scripts' inherent characteristics. Dynamic mapping protocol was implemented with the most frequent letter to get successful matching in the selected dataset. The candidate text, after matching, was sent to a receiver. The pre-agreed dynamic mapping protocol was implemented to extract secret messages in the extracting process. The current paper has the following key contributions:

1) By leveraging the characteristics of the Arabic language, it is utilized for information concealment. In this study, the researchers employed a set of Arabic language features to hide data.
2) The researchers used a dynamic mapping protocol for information hiding based on the frequency of English letters and ASCII code distribution of Arabic language features in the selected dataset.
3) They enhanced the embedding rate and hiding capacity by hiding character/word.

The rest of this paper is organized as follows. Section 2 introduces the features of the Arabic language, section 3 presents the related works, section 4 incorporates the proposed method, section 5 explicates the experiment results, and Section 6 summarizes the conclusions of the study.

2. Arabic Language Features

Arabic, spoken by roughly 200 million people [24], is the world’s fifth most spoken language [25],[26]. Arabic Internet material expands amid daily Internet activities [27]. Arabic has 28 characters and is written in a cursive form akin to

ISSN 2723-6471

Urdu and Farsi. An Arabic letter changes shape depending on where it is in a word. It could be in the first, middle, or last place, or it could be alone. Table 1 explains Arabic alphabets and their forms [28].

Table 1. Arabic alphabets forms [28]

Name	Unicode	Shapes			
		Isolated	Final	Medial	Initial
HAMZA	0621	ء			
ALEF WITH MADDA ABOVE	0622	آ	ـآ		
ALEF WITH HAMZA ABOVE	0623	أ	ـأ		
WAW WITH HAMZA ABOVE	0624	ؤ	ـؤ		
ALEF WITH HAMZA BELOW	0625	إ			
YEH WITH HAMZA ABOVE	0626	ئ	ـئ	ـئـ	ئـ
ALEF	0627	ا	ـا		
BEH	0628	ب	ـب	ـبـ	بـ
TEH MERBUTA	0629	ة	ـة		
THE	062A	ت	ـت	ـتـ	تـ
THEH	062B	ث	ـث	ـثـ	ثـ
JEEM	062C	ج	ـج	ـجـ	جـ
HAH	062D	ح	ـح	ـحـ	حـ
KHAH	062E	خ	ـخ	ـخـ	خـ
DAL	062F	د	ـد		
THAL	0630	ذ	ـذ		
RAH	0631	ر	ـر		
ZAIN	0632	ز	ـز		
SEEN	0633	س	ـس	ـسـ	سـ
SHEEN	0634	ش	ـش	ـشـ	شـ
SAD	0635	ص	ـص	ـصـ	صـ
DHAD	0636	ض	ـض	ـضـ	ضـ
TAH	0637	ط	ـط	ـطـ	طـ
ZAH	0638	ظ	ـظ	ـظـ	ظـ
AIN	0639	ع	ـع	ـعـ	عـ
GHAIN	063A	غ	ـغ	ـغـ	غـ
FEH	0641	ف	ـف	ـفـ	فـ
QAF	0642	ق	ـق	ـقـ	قـ
KAF	0643	ك	ـك	ـكـ	كـ
LAM	0644	ل	ـل	ـلـ	لـ
MEEM	0645	م	ـم	ـمـ	مـ
NOON	0646	ن	ـن	ـنـ	نـ
HEH	0647	ه	ـه	ـهـ	هـ
WAW	0648	و	ـو		
ALEF MAKSOUR	0649	ى	ـى		
YEH	064A	ي	ـي	ـيـ	يـ
HAMZA	0621	ء			
ALEF WITH MADDA ABOVE	0622	آ	ـآ		
ALEF WITH HAMZA ABOVE	0623	أ	ـأ		

 ISSN 2723-6471

Each word usually consists of more than two connected letters. One, two, or three dots are put above or below the letter in some Arabic letters. Unlike English, which lacks multipoint letters, Arabic features 15 pointed letters, 5 of which are multipoint, see table 2.

Table 2. Dots in Arabic letters

No. of dots	Arabic letters
0	ا , ح , د , ر , س , ص , ط , ع , ك , ل , م , و ، ه، ء، ؤ، ئ
1	ب، ج، خ، ذ، ز، ض، ظ، غ، ف، ن
2	ت، ق، ي
3	ث، ش

Arabic symbols such as fathah, dhammah, kasrah, sukon, and shaddah use diacritical marks, which are short vowels. Tanween can alternatively be formed from two dhammas, kasrah, or fathah. These diacritical marks can be positioned above or below the characters and are written as strokes. A word's meaning can be altered by altering a character's diacritical mark. Arabic readers are used to inferring meaning from context when reading undiacritical texts, see table 3.

Table 3. Diacritical marks of Arabic language

Name	Diacritical Marks
Fathah	ـَ
Dhammah	ـُ
Kasrah	ـِ
Sukon	ـْ
Shaddah	ـّ
Two Fathah	ـً
Two Dhammas	ـٌ
Two Kasrah	ـٍ

The different character forms that are used to write an Arabic sentence are another characteristic of the Arabic script. Arabic characters are written from right to left, and some do not connect to the characters that come after them. An Arabic character's shape in a sentence depends on where it falls within the word. A character can produce one of four shapes when it is isolated, connected to its position in the word from the right (ending form), connected from the left (starting form), or connected from both sides (middle form) [29]. The extended character that appears between words to write (ـ) is another distinctive aspect of Arabic script, commonly referred to as Kashida, such as 'كتب' extended letter 'ك ' to 'كـــتب' [30].

Having a lot of sharp edges is another distinctive feature of the Arabic characters found by [31], see Table 4. The Arabic letters range from one sharp edge (that is, و) to five sharp edges (that is, ك). The sharp letters in Arabic are categorized into five groups based on the number of edges [32]. The proposed method used 2 sharp edges, as shown in table 4.

Table 4. The number of sharp edges in Arabic letters [32]

The number of sharp edges	Arabic letters
1	و ة ف ه م
2	ا ب ت ث د ذ ر ز ي ل ط ظ ن
3	غ ع ء ج ح خ
4	س ش
5	ك

Each Arabic letter in this standard has a general code, and for each of its several presentation forms within a word, it has a separate code. For instance, 0628 is the general code for the letter Beh, FE8F represents an isolated form, FE90

denotes the ultimate form, FE91 indicates the beginning form, and FE92 denotes the middle form. When a character is isolated, connected to its location in the word from the right (ending form), connected from the left (beginning form), or connected from both sides (middle form), it can yield one of four shapes. Arabic text steganography may employ a code from the Unicode Standard to represent each of these variants [33]. Many other characteristics can be explored in the proposed method.

3. Related Works

This section reviews some relevant works of text hiding that have been presented in different languages. Text hiding is generally regarded as one of the most difficult kinds of information hiding strategies since text files have far less surplus information than graphics or audio files. Most coverless text information hiding systems are implemented with the English and Chinese languages in mind, and the techniques used in these systems are not transferable to other languages. For instance, there are significant differences between Arabic, Chinese, English, and other European languages.

Xia et al. [2] introduced a novel method for coverless text information hiding, utilizing the Least Significant Bit (LSB) of the Character's Unicode. The approach involved the conversion of internet-collected texts into LSB texts, which were then stored in a database. Notably, this method necessitated a substantial text database and was suitable for low-capacity texts, typically around 14 or 15 bits, when the number of texts reached approximately 200,000.

In 2019, Wang K. and Gao Q. [23] introduced a coverless information hiding technique utilizing character features to represent binary digits. Their approach involved establishing a mapping function between character features and a Binary Digit String (BDS), utilizing the Parity of Chinese Characters' Stroke Number (PCCSN) as a character feature. Experimental results demonstrated a high embedding rate, reaching 6.25% in the best case and 25% in the improved method, along with enhanced security, robustness, and a high success rate. This method offered advantages such as resistance to format transformation attacks, improved semantic and statistical detection, and applicability across multiple languages. Remarkably, the technique did not require additional information and was effective with large-scale text corpora.

In 2021, Liu et al. [34] made advancements over the methods previously introduced by Chen et al. [22] and Wu et al. [35], focusing on enhancing success rates, extraction accuracy, and hiding capacity. They employed Part of Speech (POS) for concealing keyword numbers and optimizing stego-text retrieval. Chinese character components were utilized to locate marks. The "Word2Vec" language model was employed to expand the keyword set. This approach resulted in improved embedding capacity, eliminating ambiguity in locating markers. Furthermore, the method heightened extraction accuracy by increasing the number of keywords in stego-texts mapped to POS, thereby enhancing the embedding success rate and demonstrating resilience against current steganalysis techniques.

In 2021, Wang et al. [8] introduced a novel coverless text steganography technique utilizing the structures of Chinese character components. These structures were categorized into groups based on their usage frequency, representing specific Binary Digital Strings (BDS). The Minimal Square Matrix (MSM) underwent conversion into a Code Square Matrix (CSM). Through the Chinese remainder theorem, the BDSs were transformed into binary digital slices (BiDSs). During the reception, the BDS was reconstructed, and the corresponding row and column numbers were determined.

Unfortunately, there is no research on coverless information hiding in Arabic. However, oppositely, there has been much research on conventional information hiding in the Arabic language, such as concealing a secret text based on the dots of the letters [36], and concealing textual data within the visual allure of elongated letters or Kashida [26], [37], [38]. Harakats also serve a secondary purpose of concealing confidential information within the text [3], [39], [40]. Another approach allows for using different Unicode values for the same letter, enabling the concealment of information [41], and utilizing the sharp edges of Arabic letters to conceal secret bits, which is proposed by the authors in [31]. Each letter conceals secret bits according to the count of its sharp edges.

However, it is essential to note that the standard methods of information concealing above lack robust security against conventional intruders and cannot overcome steganalysis methods.

4. Method

This section provides a detailed explanation of the proposed coverless text information hiding based on built-in features of Arabic scripts. The three parts that make up the suggested method are the dynamic mapping protocol, the hiding procedure, and the extracting procedure.

4.1. Dynamic Mapping Protocol

In this step, implementing the dynamic mapping protocol is the pre-agreed protocol upon beforehand between the sender and receiver. Figure 1 shows the details of this step.

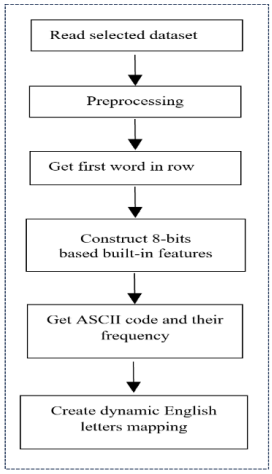


Figure 1. Dynamic mapping protocol

4.1.1. Read Selected Dataset

Three Arabic datasets are used in this paper:

1) SANAD (Single-Label Arabic News Articles Dataset), the articles were collected from three popular news websites: AlKhaleej, AlArabiya, and Akhbarona. SANAD includes (45500 articles in 7 categories) a wide range of topics, culture, finance, medical, politics, religion, sports, and tech [42].

2) Arabic Poem Comprehensive Dataset (APCD), the Arabic dataset is primarily scraped from الديوان and الموسوعة الشعرية. Once both are combined, there are 1,831,770 poetic verses in total [43].

3) Arabic Poetry Dataset (6th - 21st century) contains more than 58K poems in the dataset dating from the sixth to the current era. In addition, the poem metadata includes the poet's name, the poem's title, and its category. The source of the extracted dataset was adab.com [44], and two datasets about Arabic poetry were used from Kaggle. More details are explained in table 5.

Table 5. Summary of the selected datasets

Dataset	No. of rows	No. of words	Size /MB	Max char./row
SANAD	45,500	16,800,368	180.81	33,359
APCD	1,831,770	16,967,324	554.12	128
Arabic Poetry Dataset	58021	8,518,526	94.68	47,033

4.1.2. Preprocessing

Many processes in datasets were implemented in the Arabic news articles dataset, including 7 folders in multiple topics: culture, finance, medicine, politics, religion, sports, and tech. All text files in 7 folders were converted to CSV files. Meanwhile, the preprocessing in the other two datasets involved extracting just poetic verses from total poem metadata.

4.1.3. Get First Word in Row

The proposed method is implemented on the first word of each row in the selected dataset. In SANAD, 45,500 words were extracted, while in the APCD and Arabic poetry dataset, 1,831,770 and 58021 words were extracted, respectively.

4.1.4. Construct 8-Bit Based Built-In Features

In this step, the resulting words from the previous step were tested based on eight features to construct one byte containing 1 or 0 resulting from the presence or absence of these features. The features include: mahmoze, diacritics, isolated, two sharp edges, vowels, dotted, looping, and high frequency. Six features are implemented on the first character of the word (isolated, two sharp edges, vowels, dotted, looping, and high frequency), while the other features (mahmoze, diacritics) are implemented on the end character and whole word, respectively. The following functions explain built-in features in detail:

1) Is_ mahmoze: This function checks if a word ends with a mahmoze letter; mahmoze letters mean that they contain HAMZA [‘أ’, ‘ؤ’, ‘ئ’, ‘ء’].
2) Has_ diacritics: This function checks if a word has diacritics [‘ً’, ‘ٌ’, ‘ٍ’, ‘َ’, ‘ُ’, ‘ِ’, ‘ّ’, ‘ْ’, ‘ٰ’, ‘ٓ’, ‘ٔ’, ‘ٕ’, ‘-’].
3) Is_ isolated: It checks if a word starts with isolated letters, that is, cannot connected with the next one [‘ؤ’, ‘و’, ‘ز’, ‘ا’, ‘ر’, ‘ذ’, ‘د’].
4) Is_ 2_sharp edges: It checks if a word starts with 2_sharp edges letters ['ا', 'ب', 'ت', 'ث', 'د','ذ' 'ر', 'ز', 'ي', 'ل', 'ط', 'ظ', 'ن"].
5) Is_ vowels: It checks if a word starts with vowel letters [‘ؤ’, ‘ي’, ‘و’, ‘آ’, ‘إ’, ‘أ’, ‘ا’].
6) Is_ dotted: It checks if a word starts with dotted letters: ['ي', 'ن', 'ف', 'ق', 'غ', 'ض', 'ظ', 'ش', 'ز', 'ذ', 'خ', 'ج', 'ث', 'ت', 'ب'].
7) Is_ loping: It checks if a word starts with looping letters: ['ه', 'و', 'ق', 'ف', 'ظ', 'ط', 'ض', 'ص', 'ة', 'ؤ'].
8) Is_ high-frequency: It checks if a word starts with high-frequency letters, the high-frequency Arabic letter based on the Holy Quran [45] (see table 6). In the proposed method, seven high-frequency letters were selected [‘ا’, ‘ل’, ‘ه’, ‘و’, ‘ي’, ‘م’, ‘ن’].

Table 6. Arabic letter Arabic letter frequency using only the Quran as input source [45]

Rank	Letter	Frequency	Percentage	Rank	Letter	Frequency	Percentage
1	ا	43,542	13.17	19	ذ	4,932	1.49
2	ل	38,191	11.55	20	ح	4,140	1.25
3	ن	27,270	8.25	21	ج	3,317	1.00
4	م	26,735	8.08	22	ى	2,592	0.78
5	و	24,813	7.50	23	خ	2,497	0.76
6	ي	21,973	6.64	24	ة	2,344	0.71
7	ه	14,850	4.49	25	ش	2,124	0.64
8	ر	12,403	3.75	26	ص	2,072	0.63
9	ب	11,491	3.47	27	ض	1,686	0.51
10	ت	10,520	3.18	28	ز	1,599	0.48
11	ك	10,497	3.17	29	ء	1,578	0.48
12	ع	9,405	2.84	30	آ	1,511	0.46
13	أ	9,119	2.76	31	ث	1,414	0.43
14	ف	8,747	2.64	32	ط	1,273	0.38
15	ق	7,034	2.13	33	غ	1,221	0.37

16	س	6,012	1.82	34	ئ	1,182	0.36
17	د	5,991	1.81	35	ظ	853	0.26
18	إ	5,108	1.54	36	ؤ	673	0.20

Table 7 explains examples of Arabic words and their 8-bit based built-in features.

Table 7. Example of construct 8-bit based built-in features

	Built-in features							
Words	mah moze	diacritics	isolated	2 sharp edges	vowels	dotted	looping	high frequency
استضافت	0	0	1	1	1	0	0	1
اليَعْسُوب	0	1	1	1	1	0	0	1
زار	0	0	1	1	0	1	0	0

4.1.5. Get Ascii Code and Their Frequency

In this step, the resulting 8 bits are converted to decimal numbers (ASCII code) and find their frequency in the Arabic news dataset. The distribution of ASCII code in the selected dataset is very significant in hiding capacity. Table 8 shows the ASCII code and their frequency in the dataset, while figure 2 shows the distribution.

Table 8. ASCII Code and their frequency in SANAD dataset

No.	Code	Frequency	No.	Letter	Frequency
1	8	10961	24	121	52
2	57	6691	25	85	43
3	48	5762	26	93	40
4	20	4608	27	132	37
5	0	4427	28	185	30
6	6	2769	29	72	29
7	29	2013	30	68	27
8	4	1597	31	65	26
9	1	1312	32	157	25
10	43	940	33	136	25
11	21	936	34	70	23
12	18	714	35	112	19
13	17	701	36	129	18
14	36	370	37	134	17
15	3	253	38	131	15
16	22	227	39	149	9
17	64	172	40	67	9
18	52	138	41	82	7
19	148	135	42	81	4

20	84	123	43	176	4
21	146	68	44	145	4
22	107	63	45	100	2
23	128	53	56	195	1

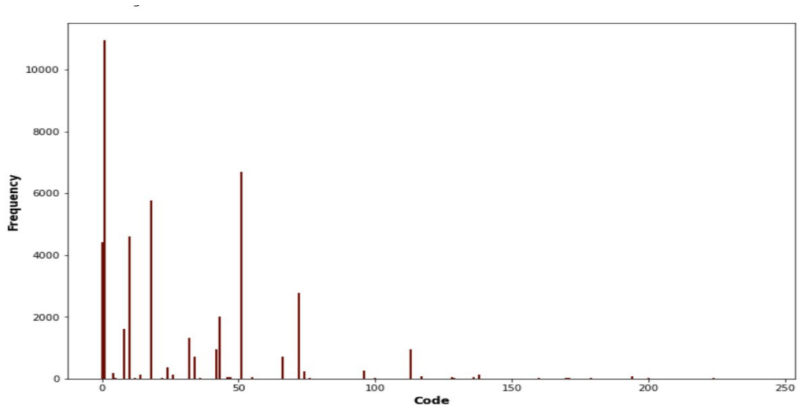


Figure 2. Distribution of ASCII Code in SANAD dataset

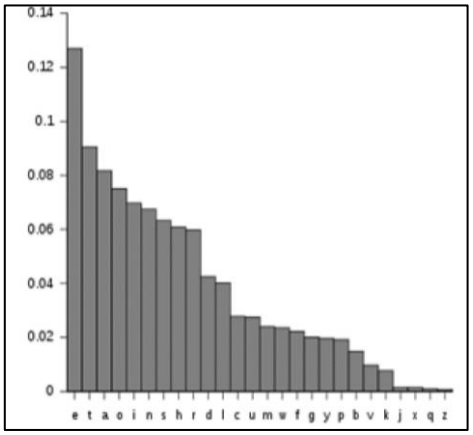
4.1.6. Create Dynamic English Letter Mapping

As shown in figure 2, the distribution of ASCII code lies in out range of English letters in the ASCII code. This required making a mapping to ensure that the English message was successfully hidden in the selected dataset.

The proposed method implemented dynamic English letter mapping to map the most frequent ASCII code with the most frequent English letters. The term "dynamic" denotes that mapping is changed dynamically based on the selected dataset itself. Figure 3 shows the frequency of English letters [12].

Letter	Frequency	Letter	Frequency
e	12.7020%	m	2.4060%
t	9.0560%	w	2.3600%
a	8.1670%	f	2.2280%
o	7.5070%	g	2.0150%
i	6.9660%	y	1.9740%
n	6.7490%	p	1.9290%
s	6.3270%	b	1.4920%
h	6.0940%	v	0.9780%
r	5.9870%	k	0.7720%
d	4.2530%	j	0.1530%
l	4.0250%	x	0.1500%
c	2.7820%	q	0.0950%
u	2.7580%	z	0.0740%

(a)



(b)

Figure 3. The English letters frequency (a): tabular data, (b) histogram [46].

The results of dynamic English letter mapping with the most frequent ASCII code in the selected news datasets and two datasets about Arabic poetry are explicated in Table 9.

Table 9. Dynamic English letter mapping results: ASCII frequencies in three datasets

SANAD				APCD				Arabic Poetry Dataset			
ASCII code	**Freq.**	**English letters**	**Mapping**	**ASCII code**	**Freq.**	**English letters**	**Mapping**	**ASCII code**	**Freq.**	**English letters**	**Mapping**
8	10,961	e	(8:'e')	43	485,718	E	(43:'e')	57	14,620	e	(57:'e')

57	6,691	t	(57:'t')	6	252,772	T	(6:'t')	0	8,086	t	(0:'t')
48	5,762	a	(48:'a')	8	226,772	A	(8:'a')	17	4,913	a	(17:'a')
20	4,608	o	(20:'o')	0	167,059	O	(0:'o')	29	4,639	o	(29:'o')
0	4,427	i	(0:'i')	20	125,386	I	(20:'i')	43	4,331	i	(43:'i')
6	2,769	n	(6:'n')	17	120,149	N	(17:'n')	20	4,018	n	(20:'n')
29	2,013	s	(29:'s')	1	104,239	S	(1:'s')	6	3,942	s	(6:'s')
4	1,597	h	(4:'h')	29	99,979	H	(29:'h)	1	3,718	h	(1:'h')
1	1,312	r	(1:'r')	4	55,470	R	(4:'r')	4	2,409	r	(4:'r')
43	940	d	(43:'d')	3	39,046	D	(3:'d')	48	1,833	d	(48:'d')
21	936	l	(21:'l')	48	36,294	L	(48:'l')	3	1,689	l	(3:'l')
18	714	c	(18:'c')	57	34,384	C	(57:'c')	18	1,104	c	(18:'c')
17	701	u	(17:'u')	21	26,130	U	(21:'u')	21	1,075	u	(21:'u')
36	370	m	(36:'m')	18	21,351	M	(18:'m')	36	302	m	(36:'m')
3	253	w	(3:'w')	36	8,892	W	(36:'w')	52	299	w	(52:'w')
22	227	f	(22:'f')	22	7,103	F	(22:'f')	22	272	f	(22:'f')
64	172	g	(64:'g')	52	5,530	G	(52:'g')	185	132	g	(185:'g')
52	138	y	(52:'y')	171	4,618	Y	(171:'y')	132	83	y	(132:'y')
148	135	p	(148:'p')	128	1,563	P	(128:'p')	171	73	p	(171:'p')
84	123	b	(84:'b')	134	1,525	B	(134:'b')	128	65	b	(128:'b')
146	68	v	(146:'v')	148	1,511	V	(148:'v')	148	48	v	(148:'v')
107	63	k	(107:'k')	132	1,426	K	(132:'k')	129	47	k	(129:'k')
128	53	j	(128:'j')	136	1,034	J	(136:'j')	145	24	j	(145:'j')
121	52	x	(121:'x')	176	694	X	(176:'x')	134	19	x	(134:'x')
85	43	q	(85:'q')	129	617	Q	(129:'q')	131	15	q	(131:'q')
93	40	z	(93:'z')	157	453	Z	(157:'z')	176	14	z	(176:'z')

4.2. Hiding And Extraction Procedures

On the sender side, the hiding procedure was started by preprocessing the secret message by removing special characters, converting them to lowercase, converting numbers to words, and tokenizing. Each character in a secret message token was converted to the corresponding ASCII code. Then, using pre-agreed dynamic English letters mapping. This protocol maps the most frequent ASCII code with the most frequent English letters. The successful matching was collected as the stego-text sent to the receiver. Figure 4 shows the hiding and extraction procedures.

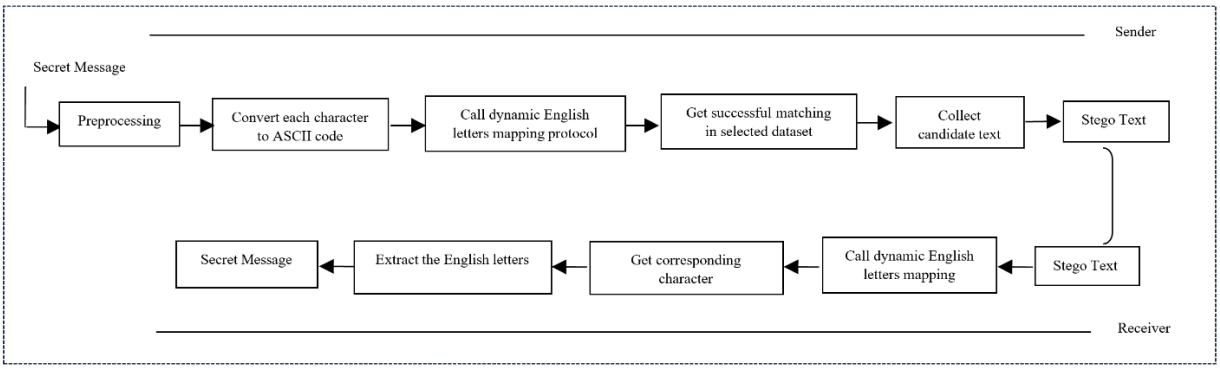


Figure 4. Hiding and extraction procedures

 ISSN 2723-6471

Example: assume the secret message = ‘hope’, Figure 5, figure 6, and figure 7 explain the details of the hiding procedure in three datasets.

	Secret Message	ASCII Code	Mapping	Matching in dataset	Built-in Features (Binary)	Built-in Features (ASCII)	Collected stego-text
0	h	104	(4: h)	خصصت	00000100	4	...خصصت مجلة تراث الصادره عن نادي تراث الإمارات م
1	o	111	(20: o)	تقيم	00010100	20	...تقيم الفدادة لينا كابيلوت معرضاً في مطلع العام
2	p	112	(148: p)	بدأ	10010100	148	...بدأ التشكيلي الإماراتي إبراهيم العوضي تجربته ا
3	e	101	(8: e)	أصدر	00001000	8	...أصدر قسم الدراسات والنشر في دائره الثقافة والإ

Figure 5. Hiding procedure results using SANAD dataset

	Secret Message	ASCII Code	Mapping	Matching in dataset	Built-in Features (Binary)	Built-in Features (ASCII)	Collected stego-text
0	h	104	(29: h)	يثوب	00011101	29	يثوب إليها كل ضيف وجانب كما رد دهداه الفلاص نصيحها
1	o	111	(0: o)	على	00000000	0	على غير ذنب أن أكون جنيته سوى قول باع كاذبي فتجهدا
2	p	112	(128: p)	عذراء	10000000	128	عذراء لم تجتل الخطاب بهجتها حتى اجتلاها عبادي بدينار
3	e	101	(43: e)	وإن	00101011	43	وإن تنظراني اليوم أقض لبانة وتستوجبا منا علي وتحمدا

Figure 6. Hiding procedure results using APCD dataset

	Secret Message	ASCII Code	Mapping	Matching in dataset	Built-in Features (Binary)	Built-in Features (ASCII)	Collected stego-text
0	h	104	(1: h)	منطرحا	00000001	1	... منطرحا امام بابك الكبير اصرخ في الظلام استجير
1	o	111	(29: o)	يا	00011101	29	...يا ضياء الحقول ياعنوه الفلاح في الساجيات من اس
2	p	112	(171: p)	ولجوء	10101011	171	...ولجوء قهرا لحياة الناس ترحل مثلما ناتي ويبقى ال
3	e	101	(57: e)	انا	00111001	57	...انا لا ازال و في يدي قدحي بالليل اين تفرق الشر

Figure 7. Hiding procedure results using Arabic Poetry dataset

As seen in figure 5, figure 6, and figure 7 the mapping protocol was dynamic based on the selected dataset, so the candidate text was different from one dataset to another.

On the receiver side, the extracting procedure is the opposite of the hiding procedure: beginning with the stego-text, calling the pre-agreed dynamic English letters mapping, and finally extracting the corresponding character to get the whole secret message. Figure 4 shows the hiding and extraction procedures.

5. Experiment Results

In this study, three databases (SANAD, APCD, and Arabic Poetry Dataset) were built from Kaggle, totaling about 829.61 MB. Unfortunately, there is no research on coverless information hiding in Arabic to compare with them, but in this work conducts several experiments to assess the performance of the proposed steganographic method in terms of hiding capacity, success rate, extracting accuracy, security analysis, availability, and validity of the algorithm, respectively.

5.1. Hiding Capacity

The number of characters or keywords that can be hidden in one text is called hiding capacity. The proposed method can hide character/word. Moreover, during the embedding process, the stego-texts do not need to embed the length of the secret message. Thus, it is helpful in improving hiding capacity. Many parameters determine the hiding capacity in the proposed method, such as secret message length, number of successful matchings, their frequencies, and the size of the selected dataset. For example, the decimal number 8 results in 10,961 matchings, which means it can hide the English letter (e) 10,961 times.

5.2. Success Rate

It refers to evaluating the performance of the embedding algorithm and extracted message quality. Let N is the number of secret messages; the success rate equation is defined by (1) [16]:

$$SR = S / N \quad (1)$$

Note: S is the number of successfully hidden messages, N is the total number of secret messages.

The hiding success rate of the proposed method can be 100% using dynamic English letters mapping protocol if the frequency of English letters of secret message <= frequency of the decimal number of matchings for any size of datasets.

5.3. Extracting Accuracy

It refers to evaluating the performance of the extracting algorithm (distance between the secret message and the extracted message). The algorithm of the proposed method can embed and extract secret messages correctly; thus, the distance is zero, and extracting accuracy is high.

5.4. Security Analysis

The security of the proposed steganography method is high level for three reasons. Firstly, there is no modification and no embedding in the selected cover. Secondly, English letter mapping, which maps the most frequent ASCII code with the most frequent English letters, provides an opportunity to ignore the original letters used. Finally, "dynamic" mapping indicates that mapping is changed dynamically based on the selected dataset. All that can improve the security of the proposed method which makes it stronger defense against format conversion attack and statistical detection.

5.5. Availability

The dataset size has an important impact on the method's availability. The proposed method can be extended to other built-in features such as the part of speech POS, n-grams, Name Entity Recognition NER, and so on.

Table 10 can summarize the comparison of proposed steganographic methods in terms of hiding capacity, success rate, extracting accuracy, security analysis, availability, and validity of the algorithm, with related works.

Table 10. Comparison of proposed method with related works

Methods	Hiding Capacity	Success Rate	Extracting Accuracy	Security Analysis	Availability
[2]	Low capacity, 14 or 15 bits in 200,000) texts			Resisting current detecting techniques, no modification in cover.	
[23]	High capacity 6.25% in best case and reached 25% in the improved method	High success rate		Resisting the format transformation attack	
[34]		Improve success rate	enhances extraction accuracy by increasing the number of keywords	withstands current steganalysis techniques	
[8]	high capacity	100%			
Our proposed method	Char per Word	100%	100%	Stronger defense against format conversion attack and statistical detection.	Can be extended to other built-in features

6. Conclusion and Future Work

In this paper, the researchers propose a new coverless text information hiding method based on built-in features of Arabic scripts. The process involved extracting the first word in each row, which was tested based on eight features to get one byte converted to a decimal number (ASCII code). Then, a dynamic mapping protocol was implemented with

the most frequently used letter. The proposed method withstands existing detection methods because it involves no modification or generation. Also, there is an enhancement in hiding capacity, which can conceal a (character per word).

A potential limitation of the proposed method is that a number of factors, including the size of the chosen dataset, the number of successful matches, the length of the secret message, and its frequencies, affect the hiding ability and success rate. Despite this, the proposed method has a high hiding success rate, high security, high availability, and excellent extraction accuracy compared to previous works.

In future work, there are three main issues to explore. The first is to improve hiding capacity by increasing the number of selected features to 16, aiming to conceal 2 bytes. The second is to embed an Arabic message utilizing 11 bits to encode Arabic letters. The third is to extend the current proposed method to built-in features of other languages, such as English, French, etc.

7. Declaration

7.1. Author Contributions

Conceptualization: S.H.R. and D.A.N.; Methodology: D.A.N.; Software: S.H.R.; Validation: S.H.R. and D.A.N.; Formal Analysis: S.H.R. and D.A.N.; Investigation: S.H.R.; Resources: D.A.N.; Data Curation: D.A.N.; Writing Original Draft Preparation: S.H.R. and D.A.N.; Writing Review and Editing: D.A.N. and S.H.R.; Visualization: S.H.R..; All authors have read and agreed to the published version of the manuscript.

7.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

7.3. Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

7.4. Institutional Review Board Statement

Not applicable.

7.5. Informed Consent Statement

Not applicable.

7.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Y. Luo and Y. Huang, “Text steganography with high embedding rate: Using recurrent neural networks to generate Chinese classic poetry,” in *ACM Workshop on Information Hiding and Multimedia Security*, vol. 1, no. 1, pp. 99–104, 2017.

[2] Z. Xia and X. Li, “Coverless Information Hiding Method Based on LSB of the Character’s Unicode,” *J. Internet Technol.*, vol. 18, no. 6, pp. 1353–1360, 2017, doi: 10.6138/JIT.2017.18.6.20160815b.

[3] O. F. A. Adeeb and S. J. Kabudian, “Arabic Text Steganography Based on Deep Learning Methods,” *IEEE Access*, vol. 10, no. September, pp. 94403–94416, 2022, doi: 10.1109/ACCESS.2022.3201019.

[4] S. S. Baawi, D. A. Nasrawi, and L. T. Abdulameer, “Improvement of ‘text steganography based on unicode of characters in multilingual’ by custom font with special properties,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 870, no. 1, 2020, doi: 10.1088/1757-899X/870/1/012125.

[5] S. Roy and M. Manasmita, "A novel approach to format based text steganography," in *Proceedings of the 2011 International Conference on Communication, Computing and Security*, vol. 1, no. 1, pp. 511-516, 2011.

[6] M. H. Shirali-Shahreza and M. Shirali-Shahreza, “Arabic/Persian text steganography utilizing similar letters with different codes Static Analysis of BPMN2.0 Process Models View project,” ResearchGate, vol. 35, no. 1B, pp. 213-218, Available:

https://www.researchgate.net/publication/268260454

[7] S. Sharma, A. Gupta, M. C. Trivedi, and V. K. Yadav, "Analysis of different text steganography techniques: A survey," in *Proceedings - 2016 2nd International Conference on Computational Intelligence and Communication Technology, CICT 2016, Institute of Electrical and Electronics Engineers Inc.*, vol. 1, no. 1, pp. 130–133, 2016. doi: 10.1109/CICT.2016.34.

[8] K. Wang, X. Yu, and Z. Zou, "A Coverless Text Steganography by Encoding the Chinese Characters' Component Structures," *Int. J. Digit. Crime Forensics*, vol. 13, no. 6, pp. 1-15, Nov. 2021, doi: 10.4018/IJDCF.20211101.oa4.

[9] W. Bhaya, A. M. Rahma, and D. AL-Nasrawi, "Text steganography based on font type in MS-word documents," *J. Comput. Sci*., vol. 9, no. 7, pp. 898–904, 2013, doi: 10.3844/jcssp.2013.898.904.

[10] S. S. Baawi, M. R. Mokhtar, and R. Sulaiman, "A comparative study on the advancement of text steganography techniques in digital media," *ARPN J. Eng. Appl. Sci*., vol. 13, no. 5, pp. 1854–1863, 2018.

[11] Yang, Z.; Jin, S.; Huang, Y.; Zhang, Y.; Li, H. Automatically generate steganographic text based on Markov model and Huffman coding. arXiv 2018, arXiv:1811.04720, pp.1-10. [Google Scholar].

[12] R. Thabit, N. I. Udzir, S. Md Yasin, A. Asmawi, N. A. Roslan, and R. Din, "A comparative analysis of arabic text steganography," *Appl. Sci*., vol. 11, no. 15, pp.2-32, 2021, doi: 10.3390/app11156851.

[13] A. M. S. Rahma, W. S. Bhaya, and D. A. Alnasrawi, "Data Hiding Method for English Scripts using Ligature Characters Unicode," *Eur. J. Sci. Res*., vol. 112, no. 4, pp. 452–459, 2013.

[14] A. M. S. Rahma, W. S. Bhaya, D. A. A. International, P. Abdul, M. S. Rahma, and D. A. Al-, "Text Steganography Based on Unicode of Characters in Multilingual," *Int. J. Eng. Res. Appl*., vol. 3, no. 4, pp. 1153–1165, 2013.

[15] N. Wu et al., "STBS-Stega: Coverless text steganography based on state transition-binary sequence," *Int. J. Distrib. Sens. Networks*, vol. 16, no. 3, pp.2-12, 2020, doi: 10.1177/1550147720914257.

[16] B. Guan, L. Gong, and Y. Shen, "A Novel Coverless Text Steganographic Algorithm Based on Polynomial Encryption," *Secur. Commun. Networks*, vol. 2022, no. 1, pp.12, 2022, doi: 10.1155/2022/1153704.

[17] S. Ali, "A State-of-the-Art Survey of Coverless Text Information Hiding," *Int. J. Comput. Netw. Inf. Secur*., vol. 10, no. 7, pp. 52–58, Jul. 2018, doi: 10.5815/ijcnis.2018.07.06.

[18] Z. Fu, H. Ji, and Y. Ding, "Label model based coverless information hiding method," *J. Internet Technol*., vol. 19, no. 5, pp. 1509–1514, 2018, doi: 10.3966/160792642018091905022.

[19] H. Ji and Z. Fu, "Coverless information hiding method based on the keyword," *Int. J. High. Perform. Compu. and Networking,* vol.14, no.1, pp.1–7, 2019.

[20] L. Xiang, J. Qin, X. Xiang, Y. Tan, and N. N. Xiong, "A robust text coverless information hiding based on multi-index method," *Intell. Autom. Soft Comput*., vol. 29, no. 3, pp. 899–914, 2021, doi: 10.32604/iasc.2021.017720.

[21] S. Shi, Y. Qi, and Y. Huang, ''An approach to text steganography based on search in Internet,'' in *Proc. Int. Comput. Symp. (ICS),* vol. 2016, no. Dec., pp. 227–232, Dec. 2016.

[22] X. Chen, H. Sun, Y. Tobe, Z. Zhou, and X. Sun, ''Coverless information hiding method based on the Chinese mathematical expression,'' *in Proc. Int. Conf. Cloud Comput. Secur., in Lecture Notes in Computer Science, Z. Huang, X. Sun, J. Luo, and J. Wang, Eds. Nanjing, China,* vol. 9483, no. 1, pp. 133–143, 2015.

[23] K. Wang and Q. Gao, "A Coverless Plain Text Steganography Based on Character Features," *IEEE Access*, vol. 7, no. 1, pp. 95665–95676, 2019, doi: 10.1109/ACCESS.2019.2929123.

[24] A. I. Alaqeel and M. S. Saleh, "Developing a Performance-based Tool for Arabic Text Steganography," *in Proceedings - 2021 IEEE 4th National Computing Colleges Conference, NCCC 2021, Institute of Electrical and Electronics Engineers Inc.*, vol. 1, no. 1, pp. 1-12, Mar. 2021. doi: 10.1109/NCCC49330.2021.9428837.

[25] R. Thabit, N. I. Udzir, S. Md Yasin, A. Asmawi, N. A. Roslan, and R. Din, "A comparative analysis of arabic text steganography," *Applied Sciences (Switzerland),* vol. 11, no. 15. MDPI AG, Aug. 01, pp. 1-12, 2021. doi: 10.3390/app11156851.

[26] A. Taha, A. S. Hammad, and M. M. Selim, "A high capacity algorithm for information hiding in Arabic text," *J. King Saud Univ. - Comput. Inf. Sci*., vol. 32, no. 6, pp. 658–665, Jul. 2020, doi: 10.1016/j.jksuci.2018.07.007.

[27] B. Al-Salemi, M. Ayob, G. Kendall, and S. A. M. Noah, "Multi-label Arabic text categorization: A benchmark and baseline comparison of multi-label learning algorithms," *Inf. Process. Manag.*, vol. 56, no. 1, pp. 212–227, 2019, doi: 10.1016/j.ipm.2018.09.008.

[28] A. S. Issn and D. A. Al-nasrawi, "From Arabic Alphabets to Two Dimension Shapes in Kufic Calligraphy Style Using Grid Board Catalog Dhamyaa A. AL-Nasrawi, Ahmed F. Almukhtar and Wafaa S. AL-Baldawi", *Commun. Appl. Sci.*, vol. 3, no. 2, pp. 42–59, 2015.

[29] N. Alifah Roslan, N. Izura Udzir, R. Mahmod, and A. Gutub, "Systematic literature review and analysis for Arabic text steganography method practically," *Egypt. Informatics J.*, vol. 23, no. 4, pp. 177–191, 2022, doi: 10.1016/j.eij.2022.10.003.

[30] A. A., F. Ridzuan, and S. Ali, "Text Steganography using Extensions Kashida based on the Moon and Sun Letters Concept," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 8, pp. 286-290, 2017, doi: 10.14569/ijacsa.2017.080838.

[31] N. A. Roslan, R. Mahmod, and N. I. Udzir, "Sharp-edges method in Arabic text steganography," *J. Theor. Appl. Inf. Technol.*, vol. 33, no. 1, pp. 32–41, 2011.

[32] E. A. Khan, "Using arabic poetry system for steganography," *Asian J. Comput. Sci. Inf. Technol.*, vol. 4, no. 6, pp. 55–61, 2014, doi: 10.15520/ajcsit.v.

[33] A. A. Obeidat, "Arabic text steganography using Unicode of non-joined to right side letters," *J. Comput. Sci.*, vol. 13, no. 6, pp. 184–191, Jun. 2017, doi: 10.3844/jcssp.2017.184.191.

[34] Y. Liu, J. Wu, and X. Chen, "An improved coverless text steganography algorithm based on pretreatment and POS," *KSII Trans. Internet Inf. Syst.*, vol. 15, no. 4, pp. 1553–1567, Apr. 2021, doi: 10.3837/tiis.2021.04.020.

[35] Y. Wu and X. Sun, "Text coverless information hiding method based on hybrid tags," *J. Internet Technol.*, vol. 19, no. 3, pp. 649–655, 2018, doi: 10.3966/160792642018051903003.

[36] Odeh A, Alzubi A, Hani QB, Elleithy K (2012)," Steganography by multipoint Arabic letters", In: *2012 IEEE long Island systems, applications and technology conference (LISAT). IEEE,* vol. 1, no. 1, pp 1–7, 2012.

[37] A. Gutub, W. Al-Alwani, and A. Mahfoodh, "Improved Method of Arabic Text SteganographyUsing the Extension „Kashida" Character," *Bahria Univ. J. Inf.*, vol. 3, no. 1, pp. 68–72, 2010.

[38] A. Odeh, K. Elleithy, and M. Faezipour, "Steganography in Arabic text using Kashida variation algorithm (KVA)," in *IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, vol. 1, no. 1, pp. 1-7, 2013.

[39] A. Odeh and K. Elleithy, "Steganography in Arabic Text Using Full Diacritics Text," presented at the 25th International Conference on Computers and Their Applications in Industry and Engineering (CAINE-2012), New Orleans, Louisiana, USA, 2012.

[40] O. Article, "Integrating Light-Weight Cryptography with Diacritics Arabic Text Stegan- ography Improved for Practical Security Applications", *J. Inf. Secur. Cybercrimes Res. 2020,* vol. 3, no. 1, pp. 13–30, 2020, doi: 10.26735/FMIT1649.

[41] N. Alanazi, E. Khan, and A. Gutub, "Functionality-Improved Arabic Text Steganography Based on Unicode Features," *Arab. J. Sci. Eng.*, vol. 45, no. 12, pp. 11037–11050, Dec. 2020, doi: 10.1007/s13369-020-04917-5.

[42] "SANAD." https://www.kaggle.com/datasets/haithemhermessi/sanad-dataset?select=Tech

[43] "APCD," kaggle. https://www.kaggle.com/datasets/mohamedkhaledelsafty/best-arabic-poem-comprehensive-dataset

[44] "Arabic Poetry Dataset," kaggle. https://www.kaggle.com/datasets/fahd09/arabic-poetry-dataset-478-2017

[45] Mohsen Madi "A study of Arabic letter frequency analysis." https://www.intellaren.com/articles/en/a-study-of-arabic-letter-frequency-analysis,2010

[46] E. Agyepong, W. J. Buchanan, and K. Jones, "Detection of Algorithmically Generated Malicious Domain," *in Conference: 6th International Conference of Advanced Computer Science & Information Technology*, vol. 6, no. 1, pp. 13–32, 2018, doi: 10.5121/csit.2018.80802.