Computers, Materials & Continua
DOI:10.32604/cmc.2022.018591
Article

Tech Science Press

Smart-Fragile Authentication Scheme for Robust Detecting of Tampering Attacks on English Text

Mohammad Alamgeer[1], Fahd N. Al-Wesabi[2,3,*], Huda G. Iskandar[3,4], Imran Khan[5], Nadhem Nemri[6], Mohammad Medani[6], Mohammed Abdullah Al-Hagery[7] and Ali Mohammed Al-Sharafi[3,8]

[1]Department of Information Systems, King Khalid University, Muhayel Aseer, Kingdom of Saudi Arabia
[2]Department of Computer Science, King Khalid University, Muhayel Aseer, Kingdom of Saudi Arabia
[3]Faculty of Computer and IT, Sana'a University, Sana'a, Yemen
[4]School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, Malaysia
[5]Department of Electrical Engineering, University of Engineering and Technology Peshawar, Pakistan
[6]Department of Information Systems, King Khalid University, Muhayel Aseer, Kingdom of Saudi Arabia
[7]Department of Computer Science, College of Computer, Qassim University, Saudi Arabia
[8]Department of Computer Science, College of Computers and Information Technology, University of Bisha, KSA
[*]Corresponding Author: Fahd N. Al-Wesabi. Email: fwesabi@gmail.com

Received: 12 March 2021; Accepted: 13 April 2021

Abstract: Content authentication, integrity verification, and tampering detection of digital content exchanged via the internet have been used to address a major concern in information and communication technology. In this paper, a text zero-watermarking approach known as Smart-Fragile Approach based on Soft Computing and Digital Watermarking (SFASCDW) is proposed for content authentication and tampering detection of English text. A first-level order of alphanumeric mechanism, based on hidden Markov model, is integrated with digital zero-watermarking techniques to improve the watermark robustness of the proposed approach. The researcher uses the first-level order and alphanumeric mechanism of Markov model as a soft computing technique to analyze English text. Moreover, he extracts the features of the interrelationship among the contexts of the text, utilizes the extracted features as watermark information, and validates it later with the studied English text to detect any tampering. SFASCDW has been implemented using PHP with VS code IDE. The robustness, effectiveness, and applicability of SFASCDW are proved with experiments involving four datasets of various lengths in random locations using the three common attacks, namely insertion, reorder, and deletion. The SFASCDW was found to be effective and could be applicable in detecting any possible tampering.

Keywords: Watermarking; soft computing; text analysis; hidden Markov model; content authentication

1 Introduction

For the research community, the reliability and security of exchanged text data through the internet is the most promising and challenging field. In communication technologies, authentication of content and automated text verification of honesty in different Languages and formats are of great significance. Numerous applications for instance; e-Banking and e-commerce render information transfer via the Internet the most difficult. In terms of content, structure, grammar, and semantics, much of the digital media transferred over the internet is in text form and is very susceptible to online transmission. During the transfer process, malicious attackers can temper such digital content [1].

For information security, many algorithms and techniques are available such as the authentication of content, verification of integrity, detection of tampering, identification of owners, access control, and copyright protection.

To overcome these issues, steganography and automated methods of watermarking are commonly used. A technique of digital-Watermarking (DWM) can be inserted into digital material through various details such as text, binary pictures, audio, and video [2,3]. A fine-grained text watermarking procedure is proposed based on replacing the white spaces and Latin symbols with homoglyph characters [4].

Several conventional methods and solutions for text watermarking were proposed [5,6] and categorized into different classifications such as linguistic, structure and image-based, and format-based binary images [7]. To insert the watermark information into the document, most of these solutions require certain upgrades or improvements to the original text in digital format material. Zero-watermarking without any alteration to the original digital material to embed the watermark information is a new technique with smart algorithms that can be used. Also, this technique can be used to generate data for a watermark in the contents of a given digital context [1,7–9].

Restricted research has centered on the appropriate solutions to verify the credibility of critical digital media online [10–12]. The verification of digital text and the identification of fraud in research earned great attention. In addition, text watermarking studies have concentrated on copyright protection in the last decade, but less interest and attention has been paid to integrity verification, identification of tampering and authentication of content due to the existence of text content based on the natural language [13].

Proposing the most appropriate approaches and strategies for dissimilar formats and materials, especially in Arabic and English languages, is the most common challenge in this area [14,15]. Therefore, authentication of content, verification of honesty, and detection of tampering of sensitive text constitute a big problem in various applications and require necessary solutions.

Some instances of such sensitive digital text content are Arabic interactive Holy Qur'an, eChecks, tests, and marks. Different Arabic alphabet characteristics such as diacritics lengthened letters and extra symbols of Arabic make it simple to modify the key meaning of the text material by making basic changes such as modifying diacritic arrangements [16]. The most popular soft computation and natural language processing (NLP) technique that supported the analysis of the text is HMM.

We suggest a highly fragile method for detecting the tampering attacks on Internet-based Arabic text (HFDATAI) by incorporating the Markov model and zero watermarking. Hence, first-order of an alphanumeric mechanism consisting of a model performing as a soft computing tool and NLP in cooperation between the zero-watermarking technique and the Markov model. In this method, for text analysis, the first order of the alphanumeric mechanism of the Markov model was used to extract the connections between the contents of the Arabic text given and to generate the main watermark. Without alterations or effects on the original text size, the watermark created is logically integrated into the original Arabic history. The embedded watermark would later be used to identify all manipulation on Arabic text obtained after transmission of text through the Internet and whether it is authentic or not.

The primary objective of the HFDATAI strategy is to meet the high accuracy of content authentication and identification of sensitive tampering attacks of Arabic text which is transmitted through the Internet.

The remainder of the article is structured as follows: In Section 2, we explain the existing works done so far. In Section 3, we discussed the suggested approach (HFDATAI). The simulation and implementation are provided in Section 4, results discussion is provided in Section 5, and finally, we conclude the article in Section 6.

2 Related Work

According to the processing domain of NLP and text watermarking, these existing methods and solutions of text watermarking reviewed in this paper are classified into linguistical, structural and zero-watermark methods [1,7,13].

Natural language is the foundation of approaches to linguistic text watermarking. The mechanism of those methods embedding the watermark is based on changes applied to the semantic and syntactic essence of plain text [1].

To enhance the capability and imperceptibility of Arabic text, a method of text watermarking has suggested based on location of the accessible words [17]. In this method, any word-space is used to mask the Boolean bit 0 or 1 that physically modifies the original text.

A text steganography technique was proposed to hide information in the Arabic language [18]. The step of this approach considers Harakat's existence in Arabic diacritics such as Kasra, Fatha, and Damma as well as reverses Fatha to cover the message.

A Kashida-marks invisible method of watermarking [19], based on the features of frequent recurrence of document security and authentication characters, was proposed. The method is based on a predetermined watermark key with a Kashida placed for a bit 1 and a bit omitted.

The method of steganography of the text has proposed based on Kashida extensions on the characters 'moon' and 'sun' to write digital contents of the Arabic language [20]. In addition, the Kashida method characters are seen alongside characters from Arabic to decide which hidden secret bits are kept by specific characters. In this form, four instances are included in the kashida characters: moon characters representing '00'; sun characters representing '01'; sun characters representing '10'; and moon characters representing '11'.

A text steganographic approach [21] based on multilingual Unicode characters has been suggested to cover details in English scripts for the use of the English Unicode alphabet in other languages. Thirteen letters of the English alphabet have been chosen for this approach. It is important to embed dual bits in a timeframe that used ASCII code for embedding 00. However, multilingual ones were used by Unicode to embed between 01, and 10, as well as 11. The algorithm of Text Watermarking is used to secure textual contents from malicious attacks according to Unicode extended characters [22]. The algorithm requires three main steps, the development, incorporation, and extraction of watermarks. The addition of watermarks is focused on the development of predefined coding tables, while scrambling strategies are often used in generating and removing the watermarking key is safe.

The substitution attack method focused on preserving the position of words in the text document has been proposed [23]. This method depends on manipulating word transitions in the text document. Authentication of Chinese text documents based on the combination of the properties of sentences and text-based watermarking approaches have been suggested [24,25]. The proposed method is presented as follows: a text of the Chinese language is split into a group of sentences, and for each word, the code of a semantic has been obtained. The distribution of semantic codes influences sentence entropy.

A zero-watermarking method has been proposed to preserve the privacy of a person who relies on the Hurst exponent and the nullity of the frames [26]. For watermark embedding, the two steps are determined to evaluate the unvoiced frames. The process of the proposed approach bases on integrating an individual's identity without notifying any distortion in the signals of medical expression.

A zero-watermarking method was proposed to resolve the security issues of text-documents of the English language, such as verification of content and copyright protection [27]. A zero-watermarking approach has been suggested based on the authentication Markov-model of the content of English text [28,29]. In this approach, to extract the safe watermark information, the probability characteristics of the English text are involved and stored to confirm the validity of the attacked text-document. The approach provides security against popular text attacks with a watermark distortion rate if, for all known attacks, it is greater than one. For the defense of English text by copyright, based on the present rate of ASCII non-vowel letters and terms, the conventional watermark approach [30] has been suggested.

According to the suggested methods, content authentication and tampering detection of digital English contents that have been ignored by researchers in the literature for many reasons. English text is a natural language dependent. On the other hand, hiding the watermark information is complicated since there is no location to hide it within text as pixels in the case of image, waves in audio and frames in video.

3 The Proposed Approach

In this paper, the authors propose a smart scheme called SFASCDW by integrating zero-watermark and soft computing techniques with get rid of external watermark information and without modifications of the original text to embed the watermark key. The first-level of alphanumeric mechanism of the Markov model is used as a soft computing technique to analyze the contents of the given English text and extract the interrelationship features of these text contents.

The main contributions of our scheme SFASCDW can be summarized as follows:

- Unlike previous work where watermarking is done with language, contents, and scale effecting, the SFASCDW scheme logically embeds watermarking with no effect on text, content, or size.
- The watermarking mechanism does not require any external knowledge in our SFASCDW approach since this watermark key is generated by text processing and the extraction of a relationship between both the content and a watermark.
- The SFASCDW scheme is highly vulnerable to any basic alteration to the English text and context defined as complex text. Somehow, the above three contributions are present only in pictures, though not in the text. That is the key argument on this paper's contribution.

Four main processes should be performed in SFASCDW, namely text analysis, watermark generation, extraction, and detection processes, as illustrated in Fig. 1.

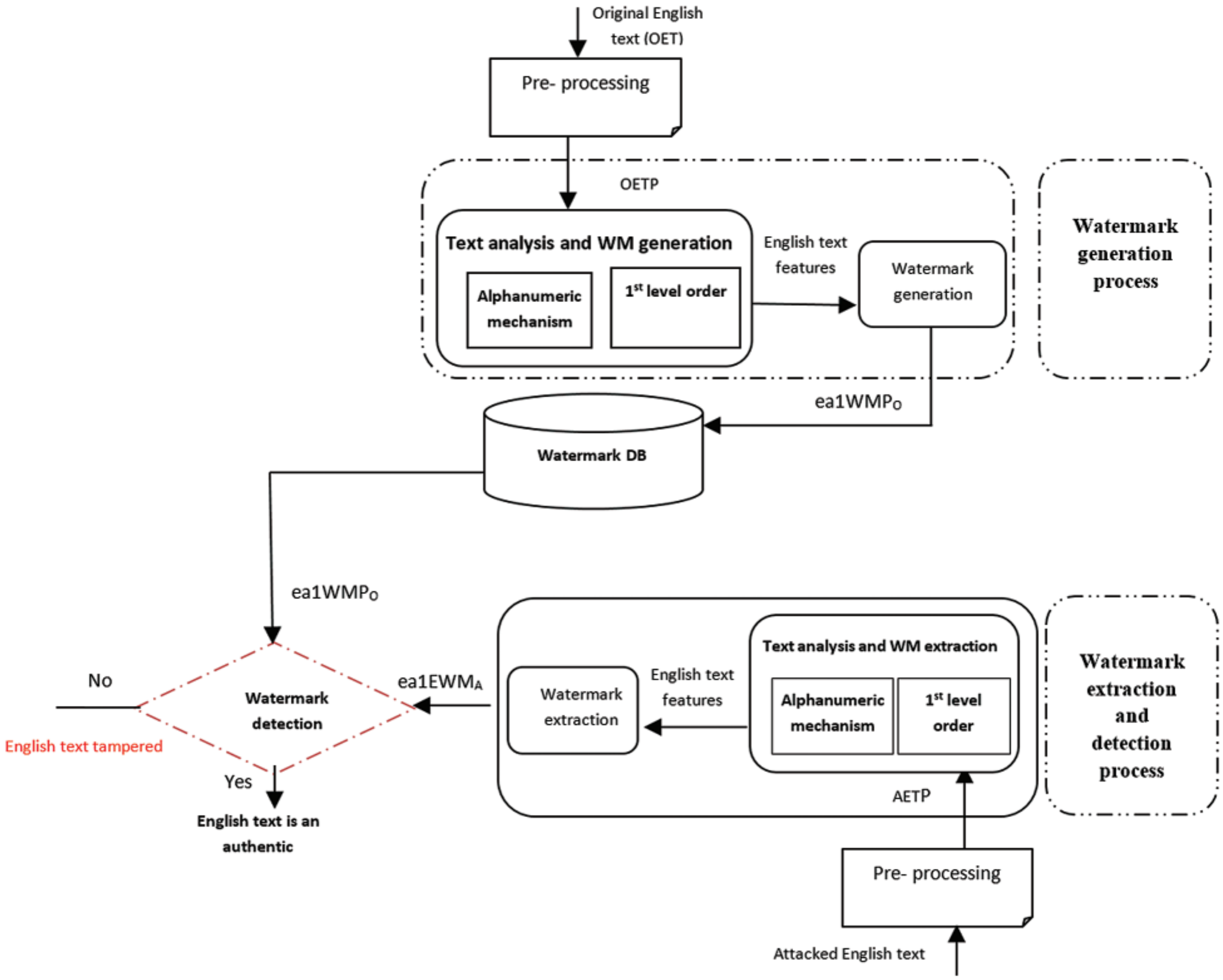


Figure 1: A general proposed model of the SFASCDW scheme

3.1 Text Analysis and Watermark Generation Process

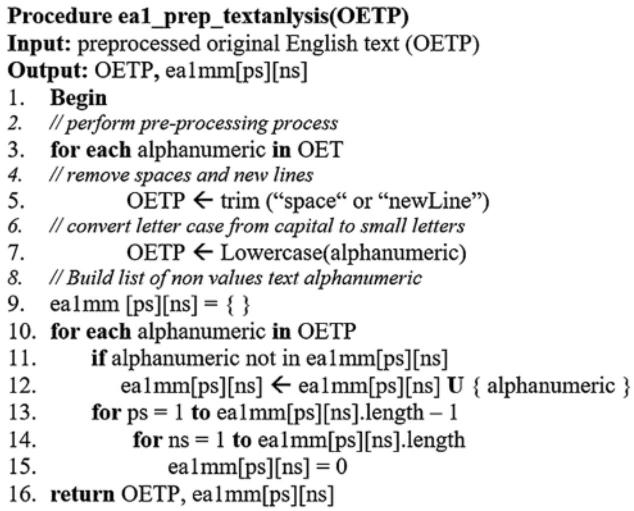
The three main sub-algorithms included in this process are preprocessing and building the Markov matrix algorithm, text analysis algorithm, and watermark generation algorithm.

3.1.1 Pre-Processing and Building the Markov Matrix Algorithm

The preprocessing of the original English text is one of the key steps in both the watermark generation and extraction processes of SFASCDW to convert letter cases and remove extra spaces and new lines, and it will directly influence the tampering detection accuracy. The original English text (OET) is required as input for the preprocessing process. The output of this algorithm is preprocessed English text (OET_P). That said, building a Markov matrix is the starting point of English text analysis and watermark generation process using the Markov model. The Markov matrix that represents the possible states and transitions available in the given text is constructed without reputations. In this approach, each unique alphanumeric within a given English text represents a present state, and each unique alphanumeric a transition in the Markov matrix. During the building process of the Markov matrix, the proposed algorithm initializes all transition values by zero to use these cells later for keeping track of the number of times that the ith unique alphanumeric is followed by the jth alphanumeric within the given English text.

Preprocessing and building the Markov matrix algorithm is executed as presented below in Algorithm 1.

Algorithm 1: Preprocessing and building Markov algorithm of SFASCDW



```
Procedure ea1_prep_textanlysis(OETP)
Input: preprocessed original English text (OETP)
Output: OETP, ea1mm[ps][ns]
Begin
// perform pre-processing process
for each alphanumeric in OET
// remove spaces and new lines
        OETP ← trim (“space“ or “newLine”)
// convert letter case from capital to small letters
        OETP ← Lowercase(alphanumeric)
// Build list of non values text alphanumeric
ea1mm [ps][ns] = { }
for each alphanumeric in OETP
    if alphanumeric not in ea1mm[ps][ns]
        ea1mm[ps][ns] ← ea1mm[ps][ns] U { alphanumeric }
    for ps = 1 to ea1mm[ps][ns].length – 1
        for ns = 1 to ea1mm[ps][ns].length
            ea1mm[ps][ns] = 0
return OETP, ea1mm[ps][ns]
```

where OET: is an original English text, OETP: is a preprocessed English text, ea1mm[ps][ns]: is a states and transitions matrix with zero values for all cells, ps: refers to present state, ns: refers to next state.

According to this algorithm, a method is presented to construct a two-dimensional matrix of Markov states and transitions named ea1mm[ps][ns], which represents the backbone of Markov model for English text analysis. The length of ea1mm[ps][ns] matrix of SFASCDW is fixed with sixty-one rows (states) and sixty-one columns (transitions). This fixed size represent twenty-eight alphabets of English letters “a–z,” ten integer numbers “0–9,” and twenty-three space and special symbols “@, $, (, /, . etc.”

3.1.2 Text Analysis Algorithm

After the Markov matrix is constructed, the text analysis process should be performed to find interrelationships between contexts of the given English text and generate watermark patterns. In this algorithm, the number of appearances of possible next-state transitions for each current state of a single alphanumeric will be calculated and constructed as transition probabilities by Eq. (1).

$$ea1mm[ps][ns] = \sum_{i,j=1}^{n-1} total\ \ Number\ \ of\ \ Transition(i,\ j) \tag{1}$$

where ea1mm[ps][ns] represents the Markov matrix of SFASCDW.

The following example of an English text sample describes the mechanism of the transition process of present state to other next states.

“The quick brown fox jumps over the brown fox who is slow jumps over the brown fox who is dead.”

When using the first-level order of alphanumeric mechanism of HMM, every unique alphanumeric is a present state. Text analysis is performed as the text is read to obtain the interrelationship between present state and the next states. Fig. 2 illustrates the available transitions of the above sample of English text.

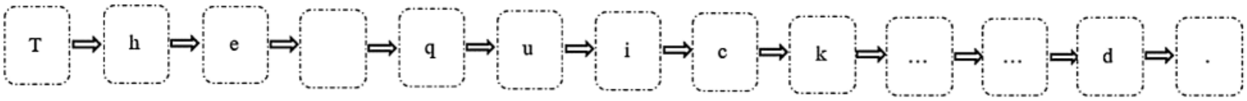
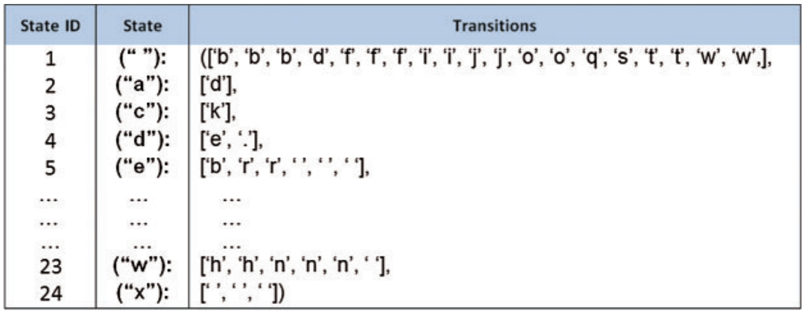


Figure 2: States representation of english text sample using SFASCDW

As a result of analyzing the given English sentence based on first-level order and alphanumeric mechanism of Markov model, the SFASCDW produced forty-nine unique present states and their sixty-one possible transitions as illustrated in Fig. 3.



State ID	State	Transitions
1	(“ ”):	([‘b’, ‘b’, ‘b’, ‘d’, ‘f’, ‘f’, ‘f’, ‘i’, ‘i’, ‘j’, ‘j’, ‘o’, ‘o’, ‘q’, ‘s’, ‘t’, ‘t’, ‘w’, ‘w’,],
2	(“a”):	[‘d’],
3	(“c”):	[‘k’],
4	(“d”):	[‘e’, ‘.’],
5	(“e”):	[‘b’, ‘r’, ‘r’, ‘ ’, ‘ ’, ‘ ’],
...	...	...
...	...	...
...	...	...
23	(“w”):	[‘h’, ‘h’, ‘n’, ‘n’, ‘n’, ‘ ’],
24	(“x”):	[‘ ’, ‘ ’, ‘ ’])

Figure 3: Sample of english text states and their transitions using SFASCDW

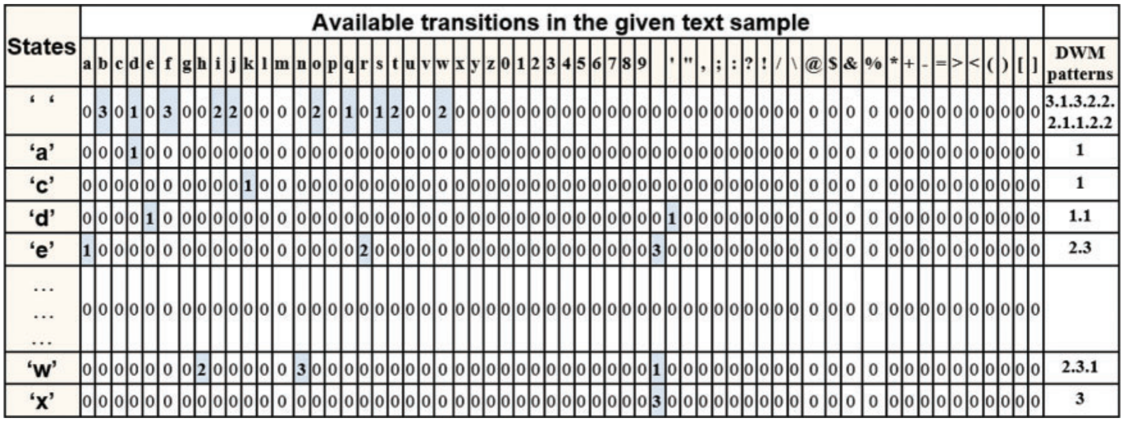
It is assumed here that ‘w’ is a present state of unique alphanumeric, and the available next transitions in the given English text sample are ‘h’, ‘h’, ‘n’, ‘n’, ‘n’, and ‘.’ It is observed that ‘h’ transitions occurred twice, ‘n’ transitions occurred three times, and ‘ ’ transitions occurred only one time.

The building of the Markov matrix and text analysis processes of the given English text sample are illustrated in Fig. 4.

3.1.3 Watermark Generation Algorithm

After performing analysis of the English text and extracting probability features, the watermark key is generated by finding all nonzero values in Markov matrix. The values in Markov matrix are concatenated sequentially to generate the original watermark pattern ea1WMPo, as given in Eq. (2) and illustrated in Fig. 5.

$$ea1WMP_O\& = ea1mm[ps][ns] \tag{2}$$



States	Available transitions in the given text sample																																																												
	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	0	1	2	3	4	5	6	7	8	9		'	"	,	;	:	?	!	/	\	@	$	&	%	*	+	-	=	>	<	(	)	[	]	DWM patterns
‘ ‘	0	3	0	1	0	3	0	0	2	2	0	0	0	0	2	0	1	0	1	2	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.1.3.2.2.2.1.1.2.2
‘a’	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
‘c’	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
‘d’	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.1
‘e’	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.3
… … …	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
‘w’	0	0	0	0	0	0	0	2	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.3.1
‘x’	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3

Figure 4: Text analysis processes of the given english text sample using SFASCDW

3.1.3.2.2.2.1.1.2.2 −1−1−
1.1− 2.3 − ... − 2.3.1 − 3

Figure 5: The generated original watermark patterns ea1WMPo

The generated ea1WMPo is stored in the watermark database beside the basic information of the given English text. The overall working of text analysis and watermark generation algorithm is illustrated below in Algorithm 2.

Algorithm 2: Text analysis and watermark generation algorithm of SFASCDW

```
Procedure ea1_wm_generation(OETP)
Input: preprocessed original English text (OETP)
Output: ea1WMP$_O$
Begin
// perform pre-processing and building Markova matrix process
ea1_prep_textanlysis(OETP)
// text analysis
pa = first_alphaumeric(OETP)
pd2 = OETP – pa // begin with 2nd alphanumeric
for each a in pd2
    ea1mm[pa][ca] = ea1mm[pa][a] + 1
    pa = ca
// watermark generation
for ps = 1 to ea1mm[ps][ns].length - 1,
    for ns = 1 to ea1mm[ps][ns].length,
        if ea1mm[ps][ns] != 0
            ea1WMP$_O$ &= ea1mm [ps] [ns]
return ea1WMP$_O$
```

where ea1WMP$_O$ is the original watermark.

3.2 Watermark Extraction and Detection Process

The attacked watermark patterns (ea1EWM$_A$) are generated for the attacked English text document (AET$_P$), and the matching rate of patterns and watermark distortion are calculated by SFASCDW in order to detect any tampering, which ensures the authenticity of given contents.

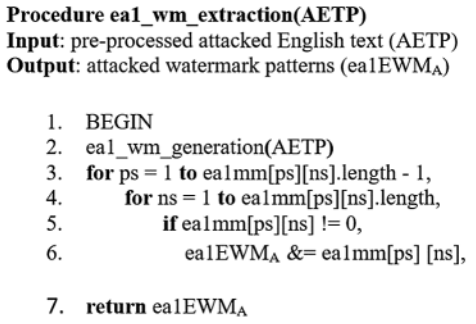
Two core algorithms involved in this process are watermark extraction and watermark detection algorithm. However, ea1EWM$_A$ is extracted from the received AET$_P$ and matched with ea1WMPo by the detection algorithm.

AET$_P$ is an input for the proposed watermark extraction algorithm. The same process is performed for the watermark generation algorithm to obtain the watermark pattern for AET$_P$.

3.2.1 Watermark Extraction Algorithm

AET$_P$ is the main input required to run this algorithm. However, the output of this algorithm is ea1EWM$_A$. The watermark extraction algorithm is illustrated below in Algorithm 3.

Algorithm 3: Watermark extraction algorithm of SFASCDW



```
Procedure ea1_wm_extraction(AETP)
Input: pre-processed attacked English text (AETP)
Output: attacked watermark patterns (ea1EWM$_A$)

BEGIN
ea1_wm_generation(AETP)
for ps = 1 to ea1mm[ps][ns].length - 1,
    for ns = 1 to ea1mm[ps][ns].length,
        if ea1mm[ps][ns] != 0,
            ea1EWM$_A$ &= ea1mm[ps] [ns],

return ea1EWM$_A$
```

where ea1EWM$_A$ is the attacked watermark.

3.2.2 Watermark Detection Algorithm

ea1EWM$_A$ and ea1WMPo are the main inputs required to run the watermark detection algorithm. However, the output of this algorithm is to determine whether the English text document is authentic or tampered. The detection process of the extracted watermark is achieved in two main phases:

- Primary matching is achieved for ea1WMP$_O$ and ea1EWM$_A$. If ea1EWM$_A$ and ea1WMP$_O$ patterns appear identical, then an alert will appear as follows: “English text document is authentic and no tampering occurred.” Otherwise, the notification will be “This English text document is tampered,” before continuing to the next phase.

- Secondary matching is achieved by matching the transition of each state in a generated pattern. This means that ea1EWM$_A$ of each state is compared with the equivalent transition of ea1WMP$_O$ as given by Eqs. (3) and (4) below

$$\mathrm{ea1PMR_T}\left(i,\ j\right)\left|\frac{ea1\,WMP_O\left[i\right]\left[j\right]-\left(ea1\,WMP_O\left[i\right]\left[j\right]-ea1EWM_A\left[i\right]\left[j\right]\right)}{ea1\,WMP_O\left[i\right]\left[j\right]}\right| \tag{3}$$

where, $ea1\_PMR_T$: represents tampering detection accuracy rate value in transition level, $(0 < ea1\_PMR_T <= 1)$

$$ea1PMR_S\left(i\right)=\left|\frac{\sum_{j=1}^{n-1}\left(\mathrm{ea1PMR_T}\left(i,\ j\right)\right)}{Total\ \ State\ \ Pattern\ \ Count\left(i\right)}\right| \tag{4}$$

where ea1_PMR$_S$: value of tampering detection accuracy rate in state level, (0 < ea1_PMR$_S$ <= 100).

After the pattern-matching rate of every state that is produced, the proposed SFASCDW scheme finds the weight (watermark robustness) of every state from all the states in the Markov matrix by using Eq. (5) below.

$$\mathrm{ea1Sw}=\sum_{i=0}^{n-1}\left|\frac{\mathrm{ea1}PMR_S(i)*Transitions\ frequency(i)}{total\ \ number\ \ of\ \ transitions}\right| \tag{5}$$

where ea1PMR refers to total watermark robustness.

The final ea1PMR of OET$_P$ and AET$_P$ are calculated by Eq. (6).

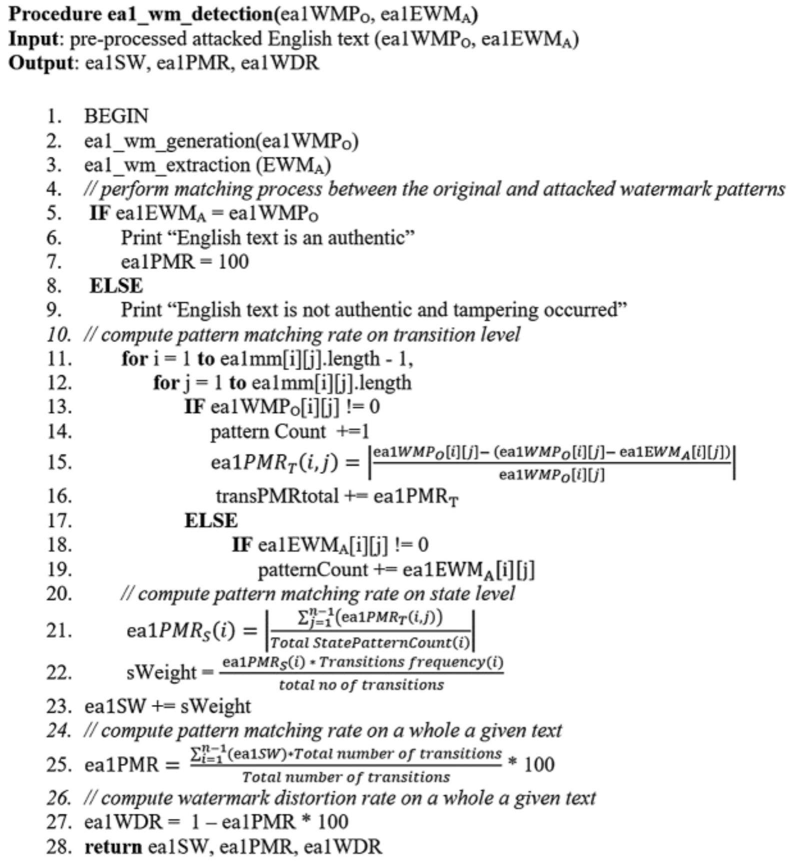
$$ea1PMR=\left|\frac{\sum_{i=1}^{n-1}\mathrm{ea1}PMRS(i)}{N}\right|*100 \tag{6}$$

The Watermark distortion rate represents the amount of tampering attacks occurring on contents of attacked English context, which is denoted by ea1WDR and calculated by Eq. (7).

$$ea1\,WDR=100-ea1PMR \tag{7}$$

The steps involved in the watermark detection algorithm are illustrated in algorithm Algorithm 4.

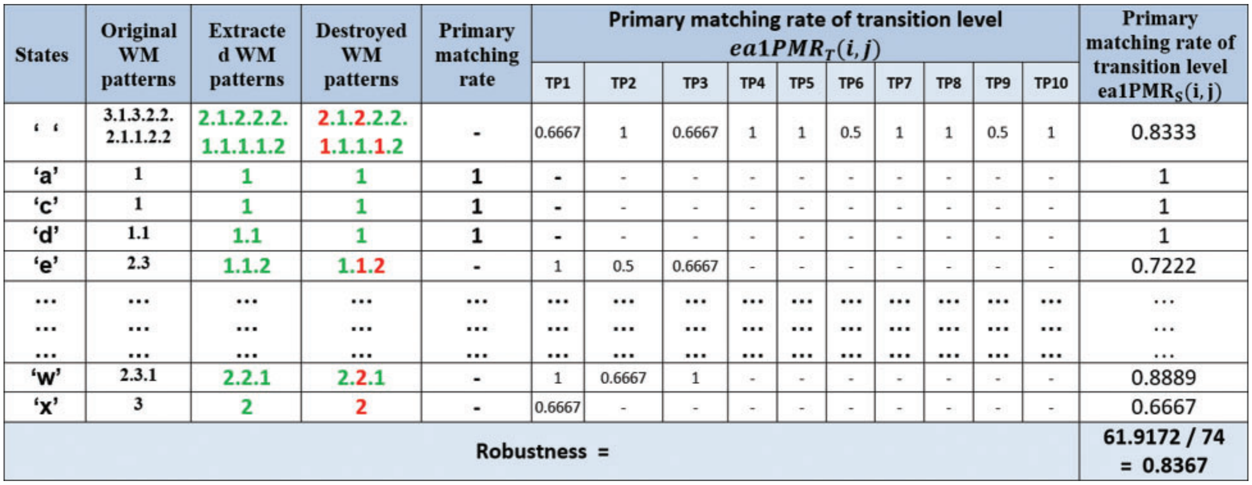
Algorithm 4: Watermark detection algorithm of SFASCDW



```
Procedure ea1_wm_detection(ea1WMP_O, ea1EWM_A)
Input: pre-processed attacked English text (ea1WMP_O, ea1EWM_A)
Output: ea1SW, ea1PMR, ea1WDR

BEGIN
ea1_wm_generation(ea1WMP_O)
ea1_wm_extraction (EWM_A)
// perform matching process between the original and attacked watermark patterns
 IF ea1EWM_A = ea1WMP_O
      Print “English text is an authentic”
      ea1PMR = 100
 ELSE
      Print “English text is not authentic and tampering occurred”
// compute pattern matching rate on transition level
      for i = 1 to ea1mm[i][j].length - 1,
           for j = 1 to ea1mm[i][j].length
                IF ea1WMP_O[i][j] != 0
                     pattern Count  +=1
                     $ea1PMR_T(i,j) = \left|\frac{ea1WMP_O[i][j] - (ea1WMP_O[i][j] - ea1EWM_A[i][j])}{ea1WMP_O[i][j]}\right|$
                     transPMRtotal += ea1PMR_T
                ELSE
                       IF ea1EWM_A[i][j] != 0
                            patternCount += ea1EWM_A[i][j]
      // compute pattern matching rate on state level
      $ea1PMR_S(i) = \left|\frac{\sum_{j=1}^{n-1}(ea1PMR_T(i,j))}{Total\ StatePatternCount(i)}\right|$
      $sWeight = \frac{ea1PMR_S(i) * Transitions\ frequency(i)}{total\ no\ of\ transitions}$
ea1SW += sWeight
// compute pattern matching rate on a whole a given text
$ea1PMR = \frac{\sum_{i=1}^{n-1}(ea1SW) * Total\ number\ of\ transitions}{Total\ number\ of\ transitions} * 100$
// compute watermark distortion rate on a whole a given text
ea1WDR =  1 – ea1PMR * 100
return ea1SW, ea1PMR, ea1WDR
```

where ea1_SW: refers to the watermark robustness rate. ea1PMR: refers to weight value of states correctly matched. ea1WDR: refers to value of watermark distortion rate (0 < ea1WDR <= 100).



States	Original WM patterns	Extracted WM patterns	Destroyed WM patterns	Primary matching rate	Primary matching rate of transition level $ea1PMR_T(i,j)$										Primary matching rate of transition level $ea1PMR_S(i,j)$
					TP1	TP2	TP3	TP4	TP5	TP6	TP7	TP8	TP9	TP10	
‘ ‘	3.1.3.2.2. 2.1.1.2.2	2.1.2.2.2. 1.1.1.1.2	2.1.2.2.2. 1.1.1.1.2	-	0.6667	1	0.6667	1	1	0.5	1	1	0.5	1	0.8333
‘a’	1	1	1	1	-	-	-	-	-	-	-	-	-	-	1
‘c’	1	1	1	1	-	-	-	-	-	-	-	-	-	-	1
‘d’	1.1	1.1	1	1	-	-	-	-	-	-	-	-	-	-	1
‘e’	2.3	1.1.2	1.1.2	-	1	0.5	0.6667	-	-	-	-	-	-	-	0.7222
… … …	… … …	… … …	… … …	… … …	… … …	… … …	… … …	… … …	… … …	… … …	… … …	… … …	… … …	… … …	… … …
‘w’	2.3.1	2.2.1	2.2.1	-	1	0.6667	1	-	-	-	-	-	-	-	0.8889
‘x’	3	2	2	-	0.6667	-	-	-	-	-	-	-	-	-	0.6667
Robustness =															61.9172 / 74 = 0.8367

Figure 6: Sample of watermark extraction and detection process using SFASCDW

The results of the watermark extraction and detection process with watermark robustness rate of the sample of the given English text after multiple random attacks performed are illustrated in Fig. 6.

4 Implementation and Simulation

To evaluate the robustness of SFASCDW, several scenarios of simulation and experiments are performed. This section depicts an implementation, simulation, and experimental environment, experiment parameters, experimental scenarios of standard English datasets, and results discussion.

4.1 Simulation and Implementation Environment

The self-developed program has been developed to test and evaluate the watermark robustness of SFASCDW. The implementation environment of SFASCDW is: CPU: Intel Core i7-4650U/ 2.3 GHz, RAM: 8.0 GB, Windows 10–64 bit, PHP programming language with VS Code IDE.

4.2 SFASCDW Simulation and Experiment

This subsection presents the robustness evaluation of SFASCDW. Many simulation and experiments scenarios are performed as shown in Tab. 1, for all forms of attacks and their volumes.

Table 1: Robustness evaluation of SFASCDW under all attacks with various volumes

Attack volume (%)	Insertion	Deletion	Reorder
5	89.97	91.20	89.21
10	85.70	92.55	82.48
20	69.58	83.37	72.95
s50	49.97	60.91	62.96

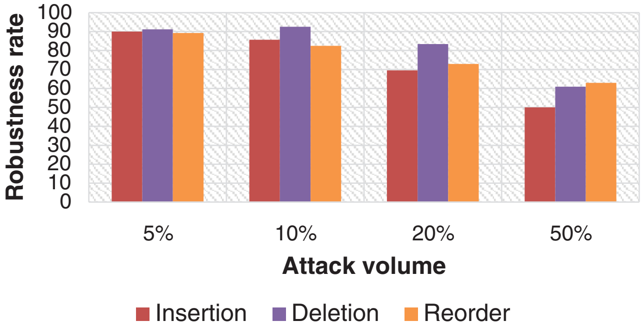


Figure 7: Robustness effect under all attacks with various volumes

From Tab. 1 above and Fig. 7 below, it appears that the SFASCDW approach gives sensitive results of robustness under all attacks in which the structure, syntax, and semantics of English text contents may be affected. As a comparison of robustness based on attack types, results show that the best robustness rate was detected in descending order by deletion, reorder, and reorder attack sequentially in all scenarios of attack volumes.

5 Comparison and Result Discussion

The robustness results were critically analyzed. This subsection displays an effect study and a comparison between SFASCDW and baseline approaches Hybrid of Natural Language Processing and Zero-Watermarking Approach (HNLPZWA) [5] and Zero-Watermarking Approach based on Fourth level order of Arabic Word Mechanism of Markov Model (ZWAFWMMM) [6]. It also contains a discussion of their effect under the major factors, namely dataset size, attack types, and volumes.

5.1 Robustness Comparison Under Attack Type Effect

Tab. 2 shows a comparison of the different attack types' effect on robustness of SFASCDW, ZWAFWMMM, and HNLPZWA approaches against all dataset sizes and all scenarios of attack volumes.

Table 2: Attack types effect on robustness of SFASCDW, ZWAFWMMM, and HNLPZWA approaches

Method	HNLPZWA	ZWAFWMMM	SFASCDW
Insertion	74.28	80.02	83.81
Deletion	59.99	69.35	82.60
Reorder	37.23	44.88	77.19

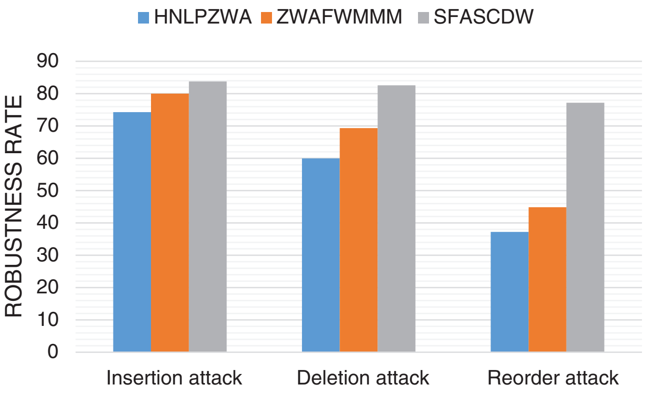


Figure 8: A compression of attack type effect on robustness of SFASCDW, ZWAFWMMM, and HNLPZWA approaches

Tab. 2 and Fig. 8 show how the robustness of SFASCDW, ZWAFWMMM, and HNLPZWA approaches is influenced by the type of tampering attack. In all cases of insertion, deletion, and reorder attacks, low effect has been detected between robustness of SFASCDW approach and baseline approaches ZWAFWMMM as well as HNLPZWA in terms of watermark robustness, which shows that the proposed SFASCDW approach is the best choice as seen as the case study on effect of attack type. This means that the proposed SFASCDW approach is strongly recommended and applicable for content authentication and tampering detection of English text documents for all types of attack.

5.2 *Robustness Comparison Under Attack Volume Effect*

Tab. 3 provides a comparison of the different attack volumes' effect on robustness against all dataset sizes and all scenarios of attack volumes. The comparison is performed using SFASCDW, ZWAFWMMM, and HNLPZWA approaches.

Table 3: Attack volume effect on robustness of SFASCDW, ZWAFWMMM, and HNLPZWA approaches

Attack volume (%)	ZWAFWMMM	HNLPZWA	SFASCDW
5	82.09	83.60	92.35
10	72.74	74.33	87.62
20	57.71	59.39	76.33
50	13.66	37.56	60.12

Tab. 3 and Fig. 9 show how the robustness is influenced by low, mid, and high attack volumes. Fig. 9 demonstrates that if the attack volume increases, the robustness also increases. In all cases of low, mid, and high attack volumes, SFASCDW outperforms both ZWAFWMMM and HNLPZWA approaches in terms of robustness in low, mid, and high volumes of attacks. This means that SFASCDW approach is strongly recommended for content authentication and tampering detection of all types of English text.

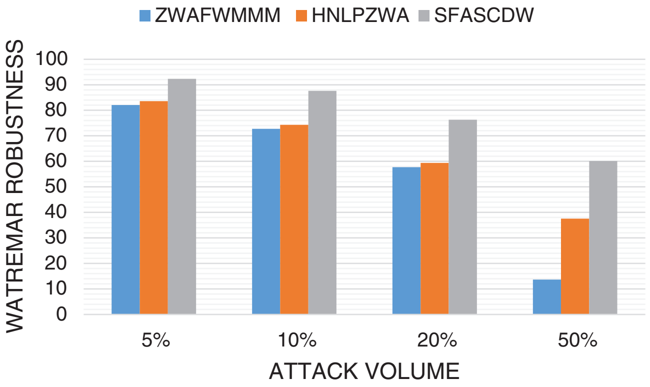


Figure 9: A compression of attack volume effect on robustness of SFASCDW, ZWAFWMMM, and HNLPZWA approaches

5.3 *Robustness Comparison Under Dataset Size Effect*

This section tests the various dataset size effect on robustness against all forms of attacks within their multiple volumes. Tab. 4 shows a comparison of that effect using SFASCDW, ZWAFWMMM, and HNLPZWA approaches.

Table 4: Dataset size effect on robustness of SFASCDW, ZWAFWMMM, and HNLPZWA approaches

Dataset size	ZWAFWMMM	HNLPZWA	SFASCDW
[ESST]	67.272	69.534	83.89
[EMST]	63.802	68.126	82.15
[EHMST]	59.233	65.108	81.02
[ELST]	54.466	62.073	78.19

The comparative results as shown in Tab. 4 and Fig. 10 reflect the robustness of the proposed SFASCDW approach. The results show that in the SFASCDW approach, the highest effects of dataset size that lead to the best robustness are ordered as ESST, EMST, EHMST, and ELST. This means that the robustness increases with the decreased document size and decreases with the increased document size. That said, the results show that SFASCDW approach outperforms both ZWAFWMMM and HNLPZWA approaches in term of robustness for all sizes of dataset.

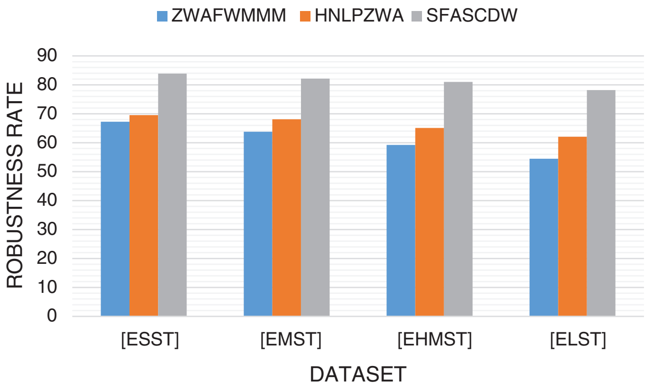


Figure 10: A compression of dataset size effect on robustness of SFASCDW, ZWAFWMMM, and HNLPZWA approaches

6 Conclusion

Based on the first-level order and alphanumeric mechanism of HMM, a smart fragile approach has been developed that is abbreviated as SFASCDW for content authentication and tampering detection of English text transmitted via the internet. SFASCDW uses a combination of the zero watermarking technique and soft computing techniques for text analysis to find inter-relationships between the contents of the given English text and the generated watermark key. The generated watermark is embedded logically in the original English context without modifications of and effect on the size of the original text. The embedded watermark is used later after the transmission of text via the internet to detect any tampering occurring on the received English text and verifies whether it is authentic or not. SFASCDW approach is implemented in PHP using VS code IDE. The simulation and experiments are performed on various standard datasets under different volumes of insertion, deletion, and reorder attacks. SFASCDW approach has been compared with ZWAFWMMM and HNLPZWA approaches. Comparison results show that

SFASCDW outperforms ZWAFWMMM and HNLPZWA in terms of general robustness because using the first-level order and alphanumeric mechanism of HMM leads to better performance and robustness in which the first order of interrelationships between alphanumeric is stronger than other levels order level of alphanumeric or words mechanisms of HMM. For future work, the authors intend to improve the robustness using other techniques and mechanisms.

Funding Statement: The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work under Grant Number (RGP.1/147/42), Received by Fahd N. Al-Wesabi. www.kku.edu.sa.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] F. N. Al-Wesabi, "A smart english text zero-watermarking approach based on third-level order and word mechanism of Markov model," *Computers, Materials & Continua*, vol. 65, no. 2, pp. 1137–1156, 2020.

[2] M. Abd-Eldayem, "A proposed security technique based on watermarking and encryption for digital imaging and communications in medicine," *Egyptian Informatics Journal*, vol. 14, no. 1, pp. 1–13, 2013.

[3] F. N. Al-Wesabi, "A hybrid intelligent approach for content authentication and tampering detection of Arabic text transmitted via Internet," *Computers, Materials & Continua*, vol. 66, no. 1, pp. 195–201, 2021.

[4] S. G. Rizzo, F. Bertini and D. Montesi, "Fine-grain watermarking for intellectual property protection," *EURASIP Journal on Information Security*, vol. 10, no. 1, pp. 804, 2019.

[5] F. N. Al-Wesabi, "Proposing high-smart approach for content authentication and tampering detection of arabic text transmitted via internet," *IEICE Transactions in Information Systems*, vol. E103, no. 10, pp. 2104–2112, 2020.

[6] F. N. Al-Wesabi, K. Mahmood and N. Nemri, "A zero watermarking approach for content authentication and tampering detection of arabic text based on fourth level order and word mechanism of Markov model," *Journal of Information Security and Applications*, vol. 52, no. 1, pp. 1–15, 2020.

[7] P. Selvama, S. Balachandran, S. Pitchai and R. Jayabal, "Hybrid transform based reversible watermarking technique for medical images in telemedicine applications," *ELSEVIER Optik*, vol. 145, no. 5, pp. 655–671, 2017.

[8] N. Hurrah, A. Parah, N. Loan, A. Sheikh, M. Elhoseny *et al.*, "Dual watermarking framework for privacy protection and content authentication of multimedia," *ELSEVIER Future Generation Computer Systems*, vol. 94, pp. 654–673, 2019.

[9] A. Panah, R. Van, T. Sellis and E. Bertino, "On the properties of non-media digital watermarking: A review of state-of-the-art techniques," *IEEE Access*, vol. 4, pp. 2670–2704, 2016.

[10] C. Qin, C. Chang and T. Hsu, "Fragile watermarking for image authentication with high-quality recovery capability," *KSII Transactions on Internet and Information Systems*, vol. 7, no. 11, pp. 2941–2956, 2013.

[11] S. Parah, J. Sheikh and G. Bhat, *StegNmark: A Joint Stego-Watermark Approach for Early Tamper Detection*, vol. 660. Switzerland: Springer International Publishing, pp. 427–452, 2017.

[12] S. Hakak, A. Kamsin, O. Tayan, M. Yamani and G. Gilkar, "Approaches for preserving content integrity of sensitive online Arabic content," *Information Processing and Management*, vol. 56, no. 2, pp. 367–380, 2019.

[13] M. Taleby, Q. Li, X. Zhu, M. Alazab and J. Zhang, "A Novel intelligent text watermarking technique for forensic identification of information on social media," *Computers and Security*, vol. 90, pp. 1–14, 2020.

[14] S. Parah, J. Sheikh, J. Akhoon and N. Loan, "Electronic health record hiding in images for smart city applications: A computationally efficient and reversible information hiding technique for secure communication," *ELSEVIER Future Generation Computer Systems*, vol. 108, no. 6, pp. 935–949, 2020.

[15] R. Ahmed and L. Elrefaei, "Arabic text watermarking: A review," *International Journal of Artificial Intelligence & Applications*, vol. 6, no. 4, pp. 1–16, 2015.

[16] K. Hameed, A. Khan, M. Ahmed and A. G. Reddy, "Towards a formally verified zero watermarking scheme for data integrity in the internet of things based-wireless sensor networks," *ELSEVIER Future Generation Computer Systems*, vol. 167, pp. 1–16, 2018.

[17] R. Alotaibi and L. Elrefaei, "Improved capacity text watermarking methods based on open word space," *Journal of King Saud University–Computer and Information Sciences*, vol. 30, no. 2, pp. 236–248, 2018.

[18] M. Memon and A. Shah, "A novel text steganography technique to Arabic language using reverse fat5th5ta," *Pakistan Journal of Engineering, Technology and Sciences*, vol. 1, no. 2, pp. 106–113, 2015.

[19] Y. Alginahi, M. Kabir and O. Tayan, "An enhanced Kashida-based watermarking approach for increased protection in arabic text-documents based on frequency recurrence of characters," *International Journal of Computer and Electrical Engineering*, vol. 6, no. 5, pp. 381–392, 2014.

[20] A. Shaker, F. Ridzuan and S. Pitchay, "Text steganography using extensions Kashida based on moon and sun letters," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 8, pp. 286–290, 2017.

[21] A. Rahma, W. Bhaya and D. Al-Nasrawi, "Text steganography based on unicode of characters in multilingual," *Journal of Engineering Research and Applications*, vol. 3, no. 4, pp. 1153–1165, 2013.

[22] N. Al-maweri, W. Adnan, A. Rahman, S. Khair and S. Syed, "Robust digital text watermarking algorithm based on unicode characters," *Indian Journal of Science and Technology*, vol. 9, no. 48, pp. 1–14, 2016.

[23] M. Bashardoost, M. Rahim, T. Saba and A. Rehman, "Replacement attack: A new zero text watermarking attack," *3D Research*, vol. 8, no. 1, 2017. https://doi.org/10.1007/s13319-017-0118-y.

[24] Y. Liu, Y. Zhu and G. Xin, ""A zero-watermarking algorithm based on merging features of sentences for chinese text," *Journal of the Chinese Institute of Engineers*, vol. 38, no. 3, pp. 391–398, 2015.

[25] P. Zhu, W. Song, A. Li, Y. Zhang and R. Tao, "A text zero watermarking algorithm based on chinese phonetic alphabets," *Wuhan University Journal of Natural Sciences*, vol. 21, no. 4, pp. 277–282, 2016.

[26] Z. Ali, M. Shamim, G. Muhammad and M. Aslam, "New zero-watermarking algorithm using hurst exponent for protection of privacy in telemedicine," *IEEE Access*, vol. 6, pp. 7930–7940, 2018.

[27] O. Tayan, Y. Alginahi and M. Kabir, "An adaptive zero-watermarking approach for text documents protection," *International Journal of Image Processing Techniques*, vol. 1, no. 1, pp. 33–36, 2014.

[28] M. Ghilan, F. Ba-Alwi and F. N. Al-Wesabi, "Combined Markov model and zero watermarking to enhance authentication of Arabic text," *Journal of Computational Linguistics Research*, vol. 5, no. 1, pp. 26–42, 2014.

[29] F. N. Al-Wesabi, A. Alsakaf and K. U. Vasantrao, "A zero text watermarking algorithm based on the probabilistic patterns for content authentication of text documents," *International Journal of Computer Engineering & Technology*, vol. 4, no. 1, pp. 284–300, 2013.

[30] H. Ahmed and M. Khodher, "Comparison of eight proposed security methods using linguistic steganography text," *Journal of Computing & Information Sciences*, vol. 12, no. 2, pp. 243–251, 2016.