Computers, Materials & Continua
DOI:10.32604/cmc.2020.012088
Article

Tech Science Press

A Hybrid Intelligent Approach for Content Authentication and Tampering Detection of Arabic Text Transmitted via Internet

Fahd N. Al-Wesabi[1,2,*]

[1]Department of Computer Science, King Khalid University, Muhayel Aseer, Saudi Arabia
[2]Faculty of Computer and IT, Sana'a University, Sana'a, Yemen
[*]Corresponding Author: Fahd N. Al-Wesabi. Email: falwesabi@kku.edu.sa
Received: 14 June 2020; Accepted: 30 June 2020

Abstract: In this paper, a hybrid intelligent text zero-watermarking approach has been proposed by integrating text zero-watermarking and hidden Markov model as natural language processing techniques for the content authentication and tampering detection of Arabic text contents. The proposed approach known as Second order of Alphanumeric Mechanism of Markov model and Zero-Watermarking Approach (SAMMZWA). Second level order of alphanumeric mechanism based on hidden Markov model is integrated with text zero-watermarking techniques to improve the overall performance and tampering detection accuracy of the proposed approach. The SAMMZWA approach embeds and detects the watermark logically without altering the original text document. The extracted features are used as a watermark information and integrated with digital zero-watermarking techniques. To detect eventual tampering, SAMMZWA has been implemented and validated with attacked Arabic text. Experiments were performed on four datasets of varying lengths under multiple random locations of insertion, reorder and deletion attacks. The experimental results show that our method is more sensitive for all kinds of tampering attacks with high level accuracy of tampering detection than compared methods.

Keywords: HMM; NLP; text analysis; zero-watermarking; tampering detection

1 Introduction

Security issues of digital text in various languages and formats have assumed great importance in communication technologies, especially in terms of content authentication, integrity verification, and copyright protection. Numerous applications, such as e-commerce and e-Banking, impose many challenges during transfer of content via the internet. Most of the digital media transferred over the internet is in text form and is highly sensitive in terms of content, structure, syntax, and semantics. Malicious attackers may temper these digital contents during the transfer process, and thus the modified content can result in incorrect decisions [1]. Several solutions of information security have been proposed for many proposes which includes encryption, data hiding, copyright protection, integrity verification and unauthorized access control [2,3]. Steganography and digital watermarking (DW) are the most common techniques of information hiding for several proposes such as content authentication and copyright

protection whenever, if any altering is made in the watermarked media, the original media still protected and prove its ownership [4]. DW uses a specific algorithm to hide information by embedding it in the digital images, audio, video or text [5]. Several traditional text watermarking methods and solutions have been proposed and classified in various categories such as structure based, linguistic based, binary image based and format based [6]. Most of these solutions require some modifications or transformations on original digital text contents in order to embed the watermark information within text. Zero-watermarking is a modern technique that can be used with smart algorithms without any modification on original digital contents to embed the watermark information [7,8]. The most challenges in this area involve developing the appropriate methods to hide information in the sensitive text contents without any modification of it [9]. In last decade, the most common digital media transferred among various internet applications is in the form of text. Nevertheless, limited research focused in text solutions because text is natural language dependent and it is difficult to hide security information unlike images which security information can hide in pixels, audio in waves and video in frames [10]. Digital Holy Qur'an in Arabic, eChecks, online exams and marking are some examples of such sensitive digital text content. Various features of Arabic alphabets such as diacritics, extended letters, and other Arabic symbols make it easy to change the main meaning of text content by making simple modifications such as changing diacritics arrangements [11–14]. Hidden Markov model (HMM) is the most common technique of natural language processing (NLP), which is used for text analysis and extract the text features.

In this paper, the authors present an intelligent hybrid approach for content authentication and tampering detection of Arabic text, called SAMMZWA. The proposed technique combines Markov Model and zero watermarking. The second order of alphanumeric mechanism of Markov model is used for text analysis in order to extract the interrelationships among contents of the given Arabic text which consequently generates the watermark key. The generated watermark is logically embedded in the original Arabic context without modifications of the original text. After transmission of the text, the embedded watermark is used to detect any tampering with the received Arabic text and ensures the authenticity of the transmitted text. The objective of the SAMMZWA approach is to achieve high accuracy of content authentication and sensitive detection of tampering attack in Arabic text.

The rest of the paper has five more sections. Section 2 provides a literature review of the related work. Section 3 presents SAMMZWA. Section 4 describes the implementation, simulation, and experimental details. Section 5 describes the comparison and results discussion, and Section 6 offers conclusions.

2 Literature Review

In the literature, several research on text watermarking approaches and methods have been proposed for several proposes of information security. In this paper, the authors briefly review the most common classifications of text watermarking methods which are linguistic-based watermarking, structural-based watermarking, and zero-watermarking methods [15].

2.1 Linguistic-Based Methods

The linguistic-based text watermarking methods are naturally language-based techniques, which works by making some modifications to the semantics and syntactic nature of plain text in order to embed the watermark key. In the linguistic and semantic-based approaches, information is hidden by making some manipulations on words and utilize them as watermark key using many methods and techniques such as synonym substitution, typos, noun-verbs, and text-meaning representational strings [16].

One of the syntactic-based method proposed in [17]. The proposed method uses open word space to improve the capacity of Arabic text. It works by utilize each word space to hide the binary bit 0 or 1 through which physical modification of the original text is conducted. Other syntactic-based methods

presented in [18] for copyright protection by considers the existence of Harakat (diacritics, i.e., Fat-ha, Kasra, and Damma) in the Arabic language and reverses the Fatha for message hiding. Other English text watermarking method also make use of Unicode characters to hide the watermark information within English scripts. The ASCII code used for embedding is 00, however, and the Unicode used of multilingual for embedding are 01, 10, and 11 [19].

2.2 Structural-Based Methods

The structural-based text watermarking approaches are based on content structure which alters the features or structure of the text to embed the watermark information [20]. This also include modifications in general formatting features of the original text to hide watermark key such as locations of letters or words, writing style, repeating some letters or altering the features of the text [21,22]. One of the early approaches following the structure-based approach change the locations of words in text [23]. Other structural-based approaches proposed in [24,25] for content authentication of Chinese text by merging properties of sentences and calculate its entropy. In these approaches, the contents of Chines text divided into sets of small sentences and obtain semantic code of each word, then calculate its entropy by semantic codes' frequency, and find the weight of each sentence by utilizing the sentence features such as entropy, length, relevance, and weight function. The extracted features utilized to generate the watermark by using the verbs, and nouns of the high-weight sentences.

2.3 Zero Watermark-Based Methods

Zero text watermark-based approaches are based on text features which is achieved by generating the watermark key from the text context. This means several text features should be obtained, extracted and utilized as a watermark information. Several techniques and solutions have been proposed based on text features includes number of words or sentences letters, first letter of each word, and appearance frequency of non-vowel ASCII letters and words [26–34]. One of the available text zero-watermarking approaches presented in [26] which hide the watermarking information within the social media and validate it later in terms of accuracy and reliability. Other text zero-watermarking techniques proposed in [27] to validate data integrity of text context over the internet of things. The watermark information is generated as a text features such as text size, data appearance frequency, and time of data capturing. The generated watermark will have to be embedded logically in the original contents before its transmission. Reference [28] shows a text zero-watermarking method has been developed for individual privacy protection based on certain measures such as Hurst exponent and zero crossing of the speech signals. Individual identity has to obtained to embed it as a watermark key. In the case of copyright protection of English text, a text zero-watermark methods have been proposed in [29,30] which uses the appearance frequency of non-vowel ASCII letters and words.

According to combination solutions with zero-watermarking, such solutions presented in [31,32] which uses natural language processing and zero-watermarking for content authentication. The proposed methods trying to extract some text features to obtain the text probability properties and utilize it as a watermark key. Reference [33] shows a spatial domain technique for copyright protection, data security and content authentication of multimedia images. A robust geometric features-based method has been presented in [34] to improves capacity and watermark robustness.

3 The Proposed Approach

This paper proposes a novel reliable approach by integrating NLP and text zero-watermark techniques which there is no need to embed extra information such as watermark key, or even to perform any modifications on the original text. As a result of hybrid solution, very low impact of overall complexity is

resulted in terms of time, but in the other hands, there is no possibility for attackers to figure out the mechanism work of algorithm. Second level order of alphanumeric mechanism of Markov model has been used as NLP technique to analyze the contents of Arabic text and extract the interrelationships features of these contents. The main contributions of our approach, SAMMZWA can be summarized as follows:

- Unlike the previous work, in which the watermarking is performed by effecting text, content, and size, our approach SAMMZWA embeds the watermarking logically without any effect on the text, content, and size.
- In our SAMMZWA approach, watermarking does not need any external information because the watermark key is produced as a result of text analysis and extracting the relationship between the content itself and then making it as a watermark.
- Our SAMMZWA approach is highly sensitive to any simple modification on the text and the meaning in the Arabic text, which is known as complex text, including the Arabic symbols which can change the meaning of the Arabic word. The three contributions mentioned above are found somehow only in images but not in text. This is the vital point concerning to the contribution of this paper.
- In addition, our SAMMZWA approach can effectively determine the place of tempering occurrence. This feature can be considered an advantage over Hash function method.
- SAMMZWA has been implemented, simulated using various several standard datasets, and compared to other baseline approaches under all performance metrics to find which approach gives the best accuracy of tampering detection. Simulation and comparison results prove the accuracy and effectiveness of SAMMZWA approach in terms of tampering detection of unauthorized attacks. Baseline approaches and their execution parameters are presented later in Section 5.1.

The following subsections explain in detail two main processes that should be performed in SAMMZWA, namely watermark generation and embedding process, and watermark extraction and detection process.

3.1 Watermark Generation and Embedding Process

The three main sub-processes included in this process are pre-processing, Arabic text analysis and watermark generation, and watermark embedding as illustrated in Fig. 1.

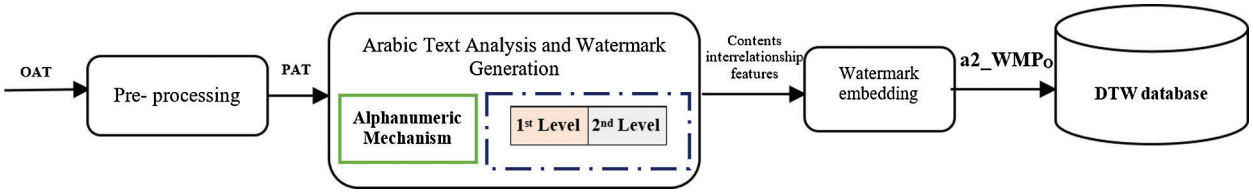

Figure 1: Text analysis, watermark generation and embedding processes of SAMMZWA

3.1.1 Pre-Processing Process

The pre-processing of the original Arabic text is a key step for Arabic text analysis and watermark generation process to remove extra spaces and new lines, and it will be influence directly on the tampering detection accuracy. The original Arabic text (OAT) is required as input for the pre-processing process.

3.1.2 Text Analysis and Watermark Generation Process

This process include two sub processes are building Markov matrix, and text analysis and watermark generation processes.

- *Building a Markov matrix* is the starting point of Arabic text analysis and watermark generation process using Markov model. A Markov matrix that represents the possible states and transitions available in a given text is constructed without reputations. In SAMMZWA approach, each unique pair of alphanumeric available in the given Arabic text represents a present state, and each unique alphanumeric represent a transition in Markov matrix. During the building process of the Markov matrix, the proposed algorithm initializes all transition values by zero to use these cells later to keep track of the number of times that the i^{th} unique pair of alphanumeric is followed by the j^{th} single alphanumeric in the given Arabic text.

Pre-processing and building Markov matrix algorithm is executed as presented in Algorithm 1.

```
PROCEDURE Pre_BMM (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
                    i.   PAT ← trim ("space" or "newLine")
6.  // Build list of non values text words
7.  a2_mm = { }
8.      for each alphanumeric in PAT
9.          if alphanumeric not in a2_list
10.             a2_mm ← a2_mm U {alphanumeric}
11.         for ps = 1 to a2_mm.length – 2
12.             for ns = 1 to a2_mm.length
13.                 a2_mm[ps][ns] = 0
14. return a2_mm
```

Algorithm 1: Pre-processing and building Markov algorithm of SAMMZWA

Where, OAT: is an original Arabic text, PAT: is a pre-processed Arabic text, a2_mm: is a states and transitions matrix with zeros 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 $a2\_mm[i][j]$, which represents the backbone of Markov model.

a2_mm[i][j] length is dynamic in which it is depending on the given text contents and size. The matrix rows refer to the states which is equal to the total number of unique pair of alphanumeric in the given text. However, the matrix columns refer to transitions which is fixed, which is equal to sixty-two possible transitions (twenty-eight alphabets of Arabic letters, space letter, ten integer numbers "0–9," and twenty-four specific symbols, i.e., (. ' " , ; : ? ! / \ @ $ & % * + - = >< [ ]).

- *Text analysis and watermark generation process*: after the Markov matrix is constructed, natural language processing and text analysis process should be performed to find the interrelationships between contexts of the given Arabic text and generate watermark patterns. In this sub-process, the appearance number of possible next transitions for each current state of pair of alphanumeric will calculated and constructed as transition probabilities by Eq. (1) below.

$$a2\_mm[ps][ns] = \sum_{i,j=1}^{n-2} trans[i][j] \tag{1}$$

where,

- trans: is total of all possible transitions.

- n: is total of all possible states.
- i: is i^{th} present state of pair of alphanumeric.
- j: is j^{th} next transition.

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

When using the second level order of alphanumeric mechanism of HMM, every unique pair of alphanumeric is a present state. Text analysis is processed as the text is read to obtain the interrelationship between present state and the next states as illustrated in Fig. 2.

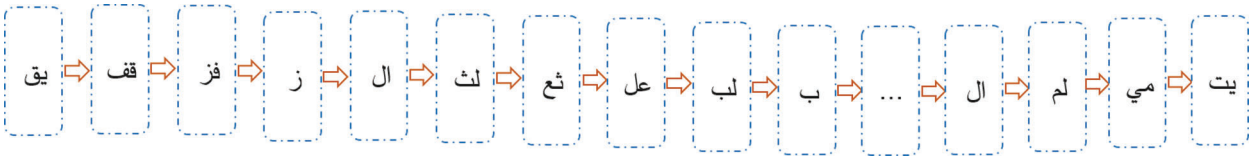

Figure 2: States representation of Arabic text sample using SAMMZWA

Text analysis is performed using HMM to extract the features and find the interrelationship between the contents of the given text, we represent 39 unique present states and their 61 possible transitions as illustrated below in Fig. 3.

State ID	Present States		Next Transition(s)
1	("يق")	⟹	[ف]
2	("قف")	⟹	[ز]
3	("فز")	⟹	[ ]
4	("ز ")	⟹	[ا]
...	.....	⟹	.....
...	.....	⟹	.....
36	("ال")	⟹	[ث، ب، س، ث، ب، ب، ى، ث، ب، م]
37	("لم")	⟹	[ي]
38	("مي")	⟹	[ت]
39	("يت")	⟹	[.]

Figure 3: Sample of an Arabic text states and their transitions using SAMMZWA

Is it possible to have two or more no-zero value in a row, for instance, it is assumed here that "ال" is a present state of pair alphanumeric, and the available next transitions are "م، ث، ب، س، ث، ب، ب، ى، ث، ب". It is observed that ten transitions are available in the given Arabic text sample and "ث" transitions repeat three times.

Algorithm of text analysis and watermark generation based on the second-level order of alphanumeric mechanism of Markov model proceeds as illustrated in Fig. 4.

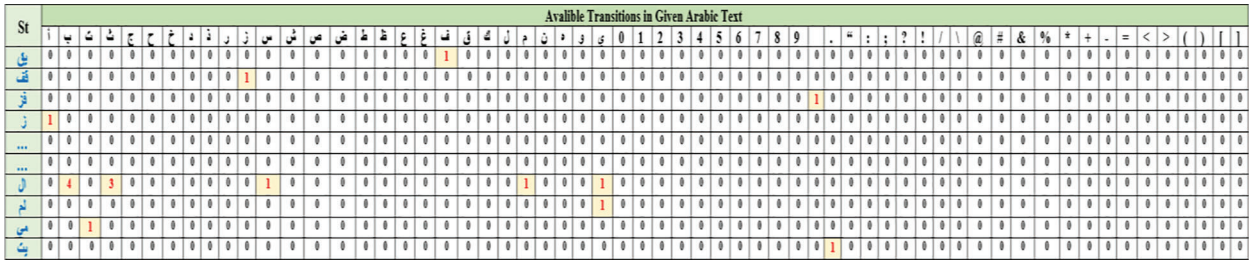

Figure 4: Feature extraction and watermark generation of the given text using SAMMZWA

Arabic text analysis and watermark generation algorithm is presented formally and executed as illustrated in Algorithm 2.

PROCEDURE WMGEN(PAT)

1. Input: PAT, IMM
2. Output: FM
3. BEGIN
4. Pre_BMM(PAT)
5. ps = first_pair_of_alphanumeric(PAT)
6. pa2 = PAT – [ps] // begin with 2nd unique pair of alphanumeric
7. fm = a2_mm
8. **for each** ns **in** pa2
9. fm[ps][ns] = fm[ps][ns] + 1
10. ps = ns
11. **return** fm

Algorithm 2: Watermark generation algorithm of SAMMZWA

where, ps: refers to the state of unique pair of alphanumeric, ns: represents the next state.

3.1.3 Watermark Embedding Process

In the proposed SAMMZWA approach, embedding the watermark have to done logically or digitally into the given text, without physically applying the watermark inside the text and no need to alter the original text. This is achieved by extracting the features of the given text and generating the watermark key by finding all non-zero values in Markov matrix and concatenate them sequentially to generate the original watermark pattern a2_WMP$_O$, as given in Eq. (2) and illustrated in Fig. 5.

1 – 1 – 1 – 1 – 4.3.1.1.1 – 1 – 1 – 1

Figure 5: Generated watermark a2_WMP$_O$ using SAMMZWA

$$a2\_WMP_O \;\&= \; a2\_mm[ps]\,[ns],\; for\; i, j = non - zero\; values\; results\; in\; a2\_mm \tag{2}$$

Watermark embedding process based on second level order of alphanumeric mechanism of Markov model is presented formally and executed as illustrated in Algorithm 3.

```
PROCEDURE WMEBED(PAT)

1.  Input: pre-processed text (PAT)
2.  Output: original watermark patterns
3.  BEGIN
4.  WMGEN(PAT)
5.  for ps = 1 to a2_arrList.Length - 2,
6.      for ns = 1 to a2_arrList.Length,
7.          if a2_mm [ps][ns] != 0
8.              a2_WMPO &= a2_mm [ps] [ns]
9.  return a2_WMPO
```

Algorithm 3: Watermark embedding algorithm of SAMMZWA

3.2 Watermark Extraction and Detection Process

While the watermarked key is kept secret and ready for detection and verification process, the original watermarked texts are shared to others via Internet. In order to verify the authenticity of the attacked text (AAT_P), the watermarked text is collected with its watermark key $a2\_EWM_A$ and compared with the original watermarked text (PAT) with its watermark key $a2\_WMP_O$.

Two sub-processes are involved in this process, which are watermark extraction and watermark detection as illustrated in Fig. 6.

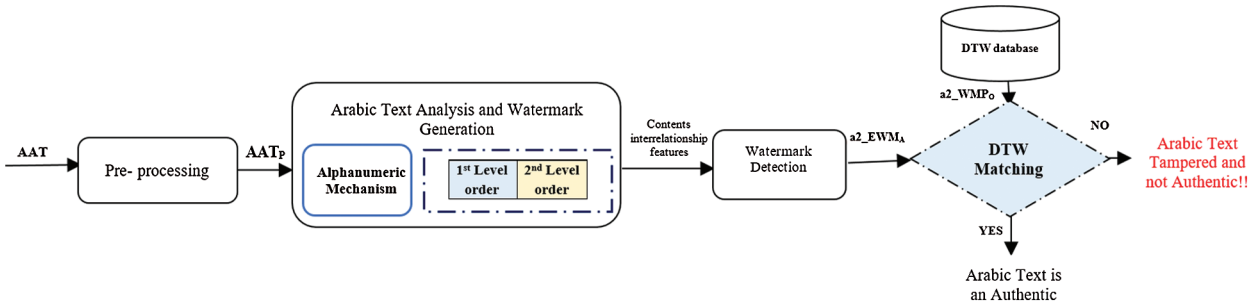

Figure 6: Watermark extraction and detection processes using SAMMZWA

3.2.1 Watermark Extraction Algorithm

The pre-processed attacked Arabic text (AAT_P) is required as input to this sub-process. However, attacked watermark patterns ($a2\_WMP_A$) is an output as illustrated in Algorithm 4.

```
PROCEDURE WMEXTR(AATP)

1.  Input: pre-processed text (AATP)
2.  Output: attacked watermark patterns (a2_WMPA).
3.  BEGIN
4.  WMGEN(AATP)
5.  for ps = 1 to a2_arrList'.Length - 2,
6.      for ns = 1 to a2_arrList'.Length,
7.          if a2_mm[ps][ns] != 0,
8.              a2_WMPA &= a2_mm'[ps] [ns],
9.  return a2_WMPA
```

Algorithm 4: Watermark extraction algorithm of SAMMZWA

3.2.2 Watermark Detection Algorithm

This process aims to verify the authenticity of the attacked text (AAT_P) and notify it is authentic or tampered. This process is achieved by compare $a2\_WMP_A$ and $a2\_WMP_O$ in both state and transition level matching as follows:

- *State level matching:* is a default matching, which compare a whole pattern of $a2\_WMP_O$ and $a2\_WMP_A$. The result of this matching is TRUE of FALSE. TRUE notification refer to authenticity of the text without any tampering occurred. Otherwise, FALSE notification refers to tampering detected and then it continues to the transition level matching.
- *Transition level matching:* each transition in $a2\_WMP_A$ will be compared with the equivalent transition of $a2\_WMP_O$ as given by Eqs. (3) and (4).

$$a2\_PMR_T(i,j)\left|\frac{a2\_WMP_O[i][j] - \left(a2\_WMP_O[i][j] - a2\_WMP_A[i][j]\right)}{a2\_WMP_O[i][j]}\right| \tag{3}$$

where,

- $a2\_PMR_T$: refers to pattern matching rate value, $(0 < a2\_PMR_T \leq 1)_T$.

$$a2\_PMR_S(i) = \left|\frac{\sum_{j=1}^{n-2}(a2\_PMR_T(i,j))}{Total\ StatePatternCount\ (i)}\right| \tag{4}$$

where,

- $a2\_PMR_S$: refers to matching rate in state level, $(0 < a2\_PMR_S \leq 100)$.

The weight of each state will be calculated as give in Eq. (5).

$$a2\_Sw = \left|\frac{a2\_PMR_S(i) * tf(i)}{m}\right| \tag{5}$$

where,

- tf: refers to value of transitions frequency.
- m: refers to total number of transitions.
- $a2\_PMR_S$: is the total pattern matching rate of i^{th} state for each unique pair of alphanumeric.
- i: is a number of all available states.

The final a2_PMR of AET_P and OAT_P are calculated by Eq. (6).

$$a2\_PMR = \left| \frac{\sum_{i=1}^{n-2} a2\_PMRS(i)}{N} \right| \tag{6}$$

where,

- N: is a total number of non-zeros values in a2_mm[i][j].

The distortion rate of watermark pattern refers to the detected tampering rate, which is denoted by a2_WDR and calculated by Eq. (7).

$$a2\_WDR = 1 - \ a2\_PMR * 100 \tag{7}$$

The watermark detection process is executed as illustrated in Algorithm 5.

PROCEDURE WMDET(a2_WMP_O, a2_WMP_A)

1. Input: pre-processed text (a2_WMP_O, a2_WMP_A)
2. Output: a2_PMR, a2_WDR
3. BEGIN
4. *// getting watermark of the original Arabic text*
5. WMGEN (a2_WMP_O)
6. *// extract watermark patterns from the attacked Arabic text*
7. WMEXTR (a2_WMP_A)
8. *// perform matching process between the original and attacked watermark patterns*
9. **IF** a2_WMP_A = a2_WMP_O
10. Print "Arabic text is authentic and no tampering occurred"
11. A2_PMR = 100
12. **Else**
13. Print "Arabic document is not authentic and tampering occurred"
14. *// compute pattern matching rate on transition level*
15. **for** i = 1 **to** a2_arrList'.Length - 2,
16. **for** j = 1 **to** a2_arrList'.Length
17. **IF** a2_WMP_O[i][j] != 0
18. pattern Count +=1
19. $a2\_PMR_T(i,j) = \left| \frac{a2\_WMP_O[i][j] - (a2\_WMP_O[i][j] - a2\_WMP_A[i][j])}{a2\_WMP_O[i][j]} \right|$
20. transPMRTotal += a2_PMR_T
21. **ELSE**
22. **IF** a2_WMP_A[i][j] != 0
23. patternCount += a2_WMP_A[i][j]
24. *// compute pattern matching rate on state level*
25. $a2\_PMR_S(i) = \left| \frac{\sum_{j=1}^{n-2}(a2\_PMR_T(i,j))}{Total\ StatePatternCount(i)} \right|$
26. $sWeight = \frac{a2\_PMR_S(i) * Transitions\ frequency(i)}{total\ no\ of\ transitions}$
27. a2_SW += stateWeight
28. *// compute pattern matching rate on a whole a given text*
29. $a2\_PMR = \frac{\sum_{i=1}^{n-2}(a2\_SW) * Total\ number\ of\ transitions}{Total\ number\ of\ transitions} * 100$
30. *// compute watermark distortion rate on a whole a given text*
31. a2_WDR = 1 – a2_PMR * 100
32. **return** a2_PMR, a2_WDR

Algorithm 5: Watermark detection algorithm of SAMMZWA

where,

- a2_PMR: refers to tampering detection accuracy ($0 < a2\_PMR \leq 100$).
- a2_WDR: refers to distortion rate ($0 < a2\_WDR_S \leq 100$).

The results of watermark extraction and detection process of the given sample of Arabic text using SAMMZWA are illustrated in Fig. 7.

States	Original WM Patterns	Extracted WM Patterns	Destroyed WM Patterns	Primary Matching Rate	L2_PMRt(I,j) of Transition Level								L2_PMRs(I,j) of State Level
					TP1	TP2	TP3	TP4	TP5	TP6	TP7	TP8	
"يق"	1	3	3	-	0.67	-	-	-	-	-	-	-	0.6667
"قف"	1	1.2	1.2	-	1	0.5	-	-	-	-	-	-	0.75
"فز"	1	1	1	1	1	-	-	-	-	-	-	-	1
"ز"	1	1	1	1	1	-	-	-	-	-	-	-	1
.....	-	-	-	-	-	-	-	-	-	-	-	-	0
.....	-	-	-	-	-	-	-	-	-	-	-	-	0
"ال"	4.3.1.1.1	1.3.1.2.1	1.3.1.2.1	-	0.25	1	1	0.5	1	-	-	-	0.75
"لم"	1	1	1	1	1	-	-	-	-	-	-	-	1
"مي"	1	1	1	1	1	-	-	-	-	-	-	-	1
"يت"	1	1	1	1	1	-	-	-	-	-	-	-	1
Total L2_PMR =													**0.8958**

Figure 7: Result of watermark extraction and detection process of the given text using SAMMZWA

As shown in Fig. 7, TP1 represents first transition of non-zero in the given text, TP2 represents second transition, and so on. Some states have only one transition, which is shown in TP1. However, some states have more than transitions, which are represented in TP1, TP2, and so on, such as "قف" and "ال" states.

4 Implementation and Simulation

To evaluate the tampering detection accuracy of SAMMZWA, several simulation and experiments are performed using self-developed program, various standard dataset, and predefined tampering attacks with various attack volumes as explained in the following sub sections.

4.1 Implementation Environment and Setup

A self-developed program has been developed to test and evaluate the performance of SAMMZWA. Implementation environment of SAMMZWA are: 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 Simulation and Experimental Parameters

Tab. 1 shows an experimental and simulation parameters and their associated values that used to perform the experiments of the proposed SAMMZWA approach.

Table 1: Experimental and simulation parameters

Parameters	Value
Arabic dataset size	[ASST, 179], [AMST, 421], [AHMST, 559] and [ALST, 2018]
Attack type	Insertion, deletion and rephrasing
Attack volumes	5%, 10%, 20% and 50%
Tampering detection accuracy	High with close to 100 Low with close to 0
a2_PMR	(High when a2_PMR > 70, Mid when 40 < a2_PMR < 70, and Low when a2_PMR < 40)

4.3 Performance Metrics

Tampering detection accuracy refers to the performance of the SAMMZWA approach, which is evaluated using the following metrics:

- Tampering detection accuracy (a2_PMR and a2_WDR) under all mentioned attack types and volumes.
- Desired tampering detection accuracy values which close to 100%.
- Comparison and results evaluation of dataset size effect, attack types effect, and attack volumes effect against tampering detection accuracy using the proposed SAMMZWA approach, HNLPZWA and ZWAFWMMM baseline approaches.

4.4 Simulation, Experiments and Results Discussion with SAMMZWA

In this sub section, author evaluates the tampering detection accuracy of SAMMZWA. The letter set cover all Arabic letters, spaces, numbers, and special symbols. Experiments were conducted in different volumes of datasets, various kinds of attacks with their rates as identified above in Tab. 1. The simulation and experiments results are shown in tabular form in Tab. 2 and graphically illustrated in Fig. 8.

Table 2: Tampering detection accuracy evaluation of SAMMZWA approach

Attack Volume	Insertion	Deletion	Reorder
5%	88.00	91.68	93.35
10%	82.47	92.32	85.24
20%	68.54	83.06	78.39
s50%	52.39	59.28	65.69

From Tab. 2 above and Fig. 8 below, we can see that SAMMZWA shows the best tampering has been detected by reorder attack in cases of both large (50%) and very low (5%) volumes of attack because reorder attack represents both insertion and deletion attacks in the same time. Whereas, in case of mid attack volume, high tampering is detected under deletion attack. This mean that, SAMMZWA gives best detection accuracy and high sensitive to tampering under both deletion and reorder attacks in all scenarios of attack volumes.

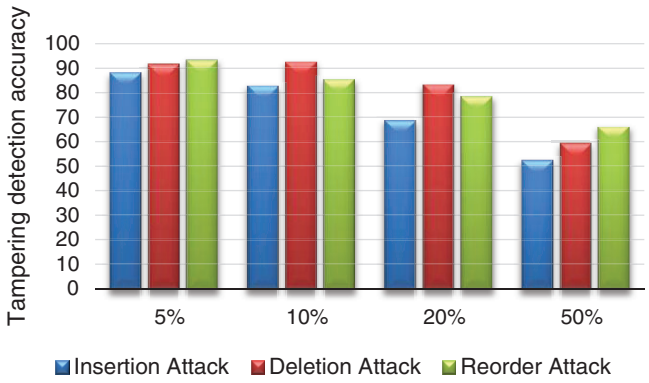

Figure 8: Tampering detection accuracy evaluation of SAMMZWA under all attacks and various volumes

5 Comparison and Result Discussion

The tampering detection accuracy results were critically analyzed, effect study and compared between SAMMZWA and baseline approaches ZWAFWMMM and HNLPZWA and shows discussion of their effect under the major factors, i.e., dataset size, attack types and volumes.

5.1 Baseline Approaches

Tampering detection accuracy of SAMMZWA is compared with HNLPZWA (an intelligent hybrid of natural language processing and zero-watermarking approach) and ZWAFWMMM (Zero-Watermarking Approach based on Fourth level order of Arabic Word Mechanism of Markov Model) [35]. Comparison is performed under all performance metrics mentioned above in Sub Section 4.3 to find which approach gives the best accuracy of tampering detection.

5.2 Comparison of SAMMZWA with ZWAFWMMM and HNLPZWA Approaches

This subsection presents the tampering detection accuracy comparison of SAMMZWA with ZWAFWMMM and HNLPZWA approaches and study their effect under core affected factors are dataset size, attack types and volumes.

5.2.1 Comparison and Results Study of Dataset Size Effect

In this subsection, authors present an evaluation of the different dataset size effects on tampering detection accuracy against all attack types under their different volumes. Fig. 3 shows a comparison of that effect using SAMMZWA along with ZWAFWMMM and HNLPZWA approaches.

As shown in the summary of the comparative results of Fig. 9 applying the SAMMZWA approach, the highest effects of dataset size that lead to the best tampering detection accuracy are ordered as ASST, AMST, AHMST and ALST, respectively. This means that tampering detection accuracy increased with the decreasing document size and decreased with the increasing document size. On the other hand, the results show that, the SAMMZWA approach outperforms both ZWAFWMMM and HNLPZWA approaches in terms of tampering detection accuracy under all scenarios of dataset sizes.

5.2.2 Comparison and Results Study of Attack Type Effect

Fig. 10 shows a comparison of the different attack types effect on tampering detection accuracy against all dataset sizes and all scenarios of attacks volumes. A comparison was performed using SAMMZWA with ZWAFWMMM and HNLPZWA approaches.

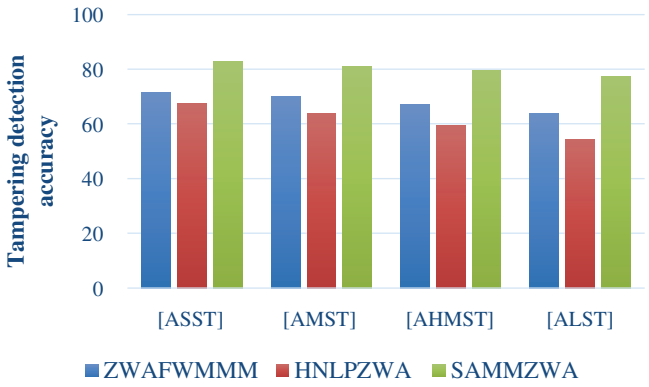

Figure 9: A comparison of dataset size effect on tampering detection accuracy using SAMMZWA with ZWAFWMMM and HNLPZWA approaches

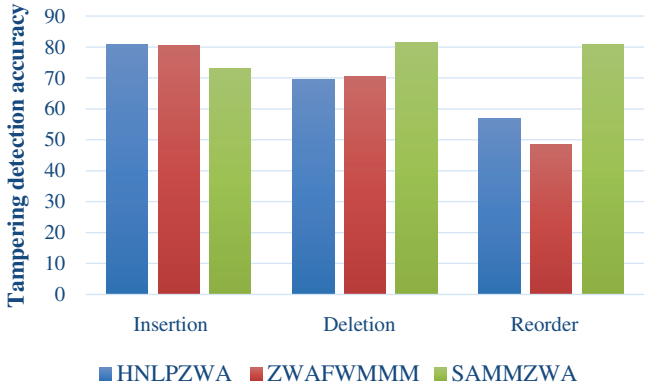

Figure 10: A comparison of attack type effect on tampering detection accuracy using SAMMZWA with ZWAFWMMM and HNLPZWA approaches

As shown in Fig. 10, the SAMMZWA approach outperforms both ZWAFWMMM and HNLPZWA in terms of tampering detection accuracy in all scenarios of deletion and rephrasing attacks. However, ZWAFWMMM and HNLPZWA approaches outperforms SAMMZWA approach in case of insertion attack. This means that SAMMZWA approach is strongly recommended and applicable for content authentication and tampering detection of Arabic text documents transmitted via internet in all cases of deletion and rephrasing attacks.

5.2.3 Comparison and Results Study of Attack Volume Effect

Fig. 11 shows a comparison of the different attack volume effects on tampering detection accuracy against all dataset sizes and all scenarios of attacks volumes. A comparison was performed using SAMMZWA with ZWAFWMMM and HNLPZWA approaches.

As shown in Fig. 11, the SAMMZWA approach outperforms ZWAFWMMM and HNLPZWA approaches in terms of tampering detection accuracy in all scenarios of low, mid and high volumes of all attacks. This means that SAMMZWA approach is strongly recommended and applicable for content authentication and tampering detection of Arabic text documents under all volumes of all attack types.

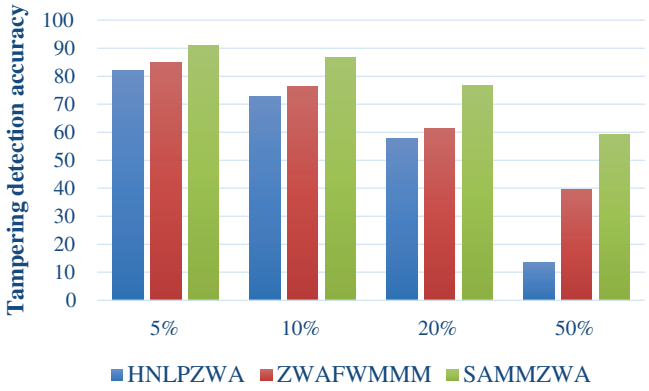

Figure 11: A compression of attack volume effect on tampering detection accuracy using SAMMZWA with ZWAFWMMM and HNLPZWA approaches

6 Conclusion

In this paper, SAMMZWA approach has been proposed by integrating a zero watermarking and natural language processing techniques for content authentication and tampering detection of Arabic text transmitted via the internet. SAMMZWA implemented using PHP self-developed program in VS code IDE as well as simulation and experiments using various standard dataset under different volumes of insertion, deletion, and rephrasing attacks. SAMMZWA approach has been compared with ZWAFWMMM and HNLPZWA approaches. Comparison results show that SAMMZWA outperforms ZWAFWMMM and HNLPZWA in terms tampering detection accuracy under deletion and reorder attacks. Although SAMMZWA approach is an efficient approach, and it is designed only for all scenarios of insertion attack. For the future work, the authors will consider detection accuracy under all scenarios of deletion and rephrasing attacks. Moreover, the authors also intend to evaluate the tampering detection accuracy using high level order of alphanumeric of Markov model.

Funding Statement: The author extends his appreciation to the Deanship of Scientific Research at King Khalid University for funding this work under grant number (R. G. P. 2/55/40/2019), Received by Fahd N. Al-Wesabi. www.kku.edu.sa

Conflicts of Interest: The author declares that he has no conflicts of interest to report regarding the present study.

References

[1] S. Nurul, A. Kamsin, L. Yee and H. Rahman, "A review of text watermarking: Theory, methods, and applications," *IEEE Access*, vol. 6, pp. 8011–8028, 2018.

[2] F. Peng, Q. Long, Z. X. Lin and M. Long, "A reversible watermarking for authenticating 2D CAD engineering graphics based on iterative embedding and virtual coordinates," *Multimedia Tools and Applications*, vol. 78, no. 19, pp. 26885–26905, 2019.

[3] A. Alwan, M. Shahidan, N. Amir, M. Hashim and M. S. Mohd, "A review and open issues of diverse text watermarking techniques in spatial domain," *Journal of Theoretical and Applied Information Technology*, vol. 96, pp. 5819–5840, 2018.

[4] C. G. Ho, J. J. Hyo, M. H. Min, K. Y. Tae and P. S. Bum, "User authentication system based on baseline-corrected ECG for biometrics," *Intelligent Automation and Soft Computing*, vol. 25, no. 1, pp. 193–204, 2019.

[5] P. Singh and R. S. Chadha, "A survey of digital watermarking techniques, applications and attacks," *International Journal of Engineering Innovation and Technologies*, vol. 2, no. 9, pp. 165–175, 2013.

[6] M. Kaur and V. Sharma, "Encryption based LSB steganography technique for digital images and text data," *International Journal of Computer Science and Network Security*, vol. 16, no. 9, pp. 90–95, 2016.

[7] Q. W. Yang, F. Peng, J. T. Li and M. Long, "Image tamper detection based on noise estimation and lacunarity texture," *Multimedia Tools and Applications*, vol. 75, no. 17, pp. 10201–10211, 2016.

[8] B. Kaur and S. Sharma, "Digital watermarking and security techniques: A review," *International Journal of Computer Science and Technology*, vol. 8, no. 2, pp. 44–47, 2017.

[9] S. Hakak, K. Amirrudin, O. Tayan, I. Yamani and G. Amin, "Approaches for preserving content integrity of sensitive online Arabic content: A survey and research challenges," *Information Processing and Management*, vol. 56, no. 2, pp. 367–380, 2019.

[10] N. Al-Maweri, R. Ali, A. Adnan, A. Ramli and S. Ahmad, "State-of-the-art in techniques of text digital watermarking: Challenges and limitations," *Journal of Computer Science*, vol. 12, no. 2, pp. 62–80, 2016.

[11] O. Tayan, M. Kabir and Y. Alginahi, "A hybrid digital-signature and zero-watermarking approach for authentication and protection of sensitive electronic documents," *Science World Journal*, vol. 2014, no. 1, pp. 1–14, 2014.

[12] S. Dhiman and O. Singh, "Analysis of visible and invisible image watermarking—A review," *International Journal of Computer Applications*, vol. 147, no. 3, pp. 36–38, 2016.

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

[14] K. Hameed, A. Khan, M. Ahmed, A. Goutham Reddy and M. M. Rathore, "Towards a formally verified zero watermarking scheme for data integrity in the Internet of things based-wireless sensor networks," *Future Generation Computer Systems*, vol. 82, pp. 274–289, 2018.

[15] S. Sameeka and P. Kalpesh, "Securing web contents through invisible text watermarking for copyright protection," *International Journal of Engineering Development and Research*, vol. 6, no. 3, pp. 257–261, 2018.

[16] J. Chen, H. F. Ma and Q. Lu, "Text watermarking algorithm based on semantic role labeling," *Proc. 3rd Int. Conf. of Digital Information Processing, Data Mining, Wireless Communication*, pp. 117–120, 2016.

[17] A. Reem and A. Lamiaa, "Improved capacity Arabic 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] S. Mujtaba and S. Asadullah, "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] S. Abdul, S. Wesam and A. Dhamyaa, "Text steganography based on Unicode of characters in multilingual," *International Journal of Engineering Research and Applications*, vol. 3, no. 4, pp. 1153–1165, 2013.

[20] A. Nasr addin, A. Wan, R. Abdul, S. Khairulmizam and M. Sharifah, "Robust digital text watermarking algorithm based on Unicode extended characters," *Indian Journal of Science and Technology*, vol. 9, no. 48, pp. 1–14, 2016.

[21] M. Kaur, "An existential review on text watermarking techniques," *International Journal of Computer Applications*, vol. 120, no. 18, pp. 29–32, 2015.

[22] R. Alotaibi and L. Elrefaei, "Arabic text watermarking: A review," *International Journal of Artificial Intelligence & Applications*, vol. 6, no. 4, pp. 01–16, 2015.

[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. 11, 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, 2014.

[25] P. Zhu, G. Xiang and W. Song, "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] T. Milad, "ANiTH: A novel intelligent text hiding technique," *IEEE Dataport*, vol. 10, 2018.

[27] H. Khizar, K. Abid, A. Mansoor and G. Alavalapati, "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.

[28] A. Zulfiqar, M. Shamim, M. Ghulam and A. Muhammad, "New zero-watermarking algorithm using Hurst exponent for protection of privacy in telemedicine," *IEEE Access*, vol. 6, pp. 7930–7940, 2018.

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

[30] M. Hanaa and A. Maisa'a, "Comparison of eight proposed security methods using linguistic steganography text," *International Journal of Computing & Information Sciences*, vol. 12, no. 2, pp. 243–251, 2016.

[31] M. Mokhtar, M. Fadl and F. Al-Wesabi, "Combined Markov model and zero-watermarking techniques to enhance content authentication of Arabic text documents," *International Journal of Computational Linguistics Research*, vol. 5, no. 1, pp. 26–42, 2014.

[32] F. Al-Wesabi, Z. Adnan and U. Kulkarni, "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, 2014.

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

[34] D. Tong, C. Zhu, N. Ren and W. Shi, "High-capacity and robust watermarking scheme for small-scale vector data," *KSII Transactions on Internet and Information Systems*, vol. 13, no. 12, pp. 6190–6213, 2019.

[35] F. Al-Wesabi, M. Khalid and N. Nadhem, "A zero watermarking approach for content authentication and tampering detection of Arabic text based on fourth level order and word mechanism of Markov model," *ELSEVIER Journal of Information Security and Applications*, vol. 52, pp. 1–15, 2020.