Computers, Materials & Continua
DOI:10.32604/cmc.2022.019594
Article

Tech Science Press

Securing Arabic Contents Algorithm for Smart Detecting of Illegal Tampering Attacks

Mesfer Al Duhayyim[1], Manal Abdullah Alohali[2], Fahd N. Al-Wesabi[3,4], Anwer Mustafa Hilal[5,*], Mohammad Medani[3] and Manar Ahmed Hamza[5]

[1]Department of Natural and Applied Sciences, College of Community - Aflaj, Prince Sattam bin Abdulaziz University, Saudi Arabia
[2]Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Saudi Arabia
[3]Department of Computer Science, King Khalid University, Muhayel Aseer, Saudi Arabia
[4]Faculty of Computer and IT, Sana'a University, Sana'a, Yemen
[5]Department of Computer and Self Development, Preparatory Year Deanship, Prince Sattam bin Abdulaziz University, AlKharj, Saudi Arabia
[*]Corresponding Author: Anwer Mustafa Hilal. Email: a.hilal@psau.edu.sa
Received: 19 April 2021; Accepted: 19 June 2021

Abstract: The most common digital media exchanged via the Internet is in text form. The Arabic language is considered one of the most sensitive languages of content modification due to the presence of diacritics that can cause a change in the meaning. In this paper, an intelligent scheme is proposed for improving the reliability and security of the text exchanged via the Internet. The core mechanism of the proposed scheme depends on integrating the hidden Markov model and zero text watermarking techniques. The watermark key will be generated by utilizing the extracted features of the text analysis process using the third order and word level of the Markov model. The Embedding and detection processes of the proposed scheme will be performed logically without the effect of the original text. The proposed scheme is implemented using PHP with VS code IDE. The simulation results, using varying sizes of standard datasets, show that the proposed scheme can obtain high reliability and provide better accuracy of the common illegal tampering attacks. Comparison results with other baseline techniques show the added value of the proposed scheme.

Keywords: Tampering detection; zero-watermarking; soft computing; text analysis; hidden Markov model

1 Introduction

In communication technologies, validation and authentication of digital text and detection of its reliability in various languages become the most challenging to secure the transmission of the contents over the Internet. Many applications, e.g., e-Banking, eLearning, eGovernment and e-Commerce, render information exchange via the Internet the most difficult.

Most of the digital media exchanged via the internet is in text form. The text data is very easy for illegal attacks to modify, and more sensitive than other media for changes whenever change volume was very low. The Arabic language is considered one of the most sensitive languages in terms of content modification due to the presence of diacritics that can cause a change in the meaning [1].

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]. Another technique, A fine-grained text watermarking procedure, is proposed based on replacing the white spaces and Latin symbols with homoglyph characters [4].

Many solutions and methods are proposed by researchers to improve the security of digital media for several purposes such as copyright protection, identity verification, and authentication [5]. Solutions for text media are limited because text data is natural language-dependent and there are no locations to hide secret information within the text. In the case of images, secure data can be hidden in pixels. However, in the case of audio, secure data can be hidden in waves, and in the case of video, it can be hidden in frames [6,7].

Several information security techniques and algorithms can be used for various purposes of content authentication and tampering detection of digital text [8]. To hide the security data in the original document, some modifications and changes are required to be performed in the original document, and can affect in terms of contents, meaning, and document size [9,10]. Text watermarking is the most appropriate technique used for this purpose. Zero text watermarking is a smart algorithm that can be used, without any changes or modifications to the original text document, to embed the watermark data. Moreover, this technique can be used to generate data for a watermark in the contents of the given text [11–13].

The verification of digital text and the identification of fraud in research earned great attention [14]. Restricted appropriate techniques and solutions are proposed to improve the authenticity and reliability of sensitive digital multimedia such as Arabic interactive Holy Qur'an and digital money [15]. The most common technique of soft computation and Natural Language Processing (NLP) that supported the analysis of the text is HMM [16].

In this paper, we propose a combined smart scheme, Smart Arabic text Watermarking scheme based on third of Word-level of Markov Model (SAWMWMM) for detecting illegal tampering of Arabic text transmitted via the Internet. The proposed scheme depends on the integration of zero text watermarking and the hidden Markov model used as an NLP technique for text analysis and extraction of Arabic text features to use it as a watermark key.

The primary objective of the SAWMWMM scheme is to meet the high accuracy and reliability of sensitive detection of illegal tampering attacks on Arabic text which is transmitted through the Internet.

The remainder of this paper is structured as follows: In Section 2, the authors review the existing works done so far and explain the suggested SAWMWMM scheme in Section 3. The simulation and implementation are provided in Section 4, the discussion of results in Section 5, and finally, the authors conclude the paper in Section 6.

2 Related Work

According to content verification, authentication, and tampering detection domain, several solutions and techniques are reviewed in this paper.

In [17], the location of the words-based method is proposed to improve the visibility and capability of the Arabic contents. The mechanism of this method is based on word space to indicate the Boolean bit whether 1 or 0 which refers to the modifications of the plain text.

A steganography-based algorithm is designed in [18] to hide secure data in the Arabic text. In this algorithm, Harakat's of the Arabic diacritics such as Damma, Fatha, and Kasra are utilized to cover messages of secure data and use them later to validate the authenticity of the text.

In [19], A Kashida-based scheme is suggested for authentication of text documents. This scheme depends on the frequent recurrence of characters and utilizes its features as watermark data with a Kashida bit 0 and/or a bit omitted. In [20], a Kashida extension-based method is developed for Arabic content authentication. The mechanism of this method is based on using 'moon' and 'sun' characters of Arabic language to organize the given context. In addition, characters of the Kashida method are used to validate the authenticity of hidden bits.

In [21], a scheme for Quran text watermarking is proposed to improve the security of Arabic text. The proposed scheme uses vowel characters with kashida. This scheme is based on a reversing technique that depends on four phases, namely, pre-processing, embedding process, extraction process, and evaluate the performance of the proposed scheme. The results show good improvement of capacity.

A text watermarking scheme is proposed in [22] to improve the security of Arabic text in the Holy Quran by using vowels with kashida. The hybrid technique used in the proposed scheme are XOR and queuing techniques.

A position-based algorithm is designed in [23] for content authentication purposes. This algorithm depends on conserving the words positions of the given text, then manipulate their transitions and utilize them as a secret key. In [24,25], Chinese text-features methods are proposed for authenticity validation of Chines text. The mechanism of these methods depends on splitting the Chinese text into small groups of sentences, obtaining the semantic code of each word, and utilizing the sentence entropy and distribution of the semantic codes as secret data which will be used later to validate the given Chines text as authentic or not.

In [26], A Hurst exponent-based watermarking method is proposed to improve the user's privacy. This method depends on an individual's identity and utilizes it as a watermark key which will be used later to evaluate the unvoiced frames. An English text-based watermarking method is presented in [27] to improve the security issues of English text. Markov-based methods are proposed in [28,29] to validate the authenticity of the English text contents. These methods depend on utilizing the probability feature of the given text as a secret watermark key. The ASCII-based method is suggested in [30] for copyright protection of English text. This method uses ASCII of non-vowel letters of the English text.

3 The Proposed Scheme (SAWMWMM)

In this paper, we propose a smart scheme called Smart Arabic text Watermarking based on third of Word-level of Markov Model (SAWMWMM). The proposed scheme depends on combining text watermark and NLP techniques to get rid of external watermark data without altering the source text to hide the watermark data. The third of word-level of the Markov model is used as

a NLP technique for text analysis of the given Arabic text, extract their interrelationships features and utilizes them as a watermark key.

The main contributions of the proposed SAWMWMM scheme can be summarized as follows:

- The accuracy of tampering detection has improved as compared with other previous approaches.
- Tampered locations will be determined in the given Arabic text.
- Unlike previous work where watermarking is done with language, contents, and scale effecting, the SAWMWMM scheme logically embeds watermarking with no effect on text, content, or size.
- The SAWMWMM scheme is highly vulnerable to any basic alteration to the Arabic 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 for this paper's contribution.

Two core phases are conducted to run the SAWMWMM scheme, firstly the watermark generating and embedding, and secondly the watermark extraction and detection as shown in Figs. 1 and 2.

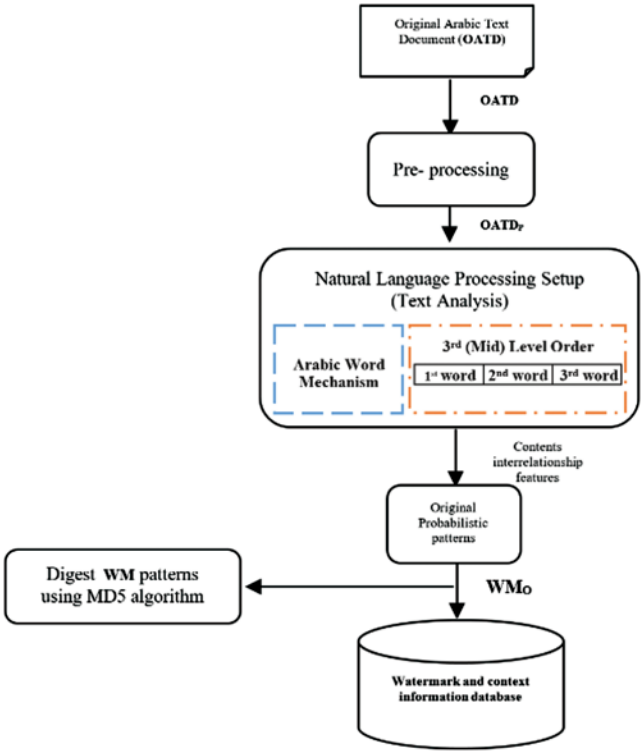

Figure 1: Core processes of the first phase of the SAWMWMM scheme

3.1 The Generating and Embedding of the Watermark

This phase consists of three sub-processes, namely, pre-processing, watermark generation, and embedding.

3.1.1 Pre-Processing and Construction of the Markov Matrix Process

The core aim of this process is to remove the extra new lines and spaces of the original given text and then create the initial matrix of the Markov model. The Original Arabic Text (OAT) is required to perform this process. The core mechanism of this process depends on states and transitions probability by dividing the given text into small series of unique triple words to create a two-dimensional matrix of the Markov model. States in the Markov matrix are represented by a unique triple of words, whereas transitions are represented by a unique of words of the given text. This process executes as presented in Algorithm 1.

Algorithm 1: Pre-processing and creating the Markov matrix of the SAWMWMM scheme

```
PROCEDURE Prep_Building_MM (OAT)

- Input: original Arabic text (OAT)
- Output:  Markov matrix with zeros initial value

1.   BEGIN
2.   // perform pre-processing process
3.   for each word in OAT
4.        PAT ← trim ("space" or "newLine")
5.   // Build list of non values text words
6.   w2_mm = { }
7.   for each word in PAT
8.        if word not in w3_list
9.           w3_mm ← w3_mm U {word}
10.       for ps = 1 to w3_mm.length – 3
11.            for ns = 1 to w3_mm.length
12.                 w3_mm[ps][ns] = 0
13.  return w3_mm
```

where *OAT*: refers to source Arabic text, *PAT*: refers to preprocessed text, *w*3_*mm*: refers to Markov matrix, *ps*: is current state, *ns*: is next state.

The matrix size of *w*3_*mm*[*i*][*j*] of SAWMWMM is dynamic where it is based on the contents of the given text which is equal to the number of the triple of words without reputations.

3.1.2 Watermark Generation Process

This process represents the important task of the proposed SAWMWMM scheme. PAT is provided to this process as input to perform text analysis and extract the features of the given Arabic text by obtaining the interrelationships between them. The extracted features will be utilized as a watermark key. In this paper, we have used the following sample of Arabic text as an example to describe the various scenarios of the proposed SAWMWMM scheme.

"يقفز الثعلب البني السريع فوق الثعلب البني البطيء للوصول إلى الثعلب البني الميت"

Fig. 3 illustrates the physical representation of the available states and transitions of the given Arabic text using the proposed SAWMWMM scheme.

The interrelationships between the contents of the given sample of the Arabic text are shown in Fig. 4 as a result of the text analysis process using the third of word-level of the Markov model. We assume that "يقفز الثعلب البني" represent the selected state, whereas the only one

transition available is "السريع". The same cases in other available states. Depending on the available states and transitions of the given text, a two-dimensional matrix should be constructed $w3\_ma[i][j]$.

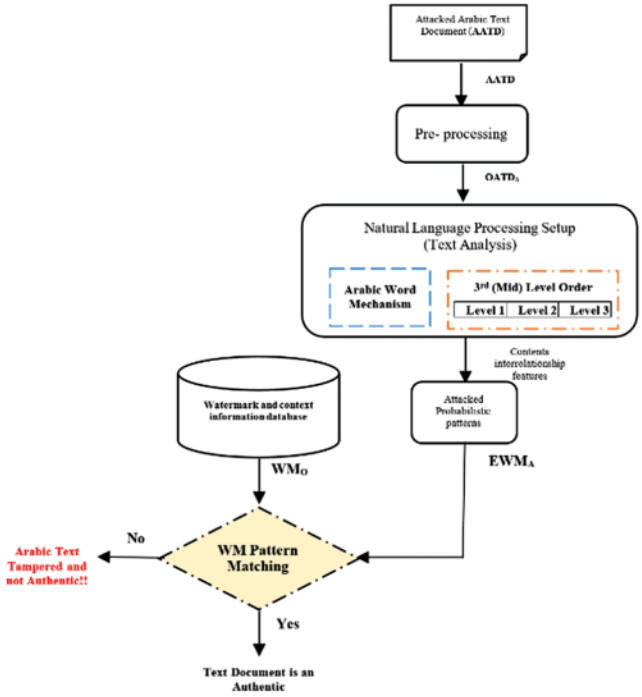

Figure 2: Core processes of the second phase of the SAWMWMM scheme

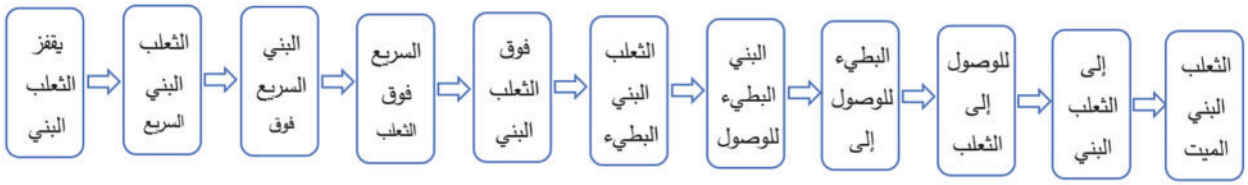

Figure 3: State's representation of Arabic text sample of the SAWMWMM scheme

In this process, the probability appearances of the possible next transition for every present state of the unique triple of words will be computed and accumulated as given by Eq. (1).

$$w3\_mm[ps][ns] = \sum_{i,j=1}^{n-3} transitions[i][j] \tag{1}$$

where n refers to the account of states.

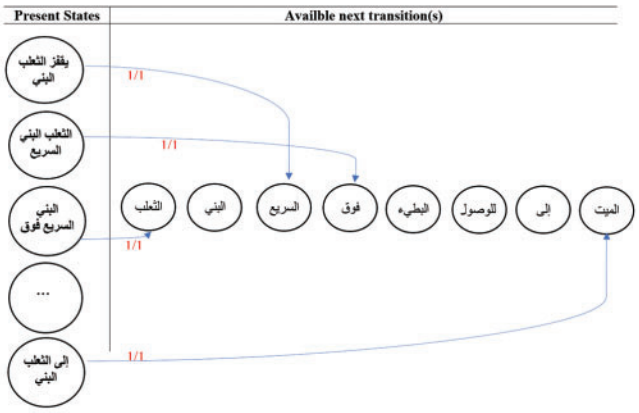

Figure 4: Representation of the inter-relationships between the given arabic text using the SAWMWMM scheme

Tab. 1 illustrates the results of the watermark generation process using the SAWMWMM scheme. Locations of the interrelationships between the Arabic contents will be saved in order to use them later in the detection process to determine the tampering locations.

Table 1: Arabic text feature extraction and watermark generation processes using the SAWMWMM scheme

States	Transitions									DWM Patterns
	يقفز	الثعلب	البني	السريع	فوق	البطيء	للوصول	إلى	الميت	
يقفز الثعلب البني	0	0	0	1	0	0	0	0	0	0,0,0,1/4,0,0,0,0,0
الثعلب البني السريع	0	0	0	0	1	0	0	0	0	0,0,0,0,1/5,0,0,0,0
البني السريع فوق	0	1	0	0	0	0	0	0	0	0,1/2,0,0,0,0,0,0,0
السريع فوق الثعلب	0	0	1	0	0	0	0	0	0	0,0,1/3,0,0,0,0,0,0
فوق الثعلب البني	0	0	0	0	0	1	0	0	0	0,0,0,0,0,1/6,0,0,0
الثعلب البني البطيء	0	0	0	0	0		1	0	0	0,0,0,0,0,0,1/7,0,0
البني البطيء للوصول	0	0	0	0	0	0	0	1	0	0,0,0,0,0,0,0,1/8,0
البطيء للوصول إلى	0	1	0	0	0	0	0	0	0	0,1/2,0,0,0,0,0,0,0
للوصول إلى الثعلب	0	0	1	0	0	0	0	0	0	0,0,1/3,0,0,0,0,0,0
إلى الثعلب البني	0	0	0	0	0	0	0	0	1	0,0,0,0,0,0,0,0,1/9

The processes of Arabic text feature extraction and watermark generation using the SAWMWMM scheme are presented formally in Algorithm 2.

Algorithm 2: Watermark generation using the SAWMWMM scheme

```
PROCEDURE ATA_WM_generation(PAT)

- Input: PAT, IMM
- Output: FM

1.  BEGIN
2.  Prep_Building_MM (PAT)
3.  pw = first_ triple_word(PAT)
4.  pd2 = PAT – [pw] // begin with 2nd triple of words
5.  fm = w3_mm
6.  for each w in pd2
7.      FM[pw][cw] = fm[pw][w] + 1
8.      pw = cw
9.  return FM
```

where *pw* is a previous word, and *cw* is a current word.

3.1.3 Watermark Embedding Process

The embedding process of the proposed SAWMWMM scheme will be performed logically by obtaining all non-zero values and their positions in the Markov matrix and concatenating them in sequential pattern W3_WMP$_O$ as given by Eq. (2), depicted in Fig. 5 and formally executed in Algorithm 3.

$$W3\_WMP_O \;\&= W3\_MM[ps][ns], \quad for\ i,j = non\text{-}zeros\ values\ resulted\ in\ W3\_MM \tag{2}$$

1,4 – 1,5 – 1,2 – 1,3 – 1,6 – 1,7 – 1,8 – 1,2– 1,3 – 1,9

Figure 5: The embedded watermark patterns W3_WMP$_O$ using the SAWMWMM scheme

Algorithm 3: Watermark embedding using the SAWMWMM scheme

```
PROCEDURE WM_embedding (PAT)

-  Input: pre-processed text (PAT)
-  Output: original watermark patterns

1.  BEGIN
2.  ATA_WM_generation (PAT)
3.  for ps = 1 to W3_arrList.Length - 3,
4.      for ns = 1 to W3_arrList.Length,
5.          if W3_MM [ps][ns] != 0
6.              W3_WMPO &= W3_MM [ps] [ns]
7.  W3_DWMPO = MD5(W3_WMPO)
8.  return W3_DWMPO, W3_WMPO
```

where $W3\_WMP_O$ refers to the original pattern of the embedded watermark.

3.2 The Extraction and Detection of the Watermark

This phase consists of two sub-processes, namely, watermark extraction and watermark detection.

3.2.1 Watermark Extraction Process

The core aim of this process is to extract the embedded watermark within the preprocessed text PAT and provide it to the detection process through $W3\_WMP_A$ as shown in Algorithm 4.

Algorithm 4: Watermark extraction using the SAWMWMM scheme

```
PROCEDURE WM_extraction(PATA)

- Input: pre-processed text (PATA)
- Output: attacked watermark patterns (WMPA).

1.   BEGIN
2.   ATA_WM_generation (PATA)
3.   for ps = 1 to W3_arrList'.Length - 3,
4.       for ns = 1 to W3_arrList'.Length,
5.           if W3_MM'[ps][ns] != 0,
6.              W3_WMPA &= W3_MM'[ps] [ns],
7. return W3_WMPA
```

where PAT_P is a pre-processed attacked Arabic text and $W3\_WMP_A$ is an attacked watermark pattern.

3.2.2 Watermark Detection Process

$W3\_WMP_A$ and $W3\_WMP_O$ are the core input needed to run the watermark detection process. However, the output of this process is to notify whether the Arabic text contents are tampered or not.

The detection process of the extracted watermark is performed at the transition level by computing the absolute value of the original and attacked watermark patterns as given by Eq. (3). However, detection at the state level is achieved by computing the total value of transition level divided by the total numbers of transitions as given by Eq. (4).

$$\mathrm{W3\_PMR_T}(i,j) = \left| \frac{W3_{WMPO}[i][j] - \left(W3_{WMPO}[i][j] - W3_{WMP_A[i][j]}\right)}{W3_{WMPO}[i][j]} \right| \tag{3}$$

where $\mathrm{W3\_PMR_T}$ is the value of patterns matching in the transition degree.

$$W3\_PMR_S(i) = \left| \frac{\sum_{j=1}^{n-1} (\mathrm{W3}\_PMR_T(i,j))}{Total\ \ State\ \ Pattern\ \ Count(i)} \right| \tag{4}$$

where $\mathrm{W3\_PMR_S}$ is the value of patterns matching in the state degree.

The matching weight of each state will be calculated by Eq. (5)

$$\text{W3\_}Sw = \left| \frac{\text{W3\_}PMR_S(i) * Transitions\ frequency(i)}{total\ number\ of\ transitions} \right| \tag{5}$$

where W3_PMR_S refers to the matching value of the i^{th} state.

The final $W3\_PMR$ of both AATD_P and OATD_P are computed by Eq. (6)

$$W3\_PMR = \left| \frac{\sum_{i=1}^{n-1} \text{W3\_}PMRS(i)}{N} \right| * 100 \tag{6}$$

The distortion value refers to the tampering rate of illegal attacks that occurred on the attacked Arabic text which is designated by W3_WDR and calculated by Eq. (7).

$$W3\_WDR = 100 - W3\_PMR \tag{7}$$

The steps involved in the watermark detection algorithm are as given in Algorithm 5.

Algorithm 5: Watermark detection using the SAWMWMM scheme

PROCEDURE WM_detection (W3_WMP_O, W3_WMP_A)

- Input: pre-processed text (W3_WMP_O, W3_WMP_A)
- Output: W3_PMR, W3_WDR

1. BEGIN
2. ATA_WM_generation (W3_WMP_O)
3. *WM_extraction* (WMP_A)
4. **IF** $\text{W3\_WMP}_A = \text{W3\_WMP}_O$
5. Print "Arabic document is authentic and no tampering occurred"
6. W3_PMR = 100
7. **Else**
8. Print "Arabic document is not authentic and tampering occurred"
9. **for** i = 1 **to** W3_arrList'.Length - 3,
10. **for** j = 1 **to** W3_arrList'.Length
11. **IF** $\text{W3\_WMP}_O[i][j]$!= 0
12. pattern Count +=1
13. $\text{W3\_}PMR_T(i,j) = \left| \frac{\text{W3\_}WMP_O[i][j] - (\text{W3\_}WMP_O[i][j] - \text{W3\_}WMP_A[i][j])}{\text{W3\_}WMP_O[i][j]} \right|$
14. transPMRTotal += W3_PMR_T
15. **Else IF** $\text{W3\_WMP}_A[i][j]$!= 0
16. patternCount += $\text{W3\_WMP}_A[i][j]$
17. $\text{W3\_}PMR_S(i) = \left| \frac{\sum_{j=1}^{n}(\text{W3\_}PMR_T(i,j))}{Total\ StatePatternCount(i)} \right|$
18. sWeight = $\frac{\text{W3\_}PMR_S(i) * Transitions\ frequency(i)}{total\ no\ of\ transitions}$
19. W3_SW += stateWeight
20. W3_PMR = $\frac{\sum_{i=1}^{n}(\text{W2\_}SW) * Total\ number\ of\ transitions}{Total\ number\ of\ transitions} * 100$
21. W3_WDR = 1 – W3_PMR * 100
22. **return** w3_PMR, w3_WDR

Tab. 2 illustrates the results of the watermark extraction and detection processes using the SAWMWMM scheme.

Table 2: Watermark extraction and detection processes using the SAWMWMM scheme

States	Original WM patterns	Extracted WM patterns	Destroyed WM patterns	Primary matching rate	Secondary matching rate of transition level $PMR_T(i,j)$			Primary matching rate of transition level $PMR_S(i,j)$
					TP1	TP2	TP3	
يقفز الثعلب البني	1/4	1/4	0	1	-	-	-	1
الثعلب البني السريع	1/5	1/2, 1/5	1/5	0	0	0.5	-	0.5
البني السريع فوق	1/2	1/3	1/2	0	-	-	-	0
السريع فوق الثعلب	1/3	1/3	0	1	0	-	-	1
فوق الثعلب البني	1/6	1/3, 1/6	1/3	0	0	0.5	-	0.5
الثعلب البني البطيء	1/7	1/7	1/7	1	-	-	-	1
البني البطيء للوصول	1/8	1/9	1/8	0	-	-	-	0
البطيء للوصول إلى	1/2	1/2, 1/5	1/5	0	0.5	0	-	0.5
للوصول إلى الثعلب	1/3	1/3, 1/5, 1/9	1/5, 1/9	0	0.33	0	0	0.33
إلى الثعلب البني	1/9	1/9	0	1	-	-	-	1
الثعلب الأسود السريع	-	1/6	1/6	0	-	-	-	0
بجانب الثعلب الميت	-	1/3, 1/7	1/3, 1/7	0	-	-	-	0
				(PMR) =				5.83 / 12 = 0.4858

4 Implementation and Simulation

To evaluate the reliability and accuracy of the SAWMWMM scheme, various scenarios of simulations are performed using a various size of standard datasets.

4.1 Implementation and Simulation Environment

The self-developed software is developed to evaluate the reliability and accuracy of the proposed SAWMWMM scheme. The implementation environment of SAWMWMM is: CPU: Intel Corei7 – 4650U/2.3 GHz, RAM: 8.0 GB, Windows 10–64 bit, PHP programming language with VS Code IDE.

4.2 SAWMWMM Simulation and Experiment

Many simulations and experiment scenarios are conducted to evaluate the reliability and accuracy of the SAWMWMM scheme using common types and volumes of tampering attacks (Insertion, Deletion and Reorder) as shown in Tab. 3.

Table 3: Evaluation of the detection accuracy of the SAWMWMM scheme

Attack volume (%)	Insertion	Deletion	Reorder
5	92.83	87.48	74.63
10	87.66	80.91	60.66
15	79.99	64.91	39.50
20	61.52	37.92	19.28

The results in Tab. 3 and Fig. 6 show that, in the case of low attack volume, the low effect is detected under insertion attack. However, in the case of medium and high attack volumes, a high effect is detected under deletion and reorder attacks. Therefore, in all scenarios of attack rates, the SAWMWMM scheme gives the best detection accuracy under deletion and reorder attacks.

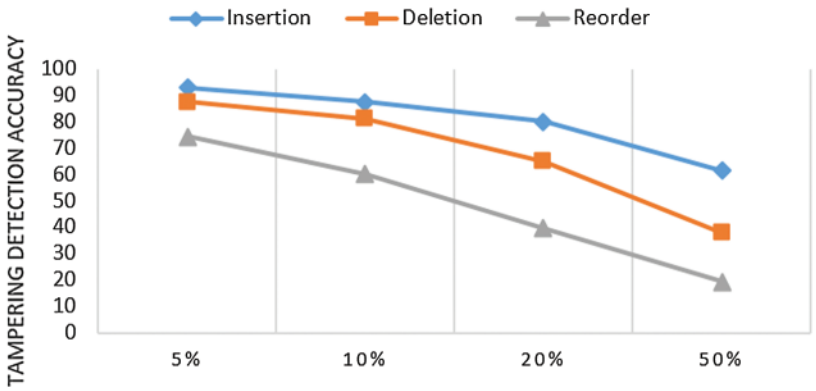

Figure 6: Accuracy evaluation of tampering detection of the SAWMWMM scheme

5 Comparison and Results Discussion

This section presents a detection accuracy effect study and comparison between SAWMWMM and other baseline methods, namely, UZWAMW [1] and HAZWCTW [3]. The results of detection accuracy under the effects of attack type, attack volume, and dataset size are critically evaluated and analysed.

5.1 Detection Accuracy Comparison Under Attack Type Effect

Tab. 4 shows a comparison effect of the different types of attack on the accuracy of SAWMWMM, UZWAMW, and HAZWCTW methods against various sizes of datasets and attack volumes.

Table 4: Effect of attack types on the detection accuracy of SAWMWMM, UZWAMW, and HAZWCTW

Attack type	HAZWCTW	UZWAMW	SAWMWMM
Insertion	74.28	80.02	80.50
Deletion	59.99	69.32	70.45
Reorder	37.23	44.88	48.36

As seen in Tab. 4 and Fig. 7, SAWMWMM outperforms UZWAMW and HAZWCTW in terms of general accuracy and reliability in all scenarios of all attacks. Hence, the SAWMWMM scheme is strongly recommended and applicable for detecting illegal tampering of all sensitive digital Arabic text.

5.2 Detection Accuracy Comparison Under Attack Volume Effect

Tab. 5 provides a critically comparison of the various attack volume effects on the detection accuracy against all sizes of the dataset and attack volumes.

As shown in Tab. 5 and Fig. 8, SAWMWMM outperforms both UZWAMW and HAZWCTW methods in terms of general performance in all volume scenarios of attack (low, mid, and high). Thus, the SAWMWMM scheme is applicable for detecting all illegal tampering

of sensitive Arabic text exchanged via the internet whenever the attack volume is very high or very low.

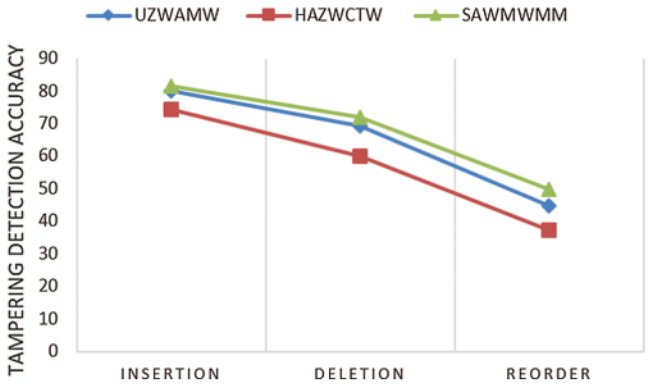

Figure 7: A comparison of attack type effect on the detection accuracy of SAWMWMM, UZWAMW and HAZWCTW methods

Table 5: Effect of attack volume on the detection accuracy of SAWMWMM, UZWAMW and HAZWCTW methods

Attack volume (%)	HAZWCTW	UZWAMW	SAWMWMM
5	82.09	83.60	84.98
10	72.74	74.33	76.21
20	57.71	59.39	61.46
50%	13.66	37.56	39.57

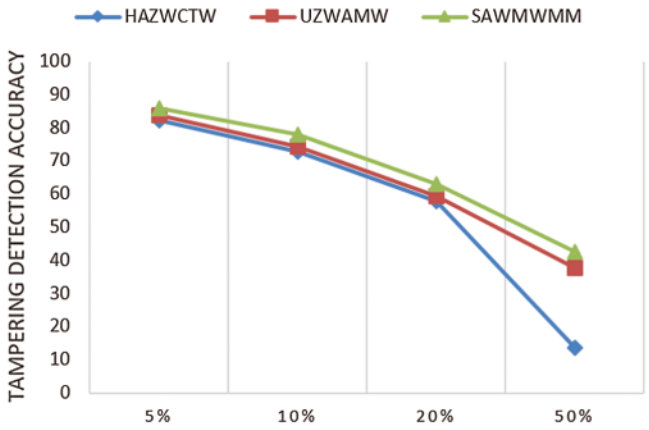

Figure 8: A comparison of attack volume effect on the detection accuracy of SAWMWMM, UZWAMW and HAZWCTW

5.3 Detection Accuracy Comparison Under Dataset Size Effect

Tab. 6 shows a comparison of different dataset size effects on the accuracy of tampering detection.

Table 6: Effect of dataset size on the detection accuracy of SAWMWMM, UZWAMW and HAZWCTW

Dataset size	HAZWCTW	UZWAMW	SAWMWMM
[ASST]	67.27	69.53	71.33
[AMST]	63.80	68.13	69.93
[AHMST]	59.23	65.11	66.90
[ALST]	54.47	62.07	63.94

The results of Tab. 6 and Fig. 9 show that in the SAWMWMM scheme, the highest effects detected of dataset size that reflect the best accuracy are ordered as ASST, ALST, AMST, and AHMST, respectively. Hence, the detection accuracy decreased with increasing document size and increased with decreasing document size. Nevertheless, the SAWMWMM scheme outperforms both UZWAMW and HAZWCTW approaches in terms of reliability and accuracy of tampering detection under all sizes of datasets.

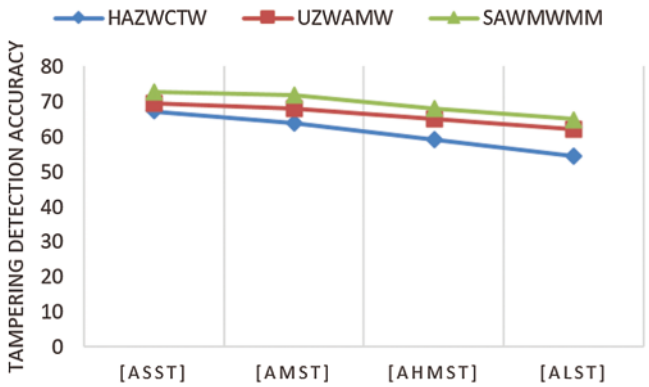

Figure 9: A comparison of dataset size effect on the detection accuracy of SAWMWMM, UZWAMW and HAZWCTW

6 Conclusion

In this paper, the SAWMWMM scheme is proposed to transfer and receive authentic Arabic content via the Internet. It depends mainly on combining zero watermarking and NLP techniques for text analysis to extract its features and utilize them as watermark data. The proposed scheme is implemented in PHP programming language using VS code IDE. It is simulated and evaluated using various sizes of standard datasets under different volumes of illegal tampering attacks, namely, insertion, deletion, and reorder. Evaluation results show that the SAWMWMM scheme is reliable, recommended, and applicable to use with e-Banking, eLearning, eGovernment and e-Commerce systems and applications to detect any tampering attacks of sensitive contents.

Compared with the UZWAMW and the HAZWCTW approaches, SAWMWMM outperforms them in terms of general performance which represents watermark capacity, reliability and detection accuracy under all scenarios of all attack types and volumes. For future work, the authors will intend to improve the reliability and accuracy using advanced techniques of soft computing.

Funding Statement: The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work under Grant Number (GRP/14/42), Received by Fahd N. Al-Wesabi. www.kku.edu.sa. This research was funded by the Deanship of Scientific Research at Princess Nourah bint Abdulrahman University through the Fast-track Research Funding Program.

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] A. Alkhafaji, N. Sjarif and M. Shahidan, "A review of comparative spatial domain techniques of steganography and watermarking," *Journal of Technology Reports of Kansai University*, vol. 62, no. 5, pp. 2291–2202, 2020.

[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. 195–2011, 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. 1–20, 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," *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," *Future Generation Computer Systems*, vol. 94, pp. 654–673, 2019.

[9] F. N. Al-Wesabi, H. G. Iskandar, S. Alzahrani, A. Abdelmaboud, M. Abdul *et al.,* "Arabic feature-based text watermarking technique for sensitive detecting tampering attack," *Computers Materials & Continua*, vol. 68, no. 3, pp. 3789–3806, 2021.

[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] F. N. Al-Wesabi, H. G. Iskandar, M. Alamgeer and M. Ghilan, "Proposing a high-robust approach for detecting the tampering attacks on english text transmitted via internet," *Intelligent Automation & Soft Computing*, vol. 26, no. 6, pp. 1267–1283, 2020.

[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] F. N. Al-Wesabi, "Entropy-based watermarking approach for sensitive tamper detection of arabic text," *Computers, Materials & Continua*, vol. 67, no. 3, pp. 3335–3648, 2021.

[15] F. N. Al-Wesabi, A. Abdelmaboud, A. Zain, M. Almazah and A. Zahary, "Tampering detection approach of arabic-text based on contents interrelationship," *Intelligent Automation & Soft Computing*, vol. 27, no. 2, pp. 483–498, 2021.

[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, no. 27, 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. Alkhafaji, N. Sjarif, M. Shahidan, N. Azmi, H. Sarkan *et al.,* "Payload capacity scheme for quran text watermarking based on vowels with Kashida," *Computers, Materials & Continua*, vol. 67, no. 3, pp. 3865–3885, 2021.

[22] A. Alkhafaji, N. Sjarif, M. Shahidan, N. Azmi, H. Sarkan *et al.,* "Tamper detection and localization for Quranic text watermarking scheme based on hybrid technique," *Computers, Materials & Continua*, vol. 68, no. 1, pp. 77–102, 2021.

[23] M. Bashardoost, M. Rahim, T. Saba and A. Rehman, "Replacement attack: A new zero text watermarking attack," *3D Research*, vol. 8, no. 1, pp. 1–9, 2017.

[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. 1–14, 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] F. N. Al-Wesabi, "Text analysis-based watermarking approach for tampering detection of english text," *Computers, Materials & Continua*, vol. 67, no. 3, pp. 3701–3719, 2021.

[29] F. N. Al-Wesabi, S. Alzahrani, F. Alyarimi, M. Abdul, N. Nemri *et al.,* "A reliable NLP scheme for english text watermarking based on contents interrelationship," *Computer Systems Science & Engineering*, vol. 37, no. 10, pp. 297–311, 2021.

[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.