Combined Markov Model and Zero Watermarking Techniques to Enhance Content Authentication of English Text Documents

Mokhtar M. Ghilan[1], Fadl M. Ba-Alwi[2], Fahd N. Al-Wesabi[3]

[1]Information Systems Department, Faculty of Computer and Information Technology
Sana'a University, Yemen

[2]Faculty of Computer and Information Technology, Sana'a University, Yemen, P.O.Box 1247

[3]Faculty of Engineering, SRTM University, Nanded, India
Department of IT, Faculty of Computing and IT, UST, Sana'a, Yemen
{mmghilan, dr.fadlbaalwi, Fwesabi}@gmail.com

ABSTRACT: *In the study of content authentication and tamper detection of digital text documents, there are very limited techniques available for content authentication of text documents using digital watermarking techniques.*

In this paper, we have extended the proposed LNMZW3 algorithm presented in [27] for content authentication and tamper detection of English text documents and abbreviated as ADV-LNMZW3. In the enhanced algorithm , third order and letter-based of Markov model has uses in this paper as soft computing tools in order to English text analyse in order to find the inter-relationships and utilize theses features to generate and detect a watermark in order to identify the status of text document such as authentic, or tampered.

One of the important enhancements has done on LNMZW3 algorithm and presented in this paper is study the performance average of our enhanced algorithm named as ADV-LNMZW3 against different sizes of datasets and compare them with two modern proposed algorithms named LNMZW1 and LNMZW2. The study of dataset size effect also has been examined in this paper against common volumes of insertion, deletion and reorder attacks.

The enhanced algorithm (ADV-LNMZW3) was implemented using PHP Programming language with Net Beans IDE 7.0. Furthermore, the effectiveness and feasibility of our ADV-LNMZW3 algorithm has proved and compared with other recent algorithms with experiments using five datasets of varying lengths and different volumes of attacks.

The experiment showed that the ADV-LNMZW3 algorithm had better performance and robustness, and it is more secure than other algorithms especially in case of insertion and deletion attacks.The effect of document size on watermark robustness was examined. The results showed that the watermark robustness is enhanced with small and medium documents size.

The results showed that our ADV-LNMZW3 algorithm was applicable for all sizes of text document, and it is recommended for tampering detection against small, and medium sizes of text documents. However, the LNMZW1 was found to be applicable under large size of text documents.

Keywords: Digital Watermarking, Watermark Robustness, Markov Model, Probabilistic Patterns, Information Hiding, Content Authentication, Tampering Detection

Received: 21 November 2013, Revised 3 January 2014, Accepted 8 January 2014

1. Introduction

With the increasing use of internet, e-commerce, and other efficient communication technologies, the copyright protection and authentication of digital contents, have gained great importance. Most of these digital contents are in text form such as email, websites, chats, e-commerce, eBooks, news, and short messaging systems/services (SMS) [1].

These text documents may be tempered by malicious attackers, and the modified data can lead to fatal wrong decision and transaction disputes [2].

Content authentication and tamper detection of digital image, audio, and video has been of great interest to the researchers. Recently, copyright protection, content authentication, and tamper detection of text document attracted the interest of researchers. Moreover, during the last decade, the research on text watermarking schemes is mainly focused on issues of copyright protection, but gave less attention on content authentication, integrity verification, and tamper detection [4]. Various techniques have been proposed for copyright protection, authentication, and tamper detection for digital text documents. Digital Watermarking (DWM) techniques are considered as the most powerful solutions to most of these problems.

The digital watermarking techniques have emerged as a solution to breaches of copyright protection, tampering detection, and content authentication of digital media. Researchers have proposed various watermarking algorithms for images, audio, and video. However, watermarking algorithms for text is very inadequate. The reason behind is the difficulty to watermark text.

Traditional text watermarking techniques for tampering detection and text authentication such as format-based, content-based, and image-based have a number of limitations. In other words, they are not applicable under random tampering attacks on all types of texts, and they need to use some transformations or modifications on contents of the text document to embed watermark information within the original text document itself which results in text capacity, quality, meaning, and value degradation.

Text watermarking algorithms such as Word Length Zero-Watermarking, Non-Vowel ASCII Characters Zero-Watermarking, and content based Zero-Watermarking are not applicable under random tampering attack to all types of text documents. These algorithms are restricted to only alphabetical watermarks, which means they are not applicable to specialized texts such as legal documents, web contents, and documents containing financial, accounting, and mathematical notations. All makes these algorithms impractical. Furthermore, text watermarking algorithms using binary text image are not robust against retyping attacks. The text watermarking methods based on semantics are language dependent and do not provide a complete practical solution for all types of texts.

The English text zero watermarking algorithm based on the probabilistic weight of Markov model presented in [24] has a number of limitations in that it is not applicable under all tampering attacks to all type of documents. However, we can make the algorithm more robust, secure, and more efficient by extending the character set and including other mechanisms and various orders of Markov model for English text analysis.

In this paper, the authors present a new zero-watermarking algorithm for content authentication of English text document via Internet. This algorithm uses digital watermarking techniques and utilizes the natural language processing in term of text analysis in order to extract text features as probabilistic patterns based on third order of Markov model and abbreviated here as ADV-ADV-LNMZW3.

The paper is organized as follows. Section 2 provides an overview of the previous work done on text watermarking. The proposed algorithm is described in detail in section 3. Section 4 presents the experimental and comparative results for the various tampering attacks such as insertion, deletion and reordering. Performance of the proposed algorithm is evaluated by multiple text datasets. The last section concludes the paper along with directions for future work.

2. Previous Work

Text watermarking techniques have been proposed and classified by many literatures based on several features and embedding modes of text watermarking. We have examined briefly some traditional classifications of digital watermarking as in literatures. These techniques involve text images, content based, format based, features based, synonym substitution based, and

syntactic structure based, acronym based, noun-verb based, and many others of text watermarking algorithms that depend on various viewpoints [1] [3] [4].

2.1 Format-based Techniques

Text watermarking techniques based on format are layout dependent. In [5], proposed three different embedding methods for text documents which are, line shift coding, word shift coding, and feature coding. In line-shift coding technique, each even line is shifted up or down depending on the bit value in the watermark bits. Mostly, the line is shifted up if the bit is one, otherwise, the line is shifted down. The odd lines are considered as control lines and used at decoding. Similarly, in word-shift coding technique, words are shifted and modify the inter-word spaces to embed the watermark bits. Finally, in the feature coding technique, certain text features such as the pixel of characters, the length of the end lines in characters are altered in a specific way to encode the zeros and ones of watermark bits. Watermark detection process is performed by comparing the original and watermarked document.

2.2 Content-based Techniques

Text watermarking techniques based on content are structure-based natural language dependent [4]. In [6] [14], a syntactic algorithm has been proposed which use syntactic structure of cover text for embedding watermark bits by performed syntactic transformations to syntactic tree diagram taking into account conserving of natural properties of text during watermark embedding process. In [18], a synonym substitution has been proposed to embed watermark by replacing certain words with their synonyms without changing the sense and context of text.

2.3 Binary Image-based Techniques

Text Watermarking techniques of binary image documents depends on traditional image watermarking techniques that based on space domain and transform domain, such as Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), and Least Significant Bit (LSB) [5]. Several formal text watermarking methods have been proposed based on embedding watermark in text image by shifting the words and sentences right or left, or shifting the lines up or down to embed watermark bits as it is mentioned above in section format-based watermarking [5][7].

2.4 Zero-based Techniques

Text watermarking techniques based on Zero-based watermarking are content features dependent. There are several algorithm es that designed for text documents have been proposed in the literatures which are reviewed in this paper [1] [19] [20] and [21].

The first algorithm has been proposed by [19] for tamper detection in plain text documents based on length of words and using digital watermarking and certifying authority techniques. The second algorithm has been proposed by [20] for improvement of text authenticity in which utilizes the contents of text to generate a watermark and this watermark is later extracted to prove the authenticity of text document. The third algorithm has been proposed by [1] for copyright protection of text contents based on occurrence frequency of non-vowel ASCII characters and words. The last algorithm has been proposed by [21] to protect all open textual digital contents from counterfeit in which is insert the watermark image logically in text and extracted it later to prove ownership. In [22], Chinese text zero-watermark algorithm has been proposed based on space model by using the two-dimensional model coordinate of word level and the sentence weights of sentence level.

2.5 Combined-based Techniques

One can say the text is dissimilar image. Thus, language has a distinct and syntactical nature that makes such techniques more difficult to apply. Thus, text should be treated as text instead of an image, and the watermarking process should be performed differently. In [23] A combined method has been proposed for copyright protection that combines the best of both image based text watermarking and language based watermarking techniques.

The above mentioned text watermarking algorithm es are not appropriate to all types of text documents under document size, types and random tampering attacks, and its mechanisms are very essential to embed and extract the watermark in which maybe discovered easily by attackers. On the other hands, these algorithms are not designed specifically to solve problem of authentication and tamper detection of text documents, and are based on making some modifications on original text document to embed added external information in text document and this information can be used later for various purposes such as content authentication, integrity verification, tamper detection, or copyright protection. This paper proposes a novel intelligent algorithm for content authentication and tamper detection of English text documents in which the watermark embedding and

extraction process are performed logically based on text analysis and extract the features of contents by using hidden Markov model in which the original text document is not altered to embed watermark.

3. The Proposed Algorithm

This paper proposes a novel algorithm based on digital watermarking techniques and third order of Markov model. The proposed algorithm named here as ADV-ADV-LNMZW3. In our ADV-LNMZW3algorithm, the original text document is not altered to embed watermark, that means the watermark embedding process is performed logically. The proposed algorithm uses the Markov model of the natural languages that is Markov chains which are used to analyze the contents of English text documents and extract the probability features of interrelationships between these contents as probabilistic patterns based on letter mechanism and third order of Markov model which are utilized to generate the watermark key that is embedded logically within the original text document or stored in the watermark database. This watermark key can be used later and matched with watermark generated from attacked document for identifying any tampering that may happen to the document and authenticating its content. This process illustrated in figure 1.

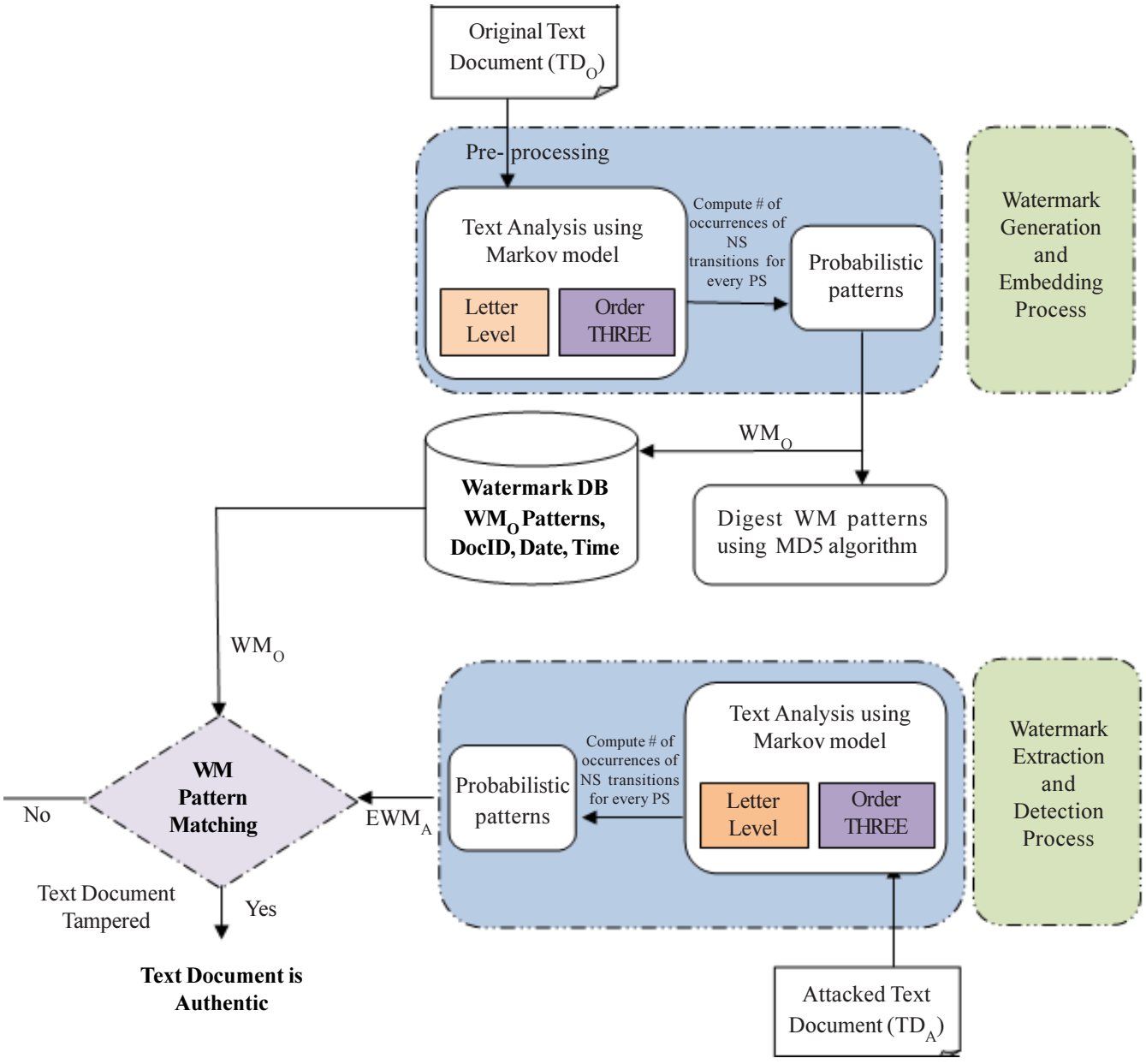

Figure 1. Watermark generation and detection processes

Before we explain the watermark generation and extraction processes in the next subsection, we present a preliminary mathematical description of Markov models of natural language text analysis.

3.1 ADV-LNMZW3 Algorithm for Text Analysis

This subsection explains how to model English text using our ADV-LNMZW3 algorithm with 3-gram order of Markov model. For example, we have an English sentence as shown in Figure 2.

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

Figure 2. Sample of English text

When we are using 3-gram, each sequence of three unique consecutive letters in the given text is a state by itself. The process of Markov transitions from state to state as the text is read.

For instance, When we are using 3-gram order of Markov model named here as ADV-ADV-LNMZW3, and as the above sample text is processed, the system generates a list of all unique letter-level of 3-gram order as a possible states as the following:

"*the*", "*he*", "*eq*", "*qu*", "*qui*", "*uic*", "*ick*", "*ck*", "*kb*", "*kbr*", "*bro*", "*row*", "*own*", "*wnf*", "*info*",…, "*dea*"

As this sample text is processed, and if the Markov chain is currently at "*the q*" state, the possible transitions for each a state that could come next are:

"the → **,** the → **q,** the → **u,** the → **i,** the → **c,** the → **k,** the → **,** the → **b,** the → **r,** the → **o,** the → **w,** the → **n,**…, the → **d"**

The previous scenario of our ADV-LNMZW3 algorithm show that the algorithm used to build possible states and transitions of the Markov chain matrix *M* [i] [j]. For instance, the sample English text shown above contains 94 character with spaces.

After being processed by the system, and represented in the Markov chains by applying the equations (1), (2), (3) and (4), we get for 49 unique states U_s and we can get for 21600 possible states P_s as sets of three consecutive letters, 61 possible transitions for each state T_s, and 2989 possible transitions for all states P_t.

$$P_s = (n-3)^3 \tag{1}$$

$$U_t = (n - R_s - 3)^3 \tag{2}$$

$$T_s = \sum_{i=1}^{i=n} U_s(i) + 3 + 3\,(3-1) \tag{3}$$

$$P_T = P_s * T_s \tag{4}$$

Where,

- n : is the total count of all character sets in the given text document.
- P_S : refers to the total of all possible states.
- U_S : is the total of all unique states (letter sets) within the given text document.
- R_S : is the count of all repeated states within Markov chain matrix.
- T_S : refers to all possible transitions for each state.
- P_T : refers to the total of possible transitions for all states.

As a result of 3-gram order of Markov model based on letter level to analyze the above sentence, we obtain Figure 3.

We can see from Figure 3 above that there are 49 unique states, and 91 actual transitions for a 3-gram order of Markov model

State	ID State	Transitions
1	(“*br*”):	([‘0’, ‘0’, ‘0’],
2	(“*bro*”):	[‘W’, ‘W’, ‘W’],
3	(“*dea*”):	[‘d’],
4	(“*ick*”):	[‘’],
5	(“*jum*”):	[‘p’, ‘p’] ,
...	...	...
...	...	...
...	...	...
48	(“*xj*”):	[‘u’] ,
49	(“*xw*”):	[‘h’, ‘h’] ,

Figure 3. Sample text states and transitions of ADV-LNMZW3

after processed to analyzing the above given sentence.Now if we consider state “*jum*” from the Figure 3 above, the next state transitions are “*p*”, “*p*”, which indicates that the transition “*p*” occurs twice.

Next we present a simple method to build the states and the Markov transition matrix $m[i,j]$which is the most basic part of text analysis using Markov model.

In the proposed algorithm ADV-LNMZW3, the text considered is not limited to alphabetic characters, but includes spaces, numbers, and special characters such as [, . ; : – ? !]. The total number of states depends on size of given text document (n) which equal $n-2$, and the total number of states is 61,these are [English letters = 26, space letter = 1, Integer numbers from 0 to 9 = 10, specific symbols such as . ‘ “ , ; : ? ! / ∴ ≅ ∃ & % * + – = >< ( ) [ ] = 24]. The entry will be used to keep track of the number of times that the character set of the text is followed by the character of the text. For, where is the length of the text document – 1”, let x be the i^{th} character set in the text and y be the $(i+1)^{st}$ character in the text. Then increment $M[x, y]$. Now the matrix M contains the counts of all transitions. Next we turn these counts into probabilities as follows, for each i from 1 to 61, sum the entries on the i^{th} row, i.e., let counter $[i] = M[i, 1] + M[i, 2] + M[i, 3] + ... + M[i, 61]$.

Now define $P[i,j] = M[i, j] /$ counter $[i]$ for all pairs i, j. This just gives a matrix of probabilities. In other words, now $P[i, j]$ is the probability of making a transition from character set i to character j. Hence a matrix of probabilities that describes a Markov model of order three for the given text is obtained.

3.2 Watermark Generation and Embedding Algorithm

The watermark generation and embedding algorithm requires the original text document as input, then as a pre-processing step it is required to perform conversion of capital letters to small letters and to remove all spaces within the text document. A watermark pattern is generated as the output of this algorithm. This watermark is then stored in watermark database along with the original text document, document identity, author name, current date and time.

This stage includes two main processes which are watermark generation and watermark embedding. Watermark generation from the original text document and embed it logically within the original watermark will be done by the embedding algorithm.

In this proposed watermark generation algorithm, the original text document (T) is to be provided by the author. Then text analysis process should be done using Markov model to compute the number of occurrences of the next state transitions (ns) for every present state (ps). Markov model of order three. A Matrix of transition probabilities that represents the number of occurrences of transition from a state to another is constructed according to the procedure explained in previous section (3.1) and can be computed by equation (5).

$$M[ps][ns] = P[i][j], \text{for } i, j = 1, 2, ., n \quad (5)$$

Where,

n: is the total number of states

i: refers to PS "*the present state*".
j: refers to NS "*the next state*".

• *P* [*i*, *j*]: is the probability of making a transition from character set *i* to character *j*.

After performing the text analysis and extracting the probability features, the watermark is obtained by identifying all the nonzero values in the above matrix. These nonzero values are sequentially concatenated to generate a watermark patterns, denoted by WMP_O as given by equation (6) and presented in figure 4.

$$WMP_O \&= M[ps][ns],\ for\ i,.\ j = nonzero\ values\ in\ the\ Markov\ matrix \qquad (6)$$

States	Transitions	DWM patterns
' br'	3	3
'bro'	3	3
'dea'	1	1
'ick'	1	1
'jum'	2	2
... ...		
'x j'	1	1
'x w'	2	2

This watermark is then generated as sequential patterns as presented in figure 5, and stored in a watermark database along with the original text document, document identity, author name, current date and time.

DWM Patterns Of LNMZW3
3 – 3 – 1 – 1 – 2 – .. – 1 – 2

Figure 5. The original watermark patterns of the given text sample based on ADV-LNMZW3

After watermark generation as sequential patterns, an MD5 message digest is generated for obtaining a secure and compact form of the watermark, notational as given by equation (7) and presented in figure 6.

$$DWM = MD5(WMP_O) \qquad (7)$$

DWM Patterns Of LNMZW3
1fd9140249a0d0ae1d27a7c5e7ab6e

Figure 6. The original watermark patterns after MD5 digesting

The proposed watermark generation and embedding algorithm of our ADV-LNMZW3 algorithm is presented formally in figure 7.

3.3 Watermark Extraction and Detection Algorithm

The watermark detection algorithm is on the base of zero-watermark, so before detection for attacked text document TD_A, the proposed algorithm still need to generate the attacked watermark patterns2 . When received the watermark patterns2, the matching rate of patterns2 and watermark distortion are calculated in order to determine tampering detection and content authentication.

This stage includes two main processes which are watermark extraction and detection. Extracting the watermark from the received attacked text document and matching it with the original watermark will be done by the detection algorithm.

DWM Generation Algorithm (A Zero Text DWM based on LNMZW3 approach)

- **Input:** Original Text Document (TD_o)
- **Output:** WMP_o VALUE, AND States and Transactions Matrix [n – 3] [61], // n: is the length of given text

1. Read the Original Text document (TD_o) and performs Pre - processing for it.

2. loops ps =1to n – 2, //Build the states of Markov matrix based on text size

- loops ns = 1 to 61, // Build the fixed 61 possibles transitions for all states of Marko matrix
  - M [ps] [ns] = Total Number of Transitions [ps] [ns]// compute the total frequencies of transitions for every state.

3. loop i =1 to n = 3,//generate original watermark

- loop j = 1 to 61,
  - IF M [i] [j] I = O // states that have transitions
  - WMP_o & = M [i] [j]

4. DMM = MDS (WMP_o) // Digest original watermark using MDS algorithm

5. Output = WMP_o, DWM

***WMP*$_o$:** Original watermark, ***n*:** is the total number of sets ***TD*$_o$:** Orignal text document array

***M*:** states and transitions matrix, ***DWM*:**Digested watermark, ***MDS*:** Hash algorithm,

***ps*:** The person state, ***ns*:** The next state.

Figure 7. Watermark generation and embedding algorithm using our ADV-LNMZW3 algorithm

The proposed watermark extraction algorithm takes the attacked text document, and performs the same watermark generation algorithm to obtain the watermark pattern for the attacked text document.

After extracting the attacked watermark pattern, the watermark detection is performed in two steps,

• Primary matching is performed on the whole watermark pattern of the original document WMP_O, and the attacked document WMP_A. If these two patterns are found the same, then the text document will be called authentic text without tampering. If the primary matching is unsuccessful, the text document will be called not authentic and tampering occurred, then we proceed to the next step.

• Secondary matching is performed by comparing the components associated with each state of the overall pattern. Which compares the extracted watermark pattern for each state with equivalent transition of original watermark pattern. This process can be described by the following mathematical equations (8), and (9).

$$PMR_T(i,j) = \left| \frac{WMP_O[i],[j] - WMP_O[i],[j] - WMP_A[i],[j]}{WMP_O[i],[j]} \right| \quad \text{for all } i,j,\ 1 \geq PMR_T(i,j) \geq 0 \tag{8}$$

$$PMR_S(i) = \left| \frac{\sum_{j=1}^{N} PMR_T(i,j))}{\text{Total state pattern}} \right| \quad \text{for all } i,j,\ 1 \geq PMR_s(i,j) \geq 0 \tag{9}$$

Where,

• ***N*:** is the number of non-zero elements in the Markov chain matrix.

Finally, the PMR is calculated by equation (10), which represents the pattern matching rate between the original and attacked text document.

$$PMR = \frac{\Sigma_{j=1}^{n} PMRS(i)}{n} \quad (10)$$

Where,

***n*:** is the total number of states. This process is illustrated in figure 8.

States	Original WM patterns	Extracted WM patterns	Destroyed WM patterns	Primary Matching rate	Primary Matching rate of transition level PMRT (*i*, *j*)		Primary Matching rate of PMRT (*i*, *j*)
					TP1	TP2	
'br'	3	2	2	-	0.6667	-	0.6667
'bro'	3	2	2	-	0.6667	-	0.6667
'dea'	1	1	1	1	-	-	1
'ick'	1	1	1	1	-	-	1
'jum'	2	2	2	1	-	-	1
...	...	...	...	...	...	...	-
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
'xj'	2	1	1	1	-	-	1
'xw'	2	1	1	-	0.5	-	0.5
Robustness (PMR)							= 58.0006 / 72 = 0.8056

Figure 8. Watermark detection process using our ADV-LNMZW3 algorithm

The watermark distortion rate refers to tampering amount occurred by attacks on contents of attacked text document, this value represent in *WDR* which we can get for it by equation (11):

$$WDR = 1 - PMR \quad (11)$$

The detection algorithm is illustrated in figure 9.

4. Experimental Setup, Results and Discussion

4.1 Introduction

After we experimented and implemented algorithms of our ADV-LNMZW3 algorithm, we evaluated the accuracy of tampering detection, and compared them with the existing two previous orders of letter level of Markov model based on the criteria of the authentication features, and under the most common nature and volume of possible tampering attacks, which are insertion, deletion, and reorder attacks in multiple random locations of the experimental datasets.

This subsection introduces results discussion and evaluation of our ADV-LNMZW3 algorithm as a contribution of this paper. A methodological description of the process is introduced, as well. Evaluation the performance of our ADV-LNMZW3 algorithm, different scenarios of tampering attacks on different dataset sizes, was conducted for different scenarios of the attack volumes and attack types. Then, the performance average was found and compared to current existing algorithms which are LZW1 and LZW2 and have presented in [25] and [26]. After that, the best performance of which algorithm was found attacks, which are insertion, deletion, and reorder attacks in multiple random locations of the experimental datasets.

This subsection introduces results discussion and evaluation of our ADV-LNMZW3 algorithm as a contribution of this paper. A methodological description of the process is introduced, as well.

• Evaluation the performance of our ADV-LNMZW3 algorithm , different scenarios of tampering attacks on different dataset sizes, was conducted for different scenarios of the attack volumes and attack types. Then, the performance average was found and compared to current existing algorithms which are LZW1 and LZW2 and have presented in [25] and [26]. After that, the best performance of which algorithm was found.

DWM Extraction and Detection Algorithm (A Zero Text DWM based on LNMZW3 approach)

- Input: Original Text Document, Attacked Text Document

- Output: WMP_O, WM_A, WMP_A. PMR, WDR, Attacked States and Transitions Matrix [n – 3][61].

1. Read WMo or Original Text (TD_O) and Attacked Text (TD_A) documents and performs Pre-processing for them.

1. Loop ps = 1 to n – 3, // Build the states of Markov chain Matrix

- Loop ns = 1 to 61, // Build the transitions for each state in Markov chain Matrix
  - M' [ps] [ns] = Total Number of Transition [ps] [ns] // compute the total frequencies of transitions for every state

2. Loop i = 1 to n –3, // Extract the embedded watermark

- Loop j = 1 to 61,
  - IF M'[i][j] != 0 // states that have transitions
  - WMP_A &= M'[i] [j]

3. Output WMP_O, WMP_A

4. IF $WMP_A = WMP_O$

- Print "*Document is authentic and no tampering occurred*"
  - PMR = 1
- Else
  - Print "*Document is not authentic and tampering occurred*"
  - For i = 1 to n – 3// Extract transition patterns and match each of them with original transition patterns
    - For j = 1 to 61
      - IF WMP_O[i] [j] != 0
        - patternCount +=1

$$PMR_T(i,j) = \left| \frac{WMP_O[i],[j] - WMP_O[i],[j] - WMP_A[i],[j]}{WMP_O[i],[j]} \right|$$

        - transPMRTotal += PMR_T
      - Else
        - IF WMP_A[i] [j] != 0
          - patternCount += WMP_A [i] [j])

$$\text{statePMR}[i] = \frac{\text{transPMRTotal}[i][j]}{\text{patternCount}[i]}$$

        - PMR_S += statePMR[i]
- Totalpattern += patternCount

5. $$\text{PMR} = \frac{PMR_S}{\text{Total pattern}}$$

6. WDR = 1 – PMR

***WMP_O*:** Original watermark, ***WMP_A*:** Attacked watermark, ***M'*:** Attacked states and transitions matrix, ***TD_A*:** Attacked text document array, ***ps*:** the present state, ns: the next state, ***PMR_T*:** Transition patterns matching rate, ***PMR_S*:** State patterns matching rate, ***PMR*:** Watermark patterns matching rate, ***WDR*:** Watermark distortion rate.

Figure 9. Watermark extraction and detection algorithm using our ADV-LNMZW3 algorithm

Results study of dataset size effect on watermark robustness against our ADV-LNMZW3algorithm and the previous algorithms (LZW1 and LZW2) for different scenarios of attack volumes and attack types. Also, the max of maximum watermark robustness was found and compared with LZW1 and LZW2 algorithms. In addition, we found the dataset size that led to the

best robustness of which algorithm.

Each step mentioned above is presented in a separate subsection in this paper.

4.2 Experimental Parameters and Setup

In order to test the proposed algorithm and compare it with other algorithms, we conducted a series of simulation experiments. The experimental environment (CPU: Intel Core™i5 M480/2.67 GHz, RAM: 8.0 GB, Windows 7, and PHP Programming language with Net Beans IDE 7.0.) is explained below:

First, we depends on performance of third order algorithm as presented in [27], and present the watermark robustness of our ADV-LNMZW3 algorithm against different sizes of Standard English text datasets.

Then, the effect of data set size has been studied on our ADV-LNMZW3 algorithm with two previous developed algorithms and compared their results.

The experiment datasets namely SST4, SST2, MST5, MST2, and LST4. The datasets are a large collection of writings of a specific kind called "*Corpora*", and a statistically representative sample of natural language texts referred to as "*Corpus*" and found in [11]. The datasets were used for all attack scenarios of the different common volumes and types. In other words, the datasets were used with 5%, 10%, 20%, and 50% of the attack volumes used for the insertion, detection, and reorder attack types determined in [27]. The details of our datasets and attacks used are shown in Table 1.

Dataset name	**Dataset size**	**Attacks Types and Volumes**		
		Insertion	**Deletion**	**Reorder**
[SST4]	179			
[SST2]	421	5%, 10%,	5%, 10%	5%, 10%,
[MST5]	469	20%, 50%	20%, 10%,	20%, 50%
[MST2]	559			
[LST4]	2018			

Table 1. Dataset names and sizes, attacked types and volumes

4.3 Performance Evaluation

In this subsection, we present the results study and analysis of performance average for our proposed algorithm named Adv-LNMZW3and compare it with previous two proposed algorithms named as (LNMZW1 and LNMZW2) against different dataset sizes as Table 1 shows and under common scenarios of insertion, deletion and reorder attacks. Common attacks scenarios are (5%, 10%, 20% and 50%) and categorized as small, medium and large attack volumes. Then, we find the best performance average of which algorithm that has the best performance.

Attack	**(LNMZW1)**	**(LNMZW2)**	**(ADV-LNMZW3)**
Insertion	79.37	80.14	81.68
Deletion	78.85	80.62	85.52
Reorder	97.92	93.19	88.52

Table 2 shows the performance averages for our previous proposed algorithm named ADV-LNMZW3 algorithm as presented in [27] and previous algorithms (LNMZW1 and LNMZW2) against all dataset sizes and all scenarios of reorder with all attack volumes.

As shown in Table 2 and Figure 10, in general all order levels of letter mechanism algorithms of Markov model, i.e., LMNZW 1, 2, and 3 perform high performance under reorder attack, and especially, LNMZW1 algorithm perform higher performance than LNMZW2 and ADV-LNMZW3 algorithms. On the contrary, our ADV-LNMZW3 algorithm perform better performance than LNMZW1 and LNMZW2 algorithms under insertion and deletion attacks.

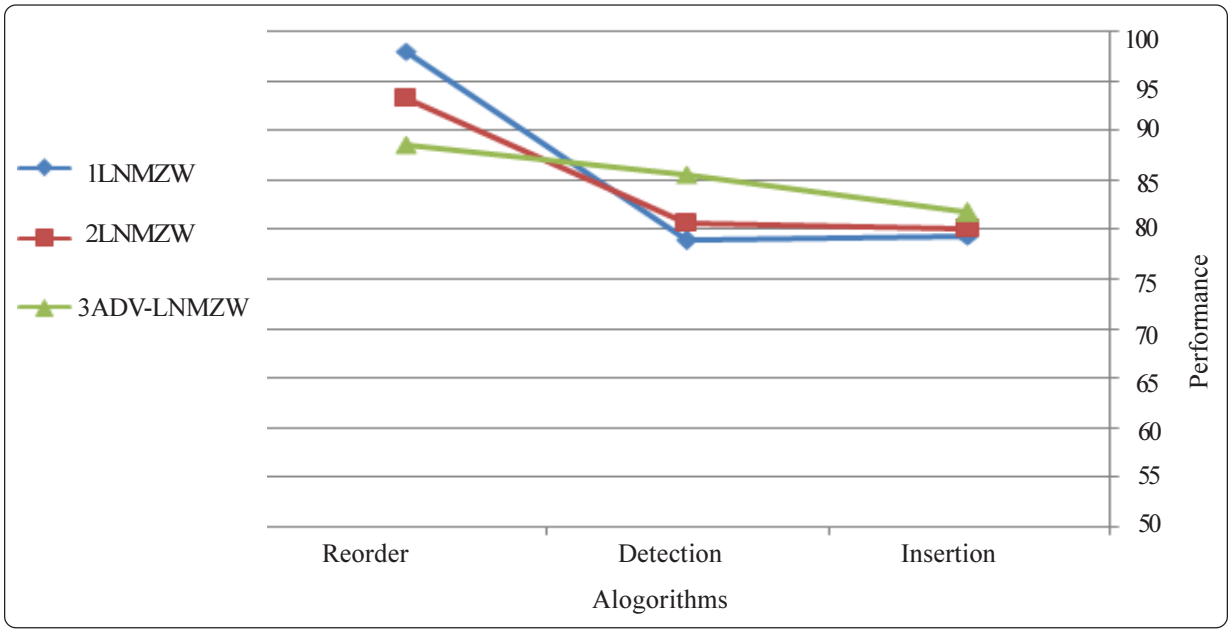

Figure 10. Average of performance averages of our ADV-LNMZW3 algorithm with LNMZW1 and LNMZW2 algorithms under all attacks

4.4 Results Study of Document Sizes Effects

In this subsection, we present an evaluation of English text documents size effects on watermark robustness against our ADV-LNMZW3 algorithm and make comparison with previous tow proposed algorithms (LNMZW1 and LNMZW2) under common scenarios of insertion, deletion and reorder attacks. Effect of document size was examined under insertion attack with four scenarios of the attack volume, i.e., 5%, 10%, 20%, and 50%.

To measure the effects of document size on watermark robustness, we depends on different dataset sizes as shown above in Table 1.

• Document Sizes Effects under Insertion Attack

Table 3 shows the average of maximum values of watermark robustness for our proposed ADV-LNMZW3 algorithm and make comparison with previous (LNMZW1 and LNMZW2) algorithms against all dataset sizes and all scenarios of insertion attack.

In terms of dataset size effect on robustness value under all scenarios of insertion attack as shown by Table 3 and Figure 11, The middle dataset named [MST2, 559] has the best robustness with average value 82.87% with our proposed algorithm (ADV-LNMZW3). In general, the results show that a slight effect is detected in all different sizes of datasets. Further, the results show that the watermark robustness gets more positive effects with large documents. This means that the robustness value decreases with small documents and increases with large documents.

Dataset Name and size	(LNMZW1)	(LNMZW2)	(ADV-LNMZW3)
[SST4, 179]	79.73	78.58	81.46
[SST2, 421]	78.00	79.74	81.43
[MST5, 469]	78.07	78.41	80.09
[MST2, 559]	82.15	82.62	82.87
[LST4, 2018]	78.92	81.36	82.58

Table 3. Comparison of dataset size effect on watermark robustness under all volumes of insertion attack

• Document Sizes Effects under Deletion Attack

Table 4 shows the average of maximum values of watermark robustness for our proposed ADV-LNMZW3 algorithm and make

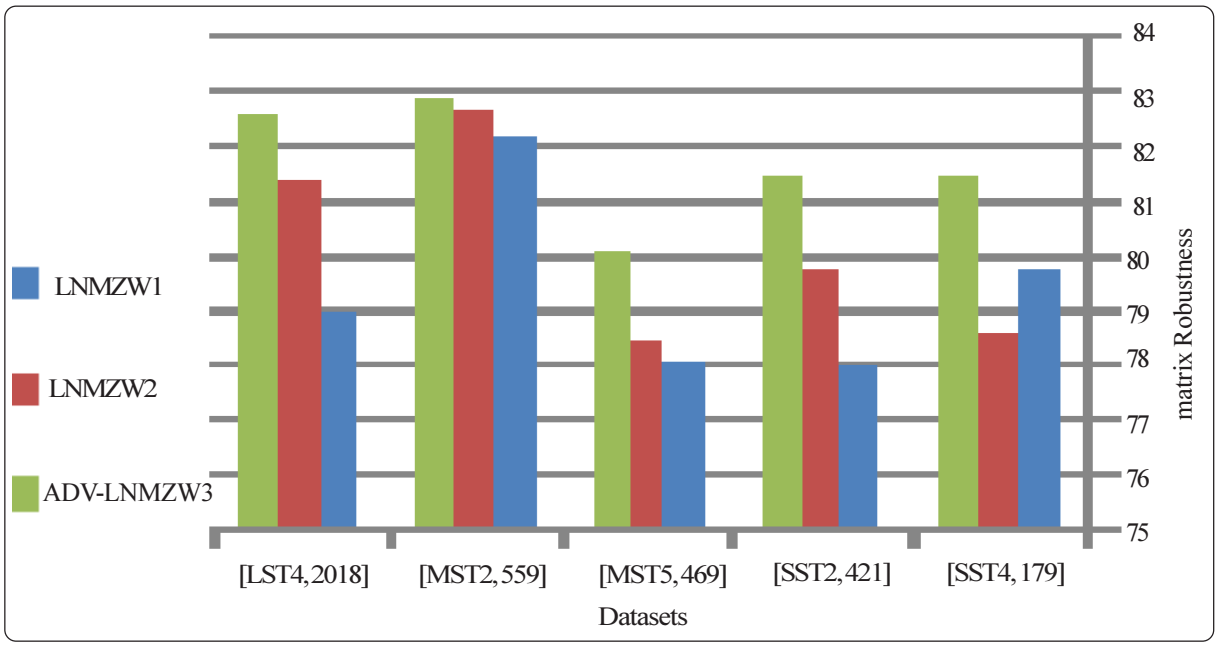

Figure 11. Comparison of dataset size effect on watermark robustness under all volumes of insertion attack

comparison with previous (LNMZW1 and LNMZW2) algorithms against all dataset sizes and all scenarios of deletion attack.

Dataset Name and Size	(LNMZW1)	(LNMZW2)	(ADV-LNMZW3)
[SST4, 179]	77.46	81.80	87.43
[SST2, 421]	78.08	80.33	85.94
[MST5, 469]	79.97	80.98	85.82
[MST2, 559]	79.11	80.22	85.77
[LST4, 2018]	79.65	79.77	82.61

Table 4. Comparison of dataset size effect on watermark robustness under all volumes of deletion attack

In terms of dataset size effect on robustness value under all scenarios of deletion attack as shown by Table 4 and Figure 12, The best average of robustness value of all datasets is found by the smallest dataset named [SST4, 179] with an average value 87.43%. The results show that there is clear effect on the watermark robustness especially with small and large datasets sizes. In general, in case of both ADV-LNMZW3 and LNMZW2 algorithms, the robustness value increases with small documents and decreases with large documents. And vice versa in the case of LNMZW1 algorithm in which the robustness value increases with large documents and decreases with small documents.

• Document Sizes Effects under Reorder Attack

Table 5 shows the average of maximum values of watermark robustness for our proposed ADV-LNMZW3 algorithm and make comparison with previous (LNMZW1 and LNMZW2) algorithms against all dataset sizes and all scenarios of reorder attack.

In terms of dataset size effect on robustness value under all scenarios of reorder attack as shown by Table 5 and Figure 13, The best average of robustness value of all datasets is found with small dataset. In general, the results show that the robustness value enhances when Ngram order of Markov model decreased and vice versa when Ngram order is increased.

As a comparison in terms of general effect of the dataset size on Ngram order of Markov model, as shown in Table 6 and Figure 15, we can say that our ADV-LNMZW3 algorithm led the best robustness value in case of small dataset size, which means ADV-LNMZW3 is recommended with small documents as shown with [SST4, 179] with robustness average 85.48% and

[SST2, 421] with robustness average 86.81%. However, LNMZW1 algorithm is recommended with mid and large documents as shown with [MST5, 469], [MST2, 559] and [LST4, 2018] with values 85.35, 86.16 and 85.15 Respectively.

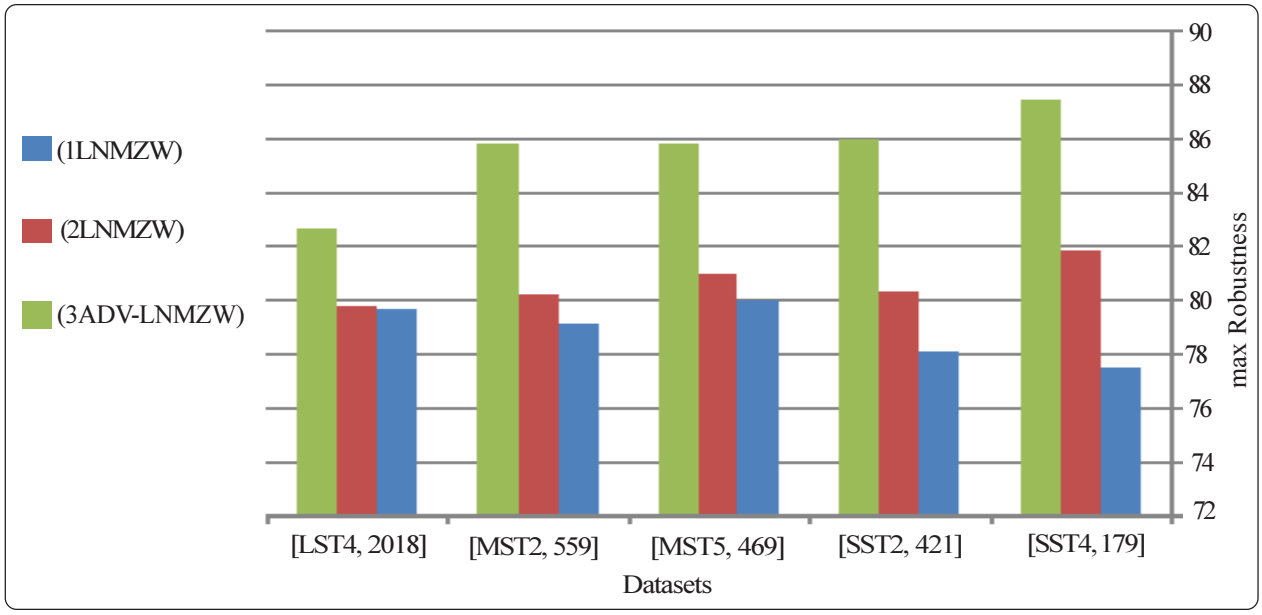

Figure 12. Comparison of dataset size effect on watermark robustness under all volumes of deletion attack

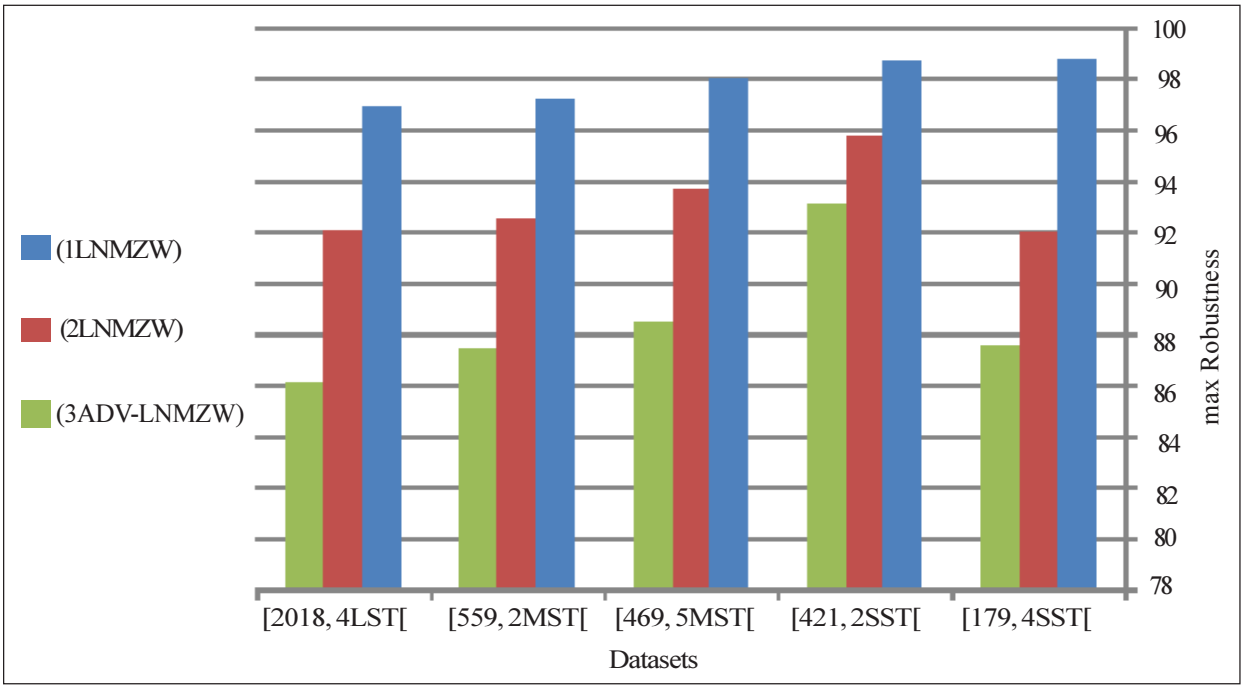

Figure 13. Comparison of dataset size effect on watermark robustness under all volumes of reorder attack

5. Conclusion

Based on natural language processing techniques (NLP), we have extend the proposed LNMZW3 algorithm presented in [27]. Third order of Markov model has uses in this paper as NLP technique to analyse English text and find the inter-relationship of contents of the text documents according to the probabilistic patterns of states and transitions to finally generate and detect a watermark.

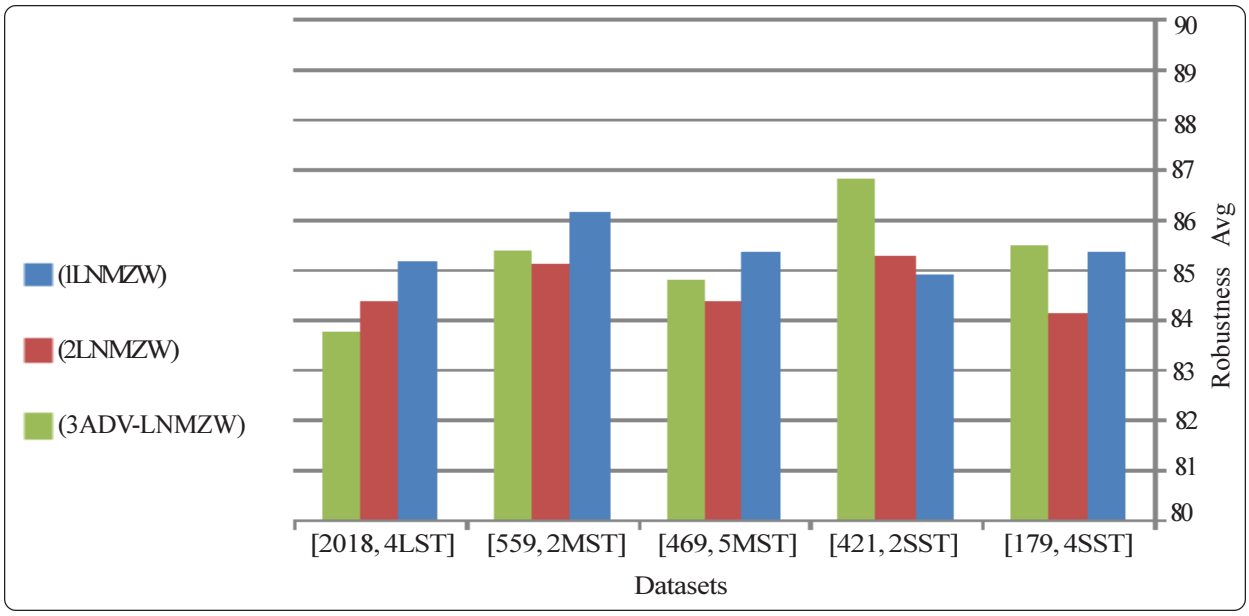

Figure 14. Comparison of dataset size effect on watermark robustness under all volumes of all attacks

Dataset Name and Size	(LNMZW1)	(LNMZW2)	(ADV-LNMZW3)
[SST4, 179]	85.33	84.12	85.48
[SST2, 421]	84.92	85.26	86.81
[MST5, 469]	85.35	84.38	84.79
[MST2, 559]	86.16	85.12	85.35
[LST4, 2018]	85.15	84.39	83.76

Table 6. Comparison of dataset size effect on watermark robustness under all volumes of all attacks

One of the important enhancements has done on LNMZW3 algorithm and presented in this paper is study the performance average of our enhanced algorithm named as ADV-LNMZW3 against different sizes of datasets and compare them with two modern proposed algorithms named LNMZW1 and LNMZW2. The study of dataset size effect also has been examined in this paper against common volumes of insertion, deletion and reorder attacks.

The experiment showed that the ADV-LNMZW3 algorithm had better performance and robustness, and it is more secure than other algorithms especially in case of insertion and deletion attacks. The effect of document size on watermark robustness was examined. The results showed that the watermark robustness is enhanced with small and medium documents size.

The results showed that our ADV-LNMZW3 algorithm was applicable for all sizes of text document, and it is recommended for tampering detection against small, and medium sizes of text documents. However, the LNMZW1 was found to be applicable under large size of text documents.

References

[1] JaliI, Z., Hamza, A., Shahidm, S., Arif, M., Mirza, A. (2010). A Zero Text Watermarking Algorithm based on Non-Vowel ASCII Characters. International Conference on Educational and Information Technology (ICET), IEEE.

[2] Suhail M. A., (2008). Digital Watermarking for Protection of Intellectual Property. A Book Published by University of Bradford, UK,.

[3] Robert, L., Science, C., (2009)Government Arts, *A Study on Digital Watermarking Techniques*. *International Journal of Recent Trends in Engineering*, 1 (2) 223-225.

[4] Zhou, X., Wang, S., Xiong, S. (2009). Security Theory and Attack Analysis for Text Watermarking. International Conference on E-Business and Information System Security, IEEE, p. 1-6.

[5] Brassil,T., Low, S., Maxemchuk, N. F.,(1999). Copyright Protection for the Electronic Distribution of Text Documents. *In*: Proceedings of the IEEE, 87 (7) 1181-1196, July.

[6] Atallah, M., Raskin, V., Crogan, M. C., Hempelmann, C. F., Kerschbaum, F., Mohamed, D., Naik, S. (2001) *Natural language watermarking: Design, analysis, andimplementation*. *In*: Proceedings of the a Fourth Hiding Workshop, LNCS 2137, p. 25-27

[7] Maxemchuk, N. F., Low, S. (1997). Marking Text Documents. *In*: Proceedings of the IEEE International Conference on Image Processing, Washington, DC, Oct 26-29, p. 13- 16.

[8] Huang, D., Yan, H., (2001). Interword distance changes represented by sine waves for watermarking text images. *IEEE Trans. Circuits and Systems for Video Technology*, 11 (12)1237 1245.

[9] Maxemchuk, N., Low, S. (1998). Performance Comparison of Two Text Marking Methods. *IEEE Journal of Selected Areas in Communications* (JSAC), 16 (4) 561-572.

[10] Low, S., Maxemchuk, N. (2000). Capacity of Text Marking Channel. *IEEE Signal Processing Letters*, 7 (12) 345 -347.

[11] Kim, M. (2008). Text Watermarking by Syntactic Analysis. 12th WSEAS International Conference on Computers, Heraklion, Greece.

[12] Meral, H., Sankur, B., Sumru, A.,Güngör, T., Sevinç, E. (2009). Natural language watermarking via morphosyntactic alterations. Computer Speech and Language, 23, p. 107-125.

[13] Jalil, Z., Mirza, A. (2009). A Review of Digital Watermarking Techniques for Text Documents. International Conference on Information and Multimedia Technology, p. 230-234, IEEE.

[14] AtaIIah, M., McDonough, C., Nirenburg, S., Raskin, V. (2000). Natural Language Processing for Information Assurance and Security: An Overview and Implementations. *In*: Proceedings 9th ACM/SIGSAC New Security Paradigms Workshop, p. 5 1-65.

[15] Meral, H., Sevinc,E., Unkar, E., Sankur, B., Ozsoy, A., Gungor, T. (2000). Syntactic tools for text watermarking. *In*: Proc. of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents, p. 65050X-65050X-12.

[16] Vybornova, O., Macq.B., (2007). Natural Language Watermarking and Robust Hashing Based on Presuppositional Analysis. IEEE International Conference on Information Reuse and Integration, IEEE.

[17] Tallah, M., Raskin, V., Hempelmann, C. (2002). Language watermarking and tamperproofing. *In*: Proc. of al.. Natural 5th International Information Hiding Workshop, Noordwijkerhout, Netherlands, p.196-212.

[18] Topkara, U., Topkara, M., Atallah, M. J. The Hiding Virtues of Ambiguity: Quantifiably Resilient Watermarking of Natural Language Text through Synonym Substitutions. *In*: Proceedings of ACM Multimedia and Security Conference, Geneva.

[19] Jalil, Z., Mirza, A., Jabeen, H. (2010). Word Length Based Zero-Watermarking Algorithm for Tamper Detection in Text Documents. 2nd International Conference on Computer Engineering and Technology, p. 378-382, IEEE10.

[20] Jalil, Z., Mirza, A., Sabir, M. (2010). Content based Zero-Watermarking Algorithm for Authentication of Text Documents. (IJCSIS) *International Journal of Computer Science and Information Security*, 7 (2).

[21] Jalil, Z., Mirza, A., Iqbal, T. (2010). A Zero-Watermarking Algorithm for Text Documents based on Structural Components. p. 1-5, IEEE.

[22] Yingjie, M., Liming, G., Xianlong,W., Tao, G. (2011). Chinese Text Zero-Watermark Based on Space Model. *In*: Proceedings of 13rd International Workshop on Intelligent Systems and Applications, p. 1-5, IEEE.

[23] Yingjie, M., Liming, G., Xianlong, W., Tao, G. (2011). Chinese Text Zero-Watermark Based on Space Model. *In*: Proceedings of Ranganathan, S., Johnsha, A., Kathirvel, K., Kumar, M. (2010). *Combined Text Watermarking*. *International Journal of Computer Science and Information Technologies*, 1 (5) 414-416.

[24] Yingjie, M., Liming, G., Xianlong,W., Tao, G. (2011). Chinese Text Zero-Watermark Based on Space Model. *In*: Proceedings of Fahd N. Al-Wesabi, Adnan Alsakaf, Kulkarni.,Vasantrao, U. (2012). A Zero Text Watermarking Algorithm based on the Probabilistic weights for Content Authentication of Text Documents, *In*: Proc. On *International Journal of Computer Applications* (IJCA), U.S.A, p. 388 - 393.

[25] Yingjie, M., Liming, G., Xianlong,W., Tao, G. (2011). Chinese Text Zero-Watermark Based on Space Model. *In*: Proceedings of Fahd N. Al-Wesabi, Adnan Alsakaf, Kulkarni., Vasantrao, U. (2013). A Zero Text Watermarking Algorithm Based on the Probabilistic Patterns for Content Authentication of Text Documents, *International Journal of Computer Engineering & Technology* (IJCET), India, 4 (1) 284 – 300.

[26] Yingjie, M., Liming, G., Xianlong,W., Tao, G. (2011). *Chinese Text Zero-Watermark Based on Space Model*. *In*: Proceedings of Fahd N. Al-Wesabi, Adnan Alsakaf, Kulkarni., Vasantrao,U. (2012). English Text Zero-Watermark Based on Markov Model of Letter Level Order Two, *Journal of Intelligent Computing*, UK, 3 (4)137-155.

[27] Yingjie, M., Liming, G., Xianlong,W., Tao, G. (2011). *Chinese Text Zero-Watermark Based on Space Model*. *In*: Proceedings of Fadl M. Ba-Alwi, Mokhtar M. Ghilan and Fahd N. Al-Wesabi. (2014). Content Authentication of English Text via Internet using Zero Watermarking Technique and Markov Model, *International Journal of Applied Information Systems* (IJAIS), Foundation of Computer Science FCS, New York, USA, 7 (1), April .