Computers, Materials & Continua
DOI:10.32604/cmc.2021.017674
Article

Tech Science Press

Arabic Feature-Based Text Watermarking Technique for Sensitive Detecting Tampering Attack

Fahd N. Al-Wesabi[1,2,*], Huda G. Iskandar[2,3], Saleh Alzahrani[4], Abdelzahir Abdelmaboud[4], Mohammed Abdul[4], Nadhem Nemri[4], Mohammad Medani[4] and Mohammed Y. Alghamdi[5]

[1]Department of Computer Science, King Khalid University, Muhayel Aseer, Kingdom of Saudi Arabia
[2]Faculty of Computer and IT, Sana'a University, Sana'a, Yemen
[3]School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, Malaysia
[4]Department of Information Systems, King Khalid University, Muhayel Aseer, Kingdom of Saudi Arabia
[5]Department of Computer Science, Faculty of Science & Arts of Baljurshi, Al-Baha University, KSA
[*]Corresponding Author: Fahd N. Al-Wesabi. Email: fwesabi@gmail.com
Received: 06 February 2021; Accepted: 15 March 2021

Abstract: In this article, a high-sensitive approach for detecting tampering attacks on transmitted Arabic-text over the Internet (HFDATAI) is proposed by integrating digital watermarking and hidden Markov model as a strategy for soft computing. The HFDATAI solution technically integrates and senses the watermark without modifying the original text. The alphanumeric mechanism order in the first stage focused on the Markov model key secret is incorporated into an automated, null-watermarking approach to enhance the proposed approach's efficiency, accuracy, and intensity. The first-level order and alphanumeric Markov model technique have been used as a strategy for soft computing to analyze the text of the Arabic language. In addition, the features of the interrelationship among text contexts and characteristics of watermark information extraction that is used later validated for detecting any tampering of the Arabic-text attacked. The HFDATAI strategy was introduced based on PHP with included IDE of VS code. Experiments of four separate duration datasets in random sites illustrate the fragility, efficacy, and applicability of HFDATAI by using the three common tampering attacks i.e., insertion, reorder, and deletion. The HFDATAI was found to be effective, applicable, and very sensitive for detecting any possible tampering on Arabic text.

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 is a major issue in different systems that need critical 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.

3 The Proposed Approach

An intelligent approach is suggested in this paper by integrating a soft computing approach with zero-watermark that do not need additional details to be embedded as the key of a watermark and do not need to make any changes to the original text inserted into a watermark. The first level order of the alphanumeric mechanism of the Markov model was to be used as a soft computer approach to evaluate the Arabic text content and to build on the interrelationship characteristics of such text content.

The main contributions of our approach HFDATAI can be summarized as follows:

- Unlike previous work where watermarking is done with language, contents, and scale effecting, the HFDATAI approach logically embeds watermarking with no effect on text, content, or size.
- The watermarking mechanism does not require any external knowledge in our HFDATAI approach since this watermark key is generated by text processing and the extraction of a relationship between both the content and a watermark.
- The HFDATAI approach is highly vulnerable to any basic alteration to the Arabic text and context defined as complex text, namely Arabic symbols that can modify the meanings of the Arabic word. 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. In addition, our approach HFDATAI can effectively determine the place of tempering occurrence. This feature can be considered an advantage over the Hash function method.

The following sections describe in detail two major processes in HFDATAI. However, the first was the generation and incorporation of watermarks and the second was the detection and extraction process of watermarks.

3.1 Watermark Generation and Embedding Phase

Core sub-processes consist of pre-processing, watermark embedding, and watermark generation algorithms as well as text analysis as illustrated in Fig. 1.

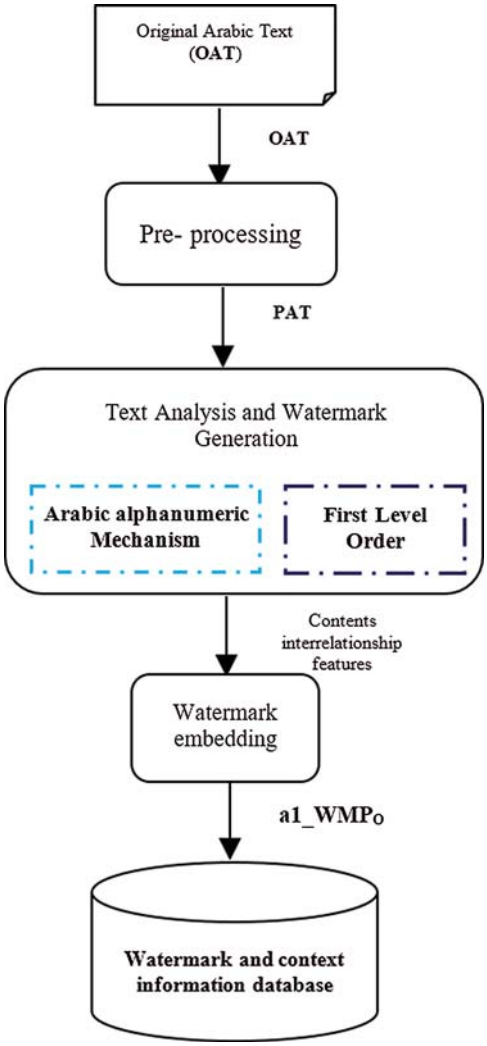

Figure 1: HFDATAI zero-watermark processes

3.1.1 Algorithm of Pre-Processing

Pre-processing the Arabic-text originality has been one of the main steps in creating and removing extra space and new lines and can have a significant effect on the precision of manipulation and watermark robustness. The original Arabic text (OAT) is necessary for the input process.

3.1.2 Algorithm of Watermark Generation

This algorithm involves the construction of the Markov matrix, the generation of watermarks, and the interpretation of text.

- *Developing Markov matrix* is a core step in developing the HFDATAI approach. A Markov chain matrix must be constructed in this process to setup the Markov model environment and represent all available states and transitions. In this approach, each unique single alphanumeric within a provided Arabic-text represents a current state, and each unique

single alphanumeric corresponds to a conversion in the matrix of Markov chain. When constructing the Markov chain matrix, zero values will be initialized for all states and transitions positions. Those positions will be uses later holding a record of the number of occurrences that the ith alphanumeric is then backed up through the jth alphanumeric and provided by Arabic-text.

The Markov matrix algorithm construction is performed as shown in Algorithm 1 below.

Algorithm 1: Algorithm of building Markov matrix using HFDATAI

PROCEDURE Prep_Building_MM (OAT)

```
1.  Input: original Arabic text (OAT)
2.  Output:  Markov matrix with zeros initial value
3.  BEGIN
4.  // perform pre-processing process
5.  for each word in OAT
6.          PAT ← trim ("space" or "newLine")
7.  // Build list of non values text alphanumerics
8.  a1_mm = { }
9.  for each alphanumeric in PAT
10.     if alphanumeric not in a1_list
11.        a1_mm ← a1_mm U { alphanumeric }
12.     for ps = 1 to a1_mm.length – 1
13.        for ns = 1 to a2_mm.length
14.           a1_mm[ps][ns] = 0
15. return a1_mm
```

where OAT: is the original Arabic text, PAT: is the preprocessed Arabic text, a1 mm: represent states and transformations matrix, ps: refers to the current state, ns: refers to the next state.

Watermark generation-based text analysis process: The proposed algorithm is performed as the second step of this process to perform Arabic text analysis and extract the features of the given text and produce watermark information. In this algorithm, there is a number of appearances of potential conversions for every present state of single alphanumeric will be computed as transition probabilities by Eq. (1).

$$a1\_mm[ps][ns] = \sum_{i,j=1}^{n-1} Total\ number\ of\ transitions\ [i][j] \tag{1}$$

where n: is the total number of states.

This example of the Arabic version demonstrates how this methodology was used to introduce the phase of transformation from the current state to the next state.

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

When you use the first stage order of the secret Markov-model alphanumeric approach, each special alphanumeric is a present state. Text processing is done as the text is read and the relationship meaning exchanged between the current and the next countries is calculated. The accessible transitions from the above sample of the Arabic text are shown in Fig. 2 below.

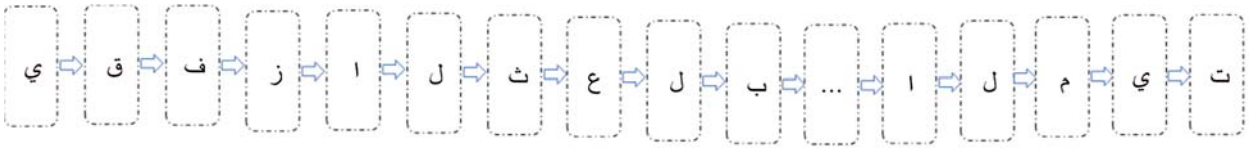

Figure 2: Arabic-text samples representation based HFDATAI

Fig. 3 illustrates the analysis results of the given Arabic sample and represents each state and its transitions based on the first scale order and alphanumeric approach of the Markov model. As shown in Fig. 3, we represent 22 unique present states of 78 possible transitions.

State ID	Present States		Next Transition(s)
1	(" ")	⟹	, [ا، ا، ا، ف، ا، ا، ا، ل، إ، ا، ا، ا]
2	("ي")	⟹	, [ق، ، ع، ، ء، ، ت]
3	("ق")	⟹	, [ف،]
4	("ف")	⟹	, [ز، و]
5	("ز")	⟹	, [ ]
...	.....	⟹	.....
...	.....	⟹	.....
19	("ا")	⟹	, [ل، ل، ل، ل، ل، ل، ل، ل، ل، ل، ل]
20	("ل")	⟹	, [ث، ب، س، ث، ب، ب، ب، ب، ي، ث، ب، م]
21	("م")	⟹	, [ي]
22	("ت")	⟹	, [.]

Figure 3: Arabic text sample transitions representation based HFDATAI

We assume that "ل" is a present state, and the available next transitions are "ب، س، ث، ب، ب، ب، ى، ث، ب، م ث،". We observe that eleven transitions are available in the sample of the given Arabic-text and "ب" transitions are repeated five times. The algorithm of watermark generation and text analysis depends on the first scale order of the alphanumeric approach as shown in Fig. 4.

States	Avalible transitions														Generated WM Patterns
	ا	ب	ت	ث	ز	س	ع	ف	ق	ل	م	و	ي		
	10	0	0	0	0	0	0	1	0	1	0	0	0	0	10.1.1
ي	1	0	1	0	0	0	1	0	0	0	0	0	0	3	1.1.1.3
ق	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1.1
ف	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1.1
ز	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
...	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
...	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ا	0	0	0	0	0	0	0	0	0	9	0	0	0	0	9
ل	0	5	0	3	0	1	0	0	0	0	1	0	1	0	5.3.1.1.1
م	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
ت	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Figure 4: Text analysis and watermark generation of the provided Arabic-text using HFDATAI

The algorithm of watermark generation and text analysis processes is formally introduced and performed as illustrated in Algorithm 2.

Algorithm 2: Watermark generation algorithm of HFDATAI

PROCEDURE ATA_WM_generation(PAT)

```
1.  Input: PAT, IMM
2.  Output: FM
3.  BEGIN
4.  Prep_Building_MM (PAT)
5.  pw = first_alpha(PAT)
6.  pd2 = PAT – [pa] // begin with 2nd alphanumeric
7.  fm = a2_mm
8.  for each a in pd2
9.      fm[pa][ca] = fm[pa][a] + 1
10.     pa = ca
11. return fm
```

3.1.3 Algorithm of Watermark Embedding

Watermark embedding has taken place logically in this method without needing to change the original text. The feature extraction of the given Arabic-text, watermark key is embedded logically by identifying all non-zero values in the Markov chain matrix. All these non-zero values are sequentially concatenated to form the original pattern of watermark key a1_WMP_O, as defined in Eq. (2) and Fig. 5.

$$a1\_WMP_O \& = a1\_mm[ps][ns], \quad for\ i, j = non\text{-}zeros\ values\ resulted\ in\ a1\_Markov\_matrix \tag{2}$$

10.1.1 – 1.1.1.1.3 – 1.1 – 1.1 – 9 – 5.3.1.1.1 - 1

Figure 5: The generated original pattern of watermark key a1_WMP_O using HFDATAI

The algorithm of the watermark embedding process using the HFDATAI approach is introduced formally and implemented as shown in Algorithm 3.

Algorithm 3: Algorithm of watermark embedding using HFDATAI

PROCEDURE WM_embedding (PAT)

```
1.  Input: pre-processed text (PAAT)
2.  Output: original watermark patterns
3.  BEGIN
4.  ATA_WM_generation (PAAT)
5.  for ps = 1 to a1_arrList.Length - 1,
6.      for ns = 1 to a1_arrList.Length,
7.          if a1_MM [ps][ns] != 0
8.              a1_WMPO &= a1_MM [ps] [ns]
9.  return a1_WMPO
```

3.2 Algorithms of Watermark Extracting and Detecting

This process consists of two key algorithms that are extracting and detecting the watermark. However, a1_EWM$_A$ is extracted from the obtained (PAAT) and matched by the detection algorithm with a1 WMP$_O$. PAAT is required as input to run this algorithm. Hence, it is necessary to perform the algorithm of watermark generation for obtaining the pattern of watermark for PAAT as presented in Fig. 6.

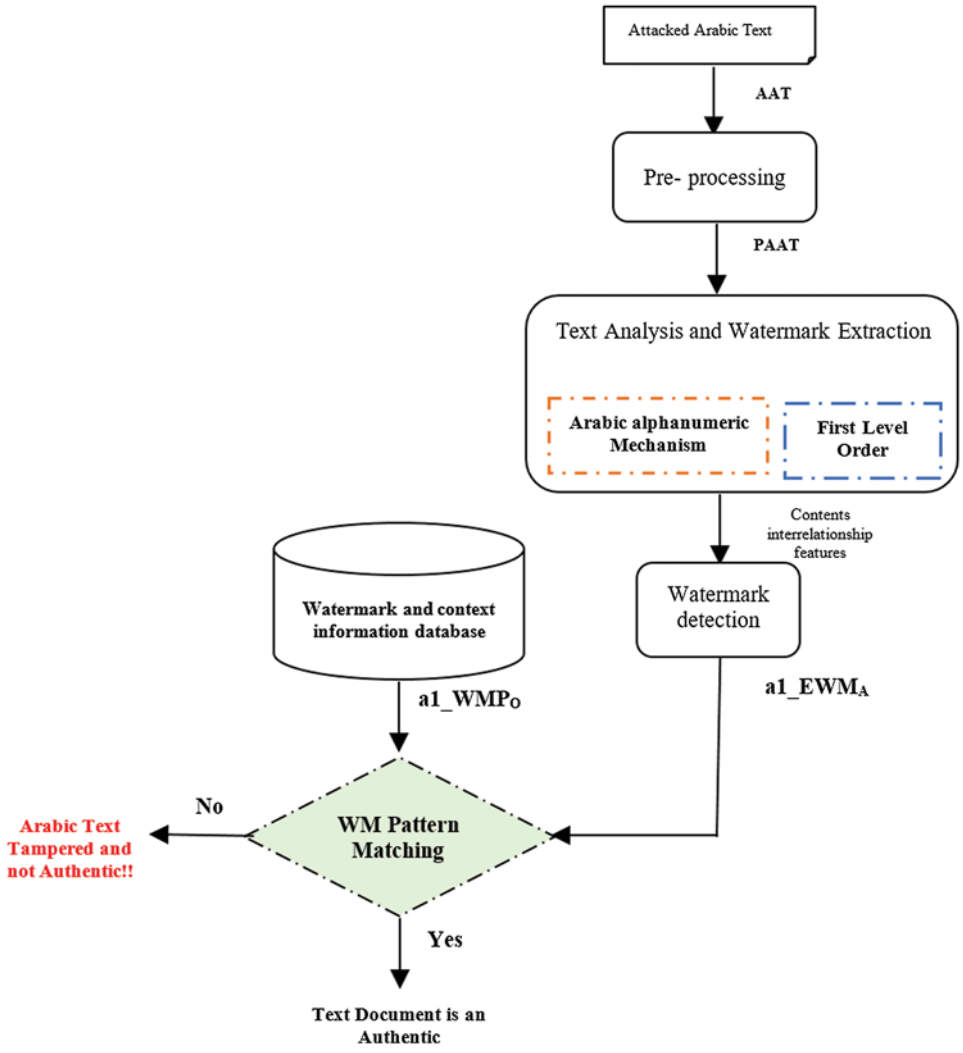

Figure 6: Zero-watermark HFDATAI processes of extraction and detection

3.2.1 Algorithm of Watermark Extraction

PAT should be provided as input to run this algorithm. Though, a1_WMPA is a core output of this algorithm as presented in Algorithm 4.

Algorithm 4: Algorithm of watermark extraction based HFDATAI

```
PROCEDURE WM_extraction(PAAT)

1.  Input: pre-processed text (PAAT)
2.  Output: attacked watermark patterns (a1_EWMA).
3.  BEGIN
4.  ATA_WM_generation (PAAT)
5.  for ps = 1 to a1_arrList'.Length - 1,
6.      for ns = 1 to a1_arrList'.Length,
7.          if a1_MM'[ps][ns] != 0,
8.              a1_EWMA &= a1_MM'[ps] [ns],
9.  return a1_EWMA
```

where, PAAT: pre-processed Arabic-text attacked, a1_EWM_A: attacked pattern of watermark key.

3.2.2 Algorithm of Watermark Detecting

$a1\_EWM_A$ and $a1\_WMP_O$ should be provided as the inputs needed for this algorithm to run. However, the status of the given Arabic-text is a core output of this algorithm which can be actual or tampered with. The watermark detection process is performed by two sub-steps which are:

- *The main matching* for $a1\_WMP_O$ and $a1\_EWM_A$ is achieved. If these two watermark patterns are similar in appearance, then there'll be a notification, "The Arabic text contents are authentic, and no tampering occurred." Likewise, the note will be rendered "This Arabic text document is tampered and not authentic," and then it continues to the next step.
- *The secondary matching* is performed by matching each state's transition status in the entire produced pattern of watermarks. This means $a1\_EWM_A$ of each state is contrasted with an analogous transition of $a1\_WMP_O$ as given by Eq. (3) and (4) below

$$a1\_PMR_T(i,j) = \left| \frac{a1\_WMP_O[i][j] - (a1\_WMP_O[i][j] - a1\_WMP_A[i][j])}{a1\_WMP_O[i][j]} \right|, \quad for\ all\ i,j\ states\ and\ transitions \tag{3}$$

where,

—$a1\_PMR_T$: represents tampering detection accuracy rate value in transition level, $(0 < a1\_PMR_T <= 1)$

$$a1\_PMR_S(i) = \left| \frac{\sum_{j=1}^{n-1} (a1\_PMR_T(i,j))}{total\ State\ Pattern\ Count(i)} \right| \quad for\ all\ i \tag{4}$$

where,

—$a1\_PMR_S$: value of tampering detection accuracy rate in state level, $(0 < a1\_PMR_S <= 100)$.

The weight of every state in the Markov matrix must be determined following the equivalent rate of every state, as seen in Eq. (5).

$$\left| a1\_Sw = \frac{a1\_PMR_S(i) * Transitions\ frequency(i)}{total\ number\ of\ transitions} \right| \tag{5}$$

where,

—a1_PMR$_S$: is the total matching value in the ith state level.

The ultimate a1_PMR of PAAT and PAT are computed by Eq. (6).

$$a1\_PMR = \left| \frac{\sum_{i=1}^{n-1} a1\_PMR_S(i)}{N} \right| \tag{6}$$

The distortion rate of the Watermark is the sum of manipulative attacks on the contents of the Arabic context that have been defined by a1 WDR and calculated by Eq. (7).

$$a1\_WDR = 1 - a1\_PMR * 100 \tag{7}$$

The algorithm of watermark detection is formally introduced and applied as seen in Algorithm 5.

Algorithm 5: Algorithm of watermark detection based HFDATAI

PROCEDURE WM_detection (a1_WMP$_O$, a1_EWM$_A$)

Input: pre-processed text (a1_WMP$_O$, a1_EWM$_A$)
Output: a1_PMR, a1_WDR
BEGIN
ATA_WM_generation (a1_WMP$_O$)
WM_extraction (EWM$_A$)
// perform matching process between the original and attacked watermark patterns
IF a1_EWM$_A$ = a1_WMP$_O$
    Print "Arabic document is authentic and no tampering occurred"
    a1_PMR = 100
ELSE
    Print "Arabic document is not authentic and tampering occurred"
// compute pattern matching rate on transition level
    for i = 1 **to** a1_arrList'.Length - 1,
        for j = 1 **to** a1_arrList'.Length
            IF a1_WMP$_O$[i][j] != 0
                pattern Count +=1
                $a1\_PMR_T(i,j) = \left| \frac{a1\_WMP_O[i][j] - (a1\_WMP_O[i][j] - a1\_EWM_A[i][j])}{a1\_WMP_O[i][j]} \right|$
                transPMRTotal += a1_PMR$_T$
            ELSE
                IF a1_EWM$_A$[i][j] != 0
                    patternCount += a1_EWM$_A$[i][j]
    // compute pattern matching rate on state level
    $a1\_PMR_S(i) = \left| \frac{\sum_{j=1}^{n}(a1\_PMR_T(i,j))}{Total\ StatePatternCount(i)} \right|$
    $sWeight = \frac{a1\_PMR_S(i) * Transitions\ frequency(i)}{total\ no\ of\ transitions}$
a1_SW += stateWeight
// compute pattern matching rate on a whole a given text
$a1\_PMR = \frac{\sum_{i=1}^{n}(a1\_SW) * Total\ number\ of\ transitions}{Total\ number\ of\ transitions} * 100$
// compute watermark distortion rate on a whole a given text
a1_WDR = 1 – a1_PMR * 100
return a1_PMR, a1_WDR

where a1_SW: refers to the weight value for properly matched states. a1_WDR: refers to the importance of the distortion rate of the watermark ($0 <$ a1_WDR$_S$ $<= 100$).

The effects of the method of watermark extraction and detection is illustrated in Fig. 7.

States	Original WM Patterns	Extracted WM Patterns	Destroyed WM Patterns	Primary Matching Rate	L1_PMRt(l,j) of Transition Level								L1_PMRs(l,j) of State Level
					TP1	TP2	TP3	TP4	TP5	TP6	TP7	TP8	
" "	10.1.1	7.2.1	7.2.1	-	0.7	0.5	1	-	-	-	-	-	**0.7333**
"ي"	1.1.1.1.3	1.2.2.1.2	1.2.2.1.2	-	1	0.5	0.5	1	0.67	-	-	-	**0.7333**
"ق"	1.1	1.1	1.1	1	1	1	-	-	-	-	-	-	**1**
"ف"	1.1	0.1	0.1	-	0	1	-	-	-	-	-	-	**0.5**
"ز"	-	-	-	-	-	-	-	-	-	-	-	-	-
.....	-	-	-	-	-	-	-	-	-	-	-	-	-
.....	-	-	-	-	-	-	-	-	-	-	-	-	-
"ر"	9	6	6	-	-	-	-	-	-	-	-	-	-
"ل"	5.3.1.1.1	2.3.1.2.1	2.3.1.2.1	-	0.4	1	1	0.5	1	-	-	-	**0.78**
"م"	1	1	1	1	1	-	-	-	-	-	-	-	**1**
"ت"	-	-	-	-	-	-	-	-	-	-	-	-	-
Total L1_PMR =													**0.7911**

Figure 7: Results of extraction of watermarks and detection using HFDATAI

As shown in Fig. 7, TP1 represents 1st transition of non-zero in the given text, TP2 represents 2nd transition, and so on. Some states have only one transition, which is shown in TP1. However, some states have more than one transitions, which are represented in TP1, TP2, ... etc. such as "ي" and "ق" states.

4 Implementation and Simulation

A variety of implementation and simulation are conducted to test the accuracy of HFDATAI output and tampering detection. This section outlines the settings for implementation and experimentation, conditions for experiments, typical dataset experimental scenarios, and a discussion of outcomes.

4.1 Simulation and Implementation Environment

The self-developed software was developed to evaluate and assess the efficiency of HFDATAII. The HFDATAI implementing environment is: CPU: Intel Core i7-4650U/2.3 GHz, RAM: 8.0 GB, Windows 10-64 bit, PHP VS Code IDE programming language.

4.2 HFDATAI Simulation and Experiment Findings

To evaluate the accuracy of tampering detection of HFDATAI, scenarios of many studies are performed as shown in Tab. 1, for all forms of attacks and their volumes.

Table 1: Assessment detection accuracy of HFDATAI under all volumes

Attack volume (%)	Insertion	Deletion	Reorder
5	84.48	92.03	89.39
10	79.18	93.58	87.54
20	61.10	81.93	76.77
s50	47.34	56.76	64.15

As the results shown in Tab. 1 and Fig. 8, it seems that the HFDATAI approach gives sensitive results of detection of tampering in all attacks that the structure, semantics, and syntax of the content of Arabic text may have been carried out. As a comparison of tampering based on attack types, the results show that the most sensitive tampering detection in all attack volume scenarios is the insertion attack.

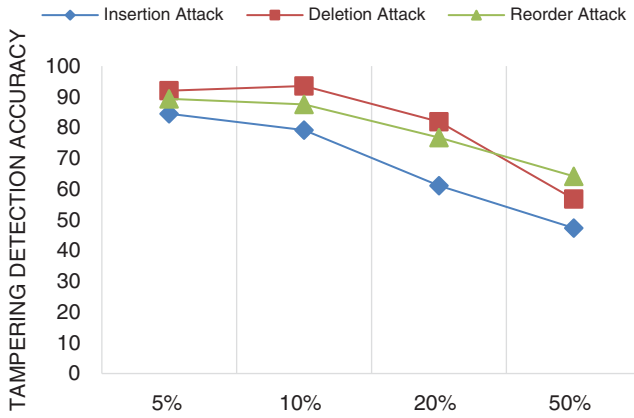

Figure 8: Detecting effect under all volumes of various attacks

5 Comparison and Result Discussion

The accuracy of tampering detection is carefully analysed and compared between HFDATAI and baseline algorithms ZWAFWMM [5] and RACAAT [30].

5.1 Results of Attack Type Impact

Tab. 2 shows a comparison of the different attack type's effects on tampering detection accuracy of HFDATAI, ZWAFWMMM, and RACAAT approaches against all dataset scales and all attack volume scenarios.

Table 2: Detection impact comparison based on attack type

Method	Insertion	Deletion	Reorder
ZWAFWMMM	80.02	69.35	44.88
RACAAT	74.28	59.99	37.23
HFDATAI	68.03	81.07	79.46

Tab. 2 and Fig. 9 demonstrate how RACAAT and ZWAFWMMM tampering detection precision for HFDATAI is determined by the form of attack. In the event of an insertion attack, a low impact between the detection accuracy of the HFDATAI approach and the ZWAFWMMM baseline approaches as well as the RACAAT approach was observed. However, with baseline approaches, the high impact has been observed when attacks are removed and reordered, and findings show that HFDATAI exceeds ZWAFWMMM and HAZWCTW with better detection precision for manipulations. This indicates that the HFDATAI suggested approach to content

authentication and manipulation of Arabic text in all forms of attacks in which reorder attacks simultaneously reflect deletion and insertion attacks are highly supported and very sensitive.

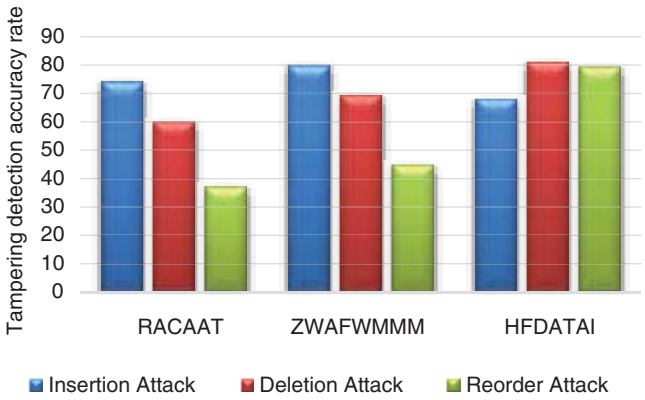

Figure 9: Attack type-based comparison of tampering detection effect

5.2 Results of Attack Rates Impact

Tab. 3 provides a comparison of the multiple attack volume effects on the performance of tampering detection for both dataset size and volume scenarios. The comparison is performed using HFDATAI with RACAAT and ZWAFWMMM approaches.

Table 3: Detection accuracy comparison based on attack rates

Attack volume (%)	ZWAFWMMM	RACAAT	HFDATAI
5	82.09	83.60	88.63
10	72.74	74.33	86.76
20	57.71	59.39	73.27
50	13.66	37.56	56.08

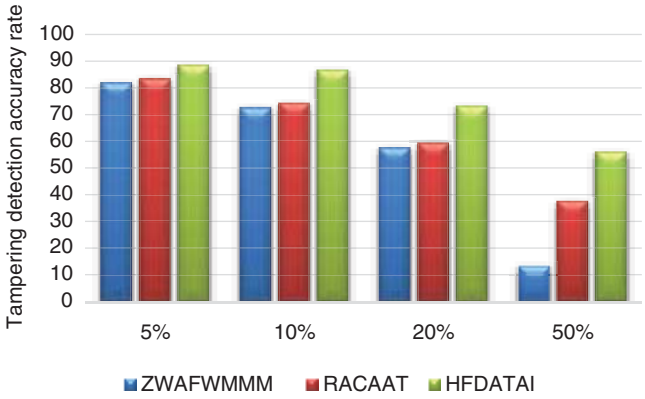

Figure 10: Attack rate-based comparison of tampering detection accuracy

Tab. 3 and Fig. 10 demonstrate how the precision of deception is affected by low, medium, and high attack amounts. Fig. 10 shows in general that as the volume of the attack increases, the accuracy of tampering detection also increases. HFDATAI conducts RACAAT and ZWAFWMM concerning their efficiency and identification precision in low, mid, and high amounts of attacks, always with low, medium and high amounts of attacks. This makes HFDATAI highly recommended for the authentication of content and the exploitation of any transmitted Arabic-text through the Internet.

5.3 Results of Dataset Impact

This section tests the various dataset size impact on watermark reliability against all forms of attacks within their multiple volumes. Tab. 4 shows a comparison of that effect using HFDATAI with RACAAT and ZWAFWMMM approaches.

Table 4: Detection accuracy comparison based on the Arabic text size

Dataset size	ZWAFWMMM	HNLPZWA	HTAZWA
[ASST]	67.272	69.534	72.93
[AMST]	63.802	68.126	76.63
[AHMST]	59.233	65.108	76.92
[ALST]	54.466	62.073	78.28

The comparative results as shown in Tab. 4 and Fig. 11 reflect the tampering detection accuracy of the HFDATAI approach suggested. The findings illustrate that in the proposed HFDATAI approach, the highest impact of the dataset scale leads to the best accuracy of tampering detection that is ordered as ASST, AMST, AHMST, and ALST, respectively. This means that the accuracy of tampering detection increases with the decreased Arabic text size and decreases with the increased Arabic text size. On the other hand, results show that the HFDATAI approach outperforms both RACAAT and ZWAFWMMM approaches in terms of tampering detection accuracy for all sizes of the Arabic dataset.

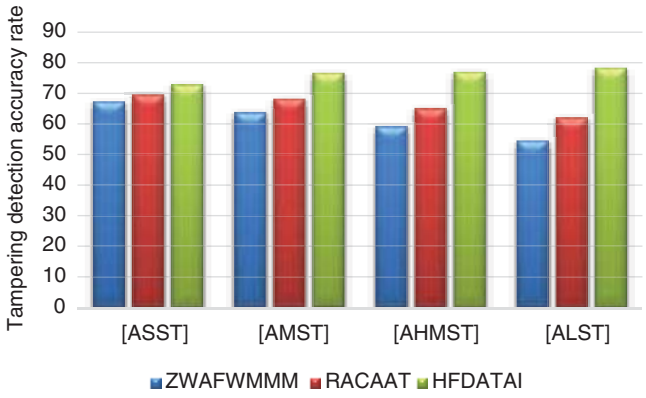

Figure 11: Arabic text size-based comparison of tampering detection impact

6 Conclusion

Cantered on the hidden Markov model mechanism of low-level order and alphanumeric, a high-sensitive approach for detecting the tampering attacks on Arabic text transmitted via the Internet (HFDATAI) has been proposed in this paper by integrating soft computing and digital watermarking techniques. Soft computing and NLP used in HFDATAI to perform a text analysis process to find interrelationships between the content of the Arabic-text provided and the main watermark created. Without modification or impact on the scale of the original text, the created watermark should logically be embedded in the original Arabic background. The embedded watermark can be used later to identify any manipulation that happens on the Arabic text after the text is distributed through the Internet. HFDATAI method in PHP programming language was developed and applied using VS code IDE. Experiments are done on various regular Arabic datasets in different amounts of attacks. The baseline approaches RACAAT and ZWAFWMM were compared to HFDATAI. The findings reveal that CZWNLPA beats RACAAT and ZWAFWMM in terms of identification precision of tampering and fragility of watermarks. Furthermore, the findings illustrate that HFDATAI refers to all Arabic literature, numbers, spaces, and special characters, for future research, the enhancement of identification precision and watermark fragility for all kinds of attacks should be considered.

Funding Statement: The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work under Grant Number (G.R.P./14/42), 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. 195–2011, 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. 804, 2019.

[5] 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.

[6] 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.

[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.

[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.