Computers, Materials & Continua
DOI:10.32604/cmc.2022.023684
Article

Tech Science Press

Digital Watermarking Scheme for Securing Textual Database Using Histogram Shifting Model

Khalid A. El Drandaly[1], Walid Khedr[1], Islam S. Mohamed[1] and Ayman Mohamed Mostafa[2,*]

[1]Faculty of Computers and Informatics, Zagazig University, Zagazig, 44519, Egypt
[2]College of Computer and Information Sciences, Jouf University, Sakaka, 72314, Saudi Arabia
*Corresponding Author: Ayman Mohamed Mostafa. Email: amhassane@ju.edu.sa
Received: 17 September 2021; Accepted: 10 November 2021

Abstract: Information security is one of the most important methods of protecting the confidentiality and privacy of internet users. The greater the volume of data, the more the need to increase the security methods for protecting data from intruders. This task can be challenging for researchers in terms of managing enormous data and maintaining their safety and effectiveness. Protection of digital content is a major issue in maintaining the privacy and secrecy of data. Toward this end, digital watermarking is based on the concept of information security through the insertion and detection of an embedded watermark in an efficient manner. Recent methodologies in the research on digital watermarking result in data distortion after embedding the watermark. This scenario can reduce the efficiency of detecting watermarks as well as violate data confidentiality. This study adapts a robust reversible histogram shifting (HS) technique for textual data in relational databases. Furthermore, the study presents a digital watermarking scheme intended for document copyright protection and proof of ownership. The major advantage of the proposed watermarking scheme is that it can protect digital data and preserve data quality. To the best of our knowledge, this research methodology is promising for use in the HS expansion model for watermarking data on non-numeric databases. In addition, the results showed that despite exposing the database to an insertion attacks at 50% and 75% of the watermark, the proposed algorithm can recover more than half of the embedded watermark in all addition and alteration attacks cases. As a result, the watermark information can be retained and restored completely.

Keywords: Information security; digital watermarking; information hiding; histogram shifting technique

1 Introduction

A relational database is a major tool for storing and managing data from information systems, which should be protected from vulnerabilities. A digital database presents a substantial value in information space. In essence, a database is created, stored, managed, and transmitted through various

computer networks, leading to an overload in the contents of the security database. Thus, securing the database is considered the last line of defense for protecting data from disclosure. Recent security breaches that can disclose confidential information from databases include user privacy threats, sensitive data threats, malicious attacks, and insider attacks [1,2]. These security breaches can result in data loss and information theft. Therefore, recent security research proposed digital watermarking technology as a promising strategy for maintaining the secrecy and integrity of databases [3]. Digital watermarking is considered a method for information hiding; that is, it hides data from attackers. Watermarking technology can protect data by updating the database and inserting watermarks in the shape of texts, binary bits, or images known only to the data owner. Protecting data through watermarking can protect intellectual property rights, promote ownership proofing, and ensure content authentication and security [4].

The watermarking process is conducted by embedding a watermark into the database content, known as the secure part. Watermarked data are transmitted across computer networks, which can be vulnerable to various security attacks. Once the destination host receives the watermarked data, the detection process is executed to detect the watermarked data without distortion. As a result, digital watermarking can be used to authenticate the owner who bears the sole right to modify the database content. However, the majority of research on watermarking related to non-numeric data exhibited high distortion rates. Thus, this study proposes a novel method for applying digital watermarking to non-numeric databases with low rates of data distortion to improve the security, robustness, and effectiveness of data. The paper contributes to the literature as follows:

- For the first time, a watermark is embedded into non-numeric relational databases using the histogram shifting (HS) of the prediction error expansion technique with a low distortion rate and high watermarking capacity.
- The proposed method is tested to verify the robustness and security of watermarked data against popular attacks, such as insertion, deletion, and alteration attacks.
- The watermark embedding process resulted in replacing the entire word in the candidate (multi-word) attribute. In this manner, the attackers cannot anticipate the watermark without using a secret key owned by the database owner. After applying the detection process, data can be recovered by 100%.
- The proposed method does not affect the usefulness of data. Moreover, it can hide watermark bits in a large space of the database. This method is more secure due to the use of a secret key owned by the database owner.

The remainder of the paper is structured as follows. Section 2 provides a review of the related literature. Section 3 discusses the structure of the algorithm using the HS model. Section 4 describes the design of the proposed algorithm. Section 5 presents the experimental results and data analysis of the proposed algorithm. Lastly, Section 6 presents the conclusion and recommends directions for future work.

2 Literature Review

An extensive literature review is conducted to explain the concept of watermarking by highlighting the major researches and techniques on watermarking database on numeric data as presented in [5–13] and non-numeric data researches as explained in [14–22]. The literature is filled with substantial studies in the watermarking field for numeric data, images, audio, and videos. Agrawal and Kiernan [5] were the first to develop the database watermarking method. The technology of database watermarking

has gained widespread interest. A database contains different types of data, mainly grouped under two categories, namely, numeric and non-numeric.

2.1 Numeric Data

Numeric data display little change with a suitable range on the least significant bit of its binary bits. Bhattacharya and Cortesi [6] proposed inserting the watermark after splitting tuples as a replacement. However, the embedding rate of difference expansion watermarking (DEW) will be reduced according to the distortion constraint. In terms of Robust Reversible Watermark (RRW), Iftikhar et al. [7] introduced a reversible and strong watermarking technique for relational data that works on numeric attributes. In this technique, features are selected using mutual information. Moreover, Wu and Shih [8] introduced a robust method that uses a genetic algorithm (GA) merged with difference expansion watermarking (GADEW) to develop a strong database watermarking solution. Alternatively, Imamoglu et al. [9] proposed a new bio-inspired optimization algorithm and developed a method for inserting watermarks into a relational database. The current study recognizes and applies DEW to achieve two objectives: reducing distortion rate and decreasing elaboration within database watermarking. As presented by Hu et al. [10], GA combined with HS watermarking produced a new robust and reversible watermarking technique called Genetic algorithm histogram shifting watermarking (GAHSW) that are applied to numeric data. Franco-Contreras and Coatrieux [11] presented a robust and lossless watermarking scheme that utilizes circular histogram modulation for relational databases. As presented by Tufail et al. [12], evolutionary techniques were applied to the proposed scheme. The MRMR-based FSS technique promotes the robustness of watermarks because it presents attackers with difficulty in anticipating altered columns. The results demonstrated that the developed WET technique outperforms recent techniques, such as DEW, GADEW, RRW (Robust Reversible Watermarking), and PEEW (Prediction-error Expansion Watermarking). Lian [13] illustrates that a reversible watermarking method based on the ant colony algorithm is combined with DEW. The results revealed its ability to maximize watermark space and promoting its robustness against several types of attack.

2.2 Non-numeric Data

Currently, watermarking textual data has become popular according to the large number of documents shared and produced daily. With the progress of the internet and communication world-wide, several text watermarking techniques have been developed. For example, Al-Haj and Odeh [14] presented a robust and blind database watermarking algorithm for entering watermark in the shape of binary images in non-numeric multi-word columns along specific database rows. Moreover, Sion et al. [15] and Sion [16] applied the watermarking process to categorical data by switching columns from text to numeric using particular rules. The least significant bit was updated to detect changes in attributes. Hanyurwimfura et al. [17] and Melkundi and Chandankhede [18] proposed computing the editing distance between two neighboring words and selecting two words with the lowest distance for the embedding process. Khanduja et al. [19] computed the American Standard Code for Information Interchange (ASCII) sum of primary keys to identify symbols added to columns, whose values can then be changed by adding corresponding characters at the end of it in the determined row to insert the watermark. Bedi et al. [20] demonstrated that the watermark is produced using eigenvalues based on the measured ASCII of non-numeric attributes. For each row, a relational matrix is created. Conversely, Khadam et al. [21] proposed a secure and robust digital watermarking method that provides copyright protection to a text document using data mining. Based on the experimental results, the method is robust against formatting attacks. Moreover, Li et al. [22] introduced a relational

database watermarking algorithm for non-numeric attributes. This method is based on the insertion of words on an attribute to replace its value using Chinese word segmentation to detect removable words. Furthermore, the method explains the rate of similarity distortion produced by entering words. Tab. 1 provides a brief overview of watermarking techniques for numeric and non-numeric relational databases. Recent research papers are compared based on their data format, watermarking information, and the percentage of recovery after insertion, alteration, and deletion attacks.

Table 1: Classification of numeric and non-numeric watermarking techniques

Technique	Data format	Watermark information	Data recovery after attacks %		
			Insertion 50%	Alteration 50%	Deletion 50%
DEW [2]	Numeric	Binary Bit	88%	90.5%	90%
RRW [3]	Numeric	Binary Image	100%	65%	50%
GADEW [4]	Numeric	Binary Bit	100%	75%	50%
FFADEW [5]	Numeric	Binary Bit	63.24%	63%	25%
GAHSW [6]	Numeric	Binary Bit	100%	50%	50%
WET [8]	Numeric	Binary Bit	100%	70%	50.50%
Imamoglu et al. [9]	Numeric	Binary Bit	–	75%	65%
Al-Haj et al. [14]	Non-numeric	Binary Bit	65%	65%	NA
Hanyurwimfura et al. [13]	Non-numeric	Binary Bit	65%	45%	50%
Khanduja et al. [15]	Non-numeric	Vowel Character	60%	60%	53.33%
Li et al. [18]	Non-numeric	Binary String	NA	80%	NA

The watermarking process on non-numeric data leads to high rates of distortion of detected data after applying insertion, alteration, and deletion attacks. Thus, we proposed an effective and robust technique to minimize the distortion rate of detected data.

3 Histogram Shifting Model

Ni et al. [23] applied the HS technique to digital image watermarking for the first time, which became the focus of scholarly attention in the recent period [24–34]. In an interval from 2006 to 2017, HS was applied to multimedia data, such as images for inserting watermarks [24]. HS is superior to the existing methods in the embedding payload. Conversely, Lin et al. [25] used HS as a multilevel technique and displayed its ability to achieve high rates in reaching the acceptable range of data distortion.

The concept of HS is based on hiding watermarks using a histogram peak value. Alghamdi et al. [35] proposed a database watermarking scheme to obtain non-zero prediction errors from two neighboring raw values in the database. Constructing the histogram involves a horizontal axis is represented by error values, whereas the frequency of each value of prediction error P_e is the vertical axis. Using Eq. (1), we compute for P_e as follows:

$$P_e = x_{i+1} - x_i, \tag{1}$$

where: $x_{i+1} - x_i$ are two neighboring values. A peak bin with non-zero frequency is found in the histogram, which represents prediction error as P. To create a vacancy near P, all bits are shifted by one cell, except for P. After that, each prediction error is considered until P is achieved. The watermark bit ($w = 0$ *or* $w = 1$) is inserted, and the new prediction error value P'_e is computed as demonstrated by Eq. (2):

$$P'_e = \begin{cases} P_e + 1 \; such \; that \; P_e > P \\ P_e + w \; such \; that \; P_e > P \\ P_e \; otherwise \end{cases} \tag{2}$$

The method used is the inverse integer Harr wavelet transform [35] to generate two new attribute values, namely, X'_i and X'_{i+1}, from the calculated P'_e. The median value of the two values is calculated, as shown in Eq. (3):

$$X_m = \left[\frac{(X'_i + X'_{i+1})}{2} \right] \tag{3}$$

The watermarked attribute values can be calculated as follows:

$$X'_i = X_m - \left\lfloor \frac{P'_e}{2} \right\rfloor \tag{4}$$

$$X'_{i+1} = X_m + \left\lfloor \frac{P'_{e+1}}{2} \right\rfloor \tag{5}$$

For example, $X_i = 102$ and $X_{i+1} = 108$ are two neighboring values. Thus, $P_e = 108 - 102 = 6$. Then, X_m will be equal to $(102 + 108)/2 = 105$. If peak bit $P = 4$, whereas the embedded watermark bit $w = 1$, then the new value of $P'_e = 4 + 1 = 5$. The new attribute values are computed as follows:

$$X'_i = 105 - \left\lfloor \frac{4}{2} \right\rfloor = 103 \tag{6}$$

$$X'_{i+1} = 105 + \left\lfloor \frac{4+1}{2} \right\rfloor = 107 \tag{7}$$

4 Proposed Watermarking Algorithm

This section presents the proposed algorithm for watermarking non-numeric and multi-word databases based on the HSW of prediction-error expansion. The advantage of using Histogram Shifting Model is that it provides a robust watermarking method and improves data quality for databases. In this scheme, all tuples are securely divided into non-overlapping subsets. A single watermark bit is embedded into some tuples of a subset by updating attribute values in the group. A watermark bit is embedded repeatedly into one group. Fig. 1 displays the overall framework for preprocessing, embedding the watermark, and detecting the watermark.

Based on Fig. 1, the following subsections present the three main phases of watermarking non-numeric databases.

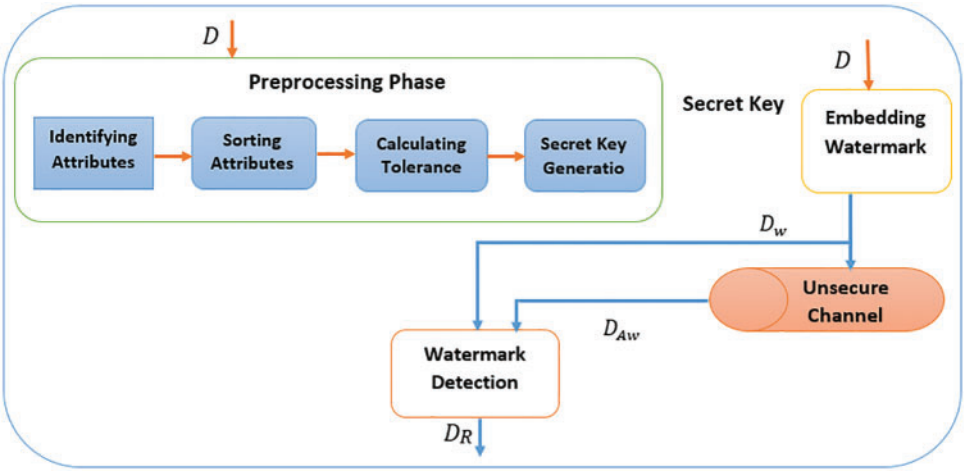


Figure 1: Schematics of the watermarking process

4.1 Preprocessing Phase

The preprocessing phase is composed of four steps:

- Identifying candidate attributes: non-numeric candidate attributes are selected from the database for inserting or embedding watermarks.
- Sorting non-numeric attributes: the selected non-numeric attributes are sorted alphabetically from A to Z based on headers to enhance the robustness of the proposed algorithms.
- Computing the semantic distortion range: the distortion range of the attributes is computed using the maximum and minimum values of each attribute based on the sum of the word ASCII. For example, the ASCII of the word {'Brand'} = 66{'B'} + 114{'r'} + 97{'a'} + 110{'n'} + 100{'d'} = 487. To compute the tolerance of attributes according to ASCII values, the tolerance of the *j*-th attribute can be computed using Eq. (8):

$$\hat{y} = \left\lfloor \frac{max[j] + min[j]}{2} \right\rfloor \tag{8}$$

- Grouping tuples: a random secret key is generated, and the group number of each tuple is determined using Eq. (9):

$$N_u = H(Ks|H(Ks|tu.Pk))mod\ N_g \tag{9}$$

where:

- N_u = a group serial number;
- "|" = the concatenation operation;
- H () = a hash function;
- K_S = the grouping secret key;
- $tu.\ PK$ = the tuple's primary key;
- N_g = the number of groups in the database, which is equal to the number of watermark bits. For example, if the watermark bits are equal to 48, then $N_g = 48$.

For a quick reference, Tab. 2 contains lists of the notations used in this paper.

Table 2: Notations used in the paper

Symbol	Description	Symbol	Description
D	Original database	D_W	Watermarked database
N	Total number of tuples in a database	D_{AW}	Watermarked database after attacks
P	Peak value	P_e	Prediction error of original database
L	Length of the watermark	P'_e	New prediction error
A_{ij}	Candidate attribute to be watermarked	Z	Total number of columns in a database
P_a	Array for storing peak points	Y'	Watermarked attribute value
Mp	Array for storing primary key information	y_r	Restored attribute value
Y	Value of attribute	A	Feature/column/attribute of original database
ŷ	Value determined by the maximum and minimum of the column	$min[j]$	Minimum of j-th column
K_S	Secret key	$max[j]$	Maximum of j-th column
$tu.PK$	Primary key of the tuple	nu	Serial number of certain groups
W^{det}	Detected watermark bits	N_g	Total number of groups
D_R	Restored database	W	Watermark bit

4.2 Embedding Phase

The HS method is mainly dependent on embedding one watermark bit in each group. Therefore, the embedding process of the watermark is considered a repetitive process of performing consecutive steps for each group. Fig. 2 depicts the process of embedding watermarks.

The overall processes for embedding the watermark are explained as follows:

- Determining the j-th column, whose value is altered using the following equation:

$$j = H(K_S|tu.PK)\%z \tag{10}$$

where j is the value of the candidate column to be watermarked, whereas z denotes the number of non-numeric attributes in the database.

- Constructing a histogram of each group based on the following steps:
  - First, compute the P_e value of each group using Eq. (11), then compute P'_e using Eq. (12). The peak bin with a non-zero frequency can be calculated using the absolute value of P_e before inserting watermarks.
  - Second, store the peak value as P in an array called P_a, which can be used as an input to the detection process. P_e can be shifted to both sides of P. In the last step, we scan each P_e to insert a 1-bit watermark W. P'_e is represented as P_e's corresponding new prediction error, which is computed using Eq. (12).

$$P_e = y - \hat{y} \tag{11}$$

$$P'e = \begin{cases} P_e + 1 & Pe \geq P + 1; \\ P_e + 1 & pe \leq -(p+1); \\ P_e & pe = p, w = 0; \\ P_e + 1 & pe = p, w = 1; \\ P_e & pe = -p, w = 0; \\ P_e - 1 & pe = -p, w = 1 \end{cases} \tag{12}$$

Then, the new attribute value y' is derived using Eq. (13):

$$y\prime = P'_e + \hat{y}. \tag{13}$$

– Lastly, embed the watermark bit in the selected j-th attribute.

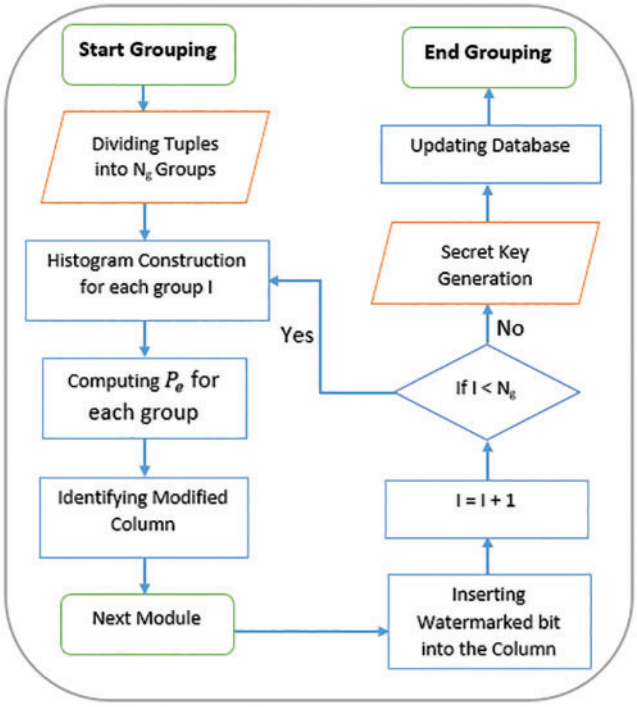


Figure 2: Process of embedding the watermark

The abovementioned steps are repeated for all rows in the database to be altered. Algorithm 1 explains the overall embedding process.

Algorithm 1: Watermarking Embedding Process

Input: Variables: D, W, z. // read non-numeric data from excel file.
Output: $D_{W,}$ $P_{a,}$ Mp.
1: Compute tolerance of the columns based on sum of word's ASCII numbers by Eq. (8).
2: Divide tuples in database in groups by using Eq. (9).
3: **for** w = 1 to l **do**
4: //loop will iterate for all watermark bits' w from 1 to length l of the watermark.
5: **for** i = 1 to N/Ng **do**
6: //loop will iterate for all tuples of each group.
7: j = H (KS | tu.PK) %z;
8: //identify marked attribute column.
9: **if** Aij = max[j] or Aij = min[j] **then**
10: j = (j + 1) %z;
11: insert tu.PK of Aij into *Mp*;
12: **End if**
13: P_e is calculated by using Eq. (11);
14: **End for**
15: p is determined by the peak point of the histogram of the absolute value of P_e;
16: insert p into pa;
17: **for** i = 1 to N/Ng **do**
18: P′e is calculated by using Eq. (12);
19: the corresponding attribute value is watermarked by using Eq. (13);
20: **End for**
21: **End for**
22: return DW, pa, *Mp*.

4.3 Watermark Detection Phase

The watermark detection and data recovery procedure are reported in detail using the following steps.

- To detect the watermark and recover the original data, sorting non-numeric attributes alphabetically from A to Z based on headers is necessary.
- Computing the minimum and maximum of columns based on the summation of word's ASCII numbers (ex: 'Brand' ASCII ≥ 66{'B'} + 114{'r'} + 97{'a'} + 110{'n'} + 100{'d'} = 487)
- Applying Eq. (9) to determine the group number for each tuple based on the secret key, which was used in Algorithm 1.
- Detecting the watermark using the HS detection technique as follows:
  – Consider y' is the value of the attribute in the watermarked database, where as $y^{\wedge}$ is the predictor of y'. Thus, the new prediction error P'_e can be calculated using Eq. (14).

$$P'_e = y' - \hat{y} \tag{14}$$

  – Consider the P'_e value that determines whether the inserted bit (w) is 0 or 1 and count the number of zeroes ≥ (a) and ones in that group ≥ (b).

- Recovering data, the value of the attribute is restored as follows:

$$y_r = \begin{cases} y' - 1, & pe' \geq p + 1; \\ y', & pe' = p; \\ y' + 1, & pe' \leq -(p + 1). \end{cases} \tag{15}$$

After computing the original value of the entire group, determine the watermark bit based on whether (a > b) watermark bit is 0; otherwise, the watermark bit is 1. Algorithm 2 describes the detection process.

Input: DW, z, pa, mp.//read watermarked data from excel file
Output: DR, W^{det}
1: Compute tolerance of columns based on sum of word's ASCII numbers by Eq. (8)
2: Divide the tuples in database in groups by using Eq. (9);
3: a = 0; b = 0;
4: **For** s = 1 to 1 **Do**
5: **For** i = 1 to N/Ng **Do**
6: //loop will iterate for each group tuples.
7: j = H (KS|tu.PK) %z;
8: // determine the column that it's value is altered
9: **If** (A^wij = max [j] or Aw ij = min [j]) and find the tu.pk of Aij in mp **Then**
10: **j** = (j+1) %Z;
11: **End If**
12: Pe' is calculated by using Eq. (13);
13: **If** Pe' = Pa[s] **Then**
14: The extracted watermark bit (W^{det}) is o;
15: **else if** Pe' = Pa [s] + 1 or Pe' = — (pa[s] + 1) **Then**
16: The Extracted Watermark bit (W^{det}) is 1;
17: **End if**
18: The original attribute value is restored by using Eq. (14);
19: **If** w^{det} = 0 **Then**
20: a = a + 1; //count the number of the extracted watermark bit (w^{det}) is 0
21: **Else**
22: b = b+1; //count the number of the extracted watermark bit (w^{det}) is 1
23: **End If**
24: **End For**
25: **If** a > b **Then**
26: w^{det} = 0
27: **Else**
28: w^{det} = 1
29: **End if** //the majority voting mechanism is used to identify the last watermark bit.
30: w^{det} = W^{det} + w^{det}
31: **End for**
32: Return D_R W^{det}

---

5 Experimental Results and Data Analysis

The performance and robustness of the technique can be evaluated by conducting experiments within the Intel Core i5 with a 2.40 GHz CPU and 8 GB RAM and with a Windows 10 operating system. The selected database was the two datasets. The first dataset [36] contains order-logs obtained from a highly reputable online shopping agency with 344 records and 27 fields. We selected eight non-numeric attributes for the first trial as part of the experimental requirements. The second dataset was that of bank churners, which contains 1,000 records and 36 attributes. Similarly, we selected seven non-numeric attributes for the second trial. The experiment was conducted in two parts. The first aimed to analyze the proposed method's performance, such as computational storage cost, computation duration, and watermarking capacity. The second compared the robustness of the proposed method with other state-of-the-art approaches by subjecting them to several well-known attacks.

5.1 Watermark Capacity, Cost, and Time Complexity Analysis

The three criteria, namely, watermark capacity, storage cost, and time complexity, are used to analyze the performance of the proposed scheme, which can embed the watermark in each tuple for each group in the database. Thus, the watermark embedding rate was increased. High levels of watermarking capacity indicate that additional watermark information can be embedded. Thus, the watermarking capacity of the proposed method can reach 100%. In other words, the embedding rate is maximized. Meanwhile, Khanduja et al. [19] cited that the column value can be changed by adding characters at the end of the column in the specified row to insert the watermark. As such, this option consumes more storage. Second, when the value of the peak point is equal to 2, the column value can be changed by replacing each word in the specified row with another word; thus, the storage space of the memory remains the same. Thus, the original and watermarked data and the recovered data are of the same size (40 KB). The computation duration of the proposed algorithm is reported in three phases, namely, preprocessing, embedding, and detection. The time complexity of the data preprocessing phase reaches 247 ms. In contrast to other reversible watermarking methods that use the GA algorithm to select the column in which to embed the watermark, such as GAHSW and RRW, the time consumption in this phase using these techniques exceeds that of the proposed method. Meanwhile, the time complexity of embedding the watermark in the proposed method is 1,813 ms whereas the time complexity for detecting the watermark was 2,183 ms.

5.2 Robustness Analysis

In this section, the robustness of the proposed scheme is evaluated under three popular attacks, namely, insertion, deletion, and alteration. The evaluation consists of two parts. The first aims to measure the extent of the effect of the HSW algorithm on the first dataset (order log) after embedding the watermark at two time points. The first pertains to the time when the value of peak point P equals 1; that is, the change was made to the data by replacing the word's first letter in each modified column. This result may influence the data negatively because it can enable an attacker to predict the original value of the attribute. The second time point is when the value of the peak point equals 2. The result denotes a change in the entire word in the altered column through the transposition and substitution of the letters of the word as a whole. In this manner, predicting the original value of the modified attribute is difficult for the attacker unless the attacker knows the secret key owned by the database owner. The study considered popular attacks, such as addition, deletion, and alteration, and conducted them on the first dataset to measure the impact of the attacks for both cases and compare the results in terms of data recovery and detection of watermark information.

A comparison of the watermark detection of HSW on the order log dataset is performed in two cases. The first case is based on a change in the entire word and while the second case is based on a change in one letter. Figs. 3–5 are presented to explain the comparison between the two cases after insertion, alteration and deletion attacks. This comparison is explained in the following subsections.

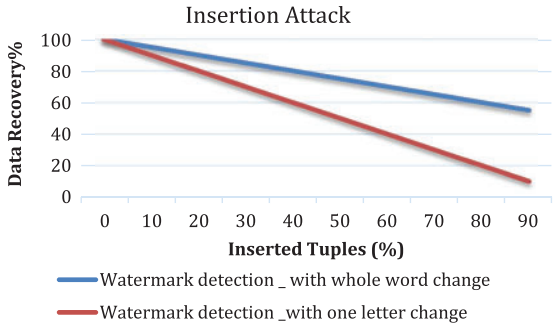


Figure 3: Comparison of watermark detection of HSW on the order log dataset after insertion attacks

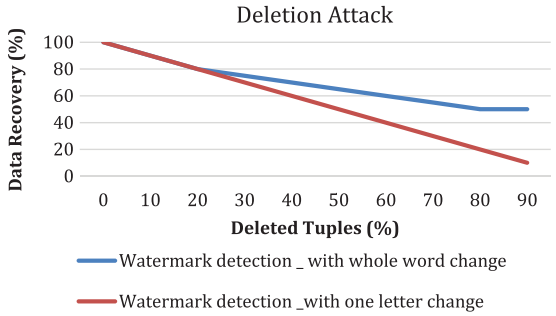


Figure 4: Comparison of watermark detection of HSW on the order log dataset after deletion attacks

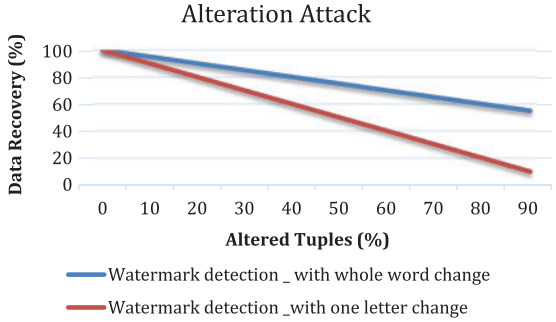


Figure 5: Comparison of watermark detection of HSW on order log dataset after alteration attacks

5.2.1 Insertion Attack

During an insertion attack, attackers are dependent on adding the subset of rows to the watermarked database to facilitate watermark detection. The tuples are added to the database at ratios ranging from 10% to 90%. Fig. 3 presents a comparison of the watermark detection of HSW on the order log dataset in two cases, namely, a change in the entire word and a change in one letter after

insertion attacks. The watermark will be detected, and data will be recovered by different values under insertion attack that; the watermark will be detected in all cases of tuples insertion in both cases of data change. This type of attack exerts a low impact on the proposed algorithm.

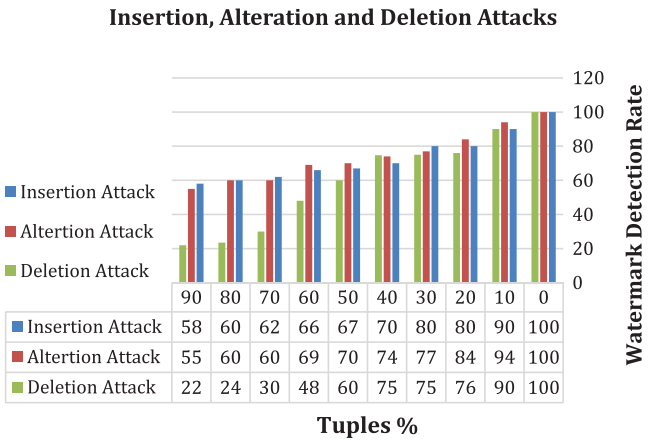
5.2.2 Deletion Attack

Under deletion attacks, attackers randomly delete rows of the watermarked database to facilitate watermark removal. The tuples are selected from the database at ratios ranging from 10% to 90% and deleted. The watermark was detected, whereas data were recovered using different values. Fig. 4 depicts a comparison of the watermark detection of HSW on the order log dataset after deletion attacks. The watermark was detected, whereas data were recovered using different values under deletion attack. However, the watermark cannot be deleted even after 90% of the database was deleted in both cases of data change.

5.2.3 Alteration Attack

During alteration attacks, attackers alter random rows or columns of the watermarked database to facilitate watermark detection. The tuples are selected from the database at ratios ranging from 10% to 90% and updated. The watermark is detected, whereas data are recovered using different values. Fig. 5 indicates that the watermark will be detected and that data will be recovered even after 90% of the database is altered in both data change cases. This type of attack was ineffective against the proposed algorithm because the watermark is embedded in more than 50% of the database.

The second part of the evaluation aims to measure the robustness of the proposed method on a different dataset, which is larger than the first dataset in terms of the number of records, to measure the data distortion rate after embedding the watermark. The experimental results demonstrate that the watermark information can be embedded in a large space of the database and that the entire word in the candidate (multi-word) attribute can be replaced. Common attacks are considered and conducted on this dataset, such as addition, deletion, and alteration. Fig. 6 presents a comparison of watermark detection rate after three different attacks namely, insertion, alteration and deletion attacks on the second dataset (bank churners).



	90	80	70	60	50	40	30	20	10	0
Insertion Attack	58	60	62	66	67	70	80	80	90	100
Altertion Attack	55	60	60	69	70	74	77	84	94	100
Deletion Attack	22	24	30	48	60	75	75	76	90	100

Figure 6: Comparison of watermark detection rate after insertion, alteration and deletion attacks on bank churners' dataset

The results indicate that the higher the insertion attack rate, the lower the watermark detection rate. The percentage of detection rate was 90% since the percentage of the insertion attack is 10%. When the rate of the insertion attack reached 90%, the watermark detection rate was 58%. This finding implies that the data did not lose its usability. Also illustrates that during an alteration attack, the watermark detection rate increased with the decrease in the rate of attack.

The percentage of watermark detection reached 94% at attack rates of 10% and 90%. Thus, the watermark can be detected at more than 50%, with a percentage reach to 55%. Therefore, this type of attack was ineffective against the proposed algorithm. In the case of deletion attacks on the same dataset. The watermark recovery rates ranged from 90% and 75% to 30% and 22.5%.

Even when 90% of the dataset is deleted, the watermark is detected at 22.5% with the increase in the attack rate. Figs. 7–9 compare the two datasets in terms of data recovery and watermark match. However, the watermark will be detected, whereas data will be recovered regardless of the number of tuples inserted, deleted, or updated on the watermark database.

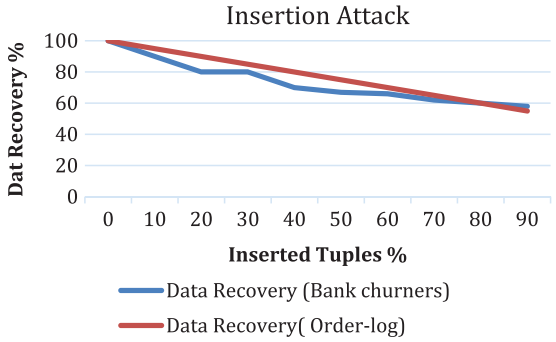


Figure 7: Insertion attack

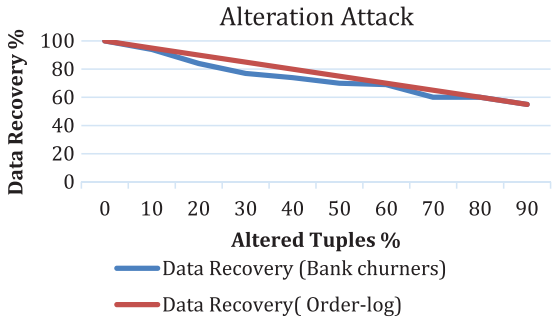


Figure 8: Alteration attack

Tab. 3 presents the overall result of watermark detection after various attacks on the watermark detection from the two databases at ratios ranging from 10% to 90%. The table demonstrates that the watermark is sufficiently detected, indicating that the proposed algorithm can recover more than half of the embedded watermark in all addition and alteration attacks cases. Although 90% of the data were deleted, the watermark detection in one of the datasets remains unclear. Nevertheless, data can be detected and restored.

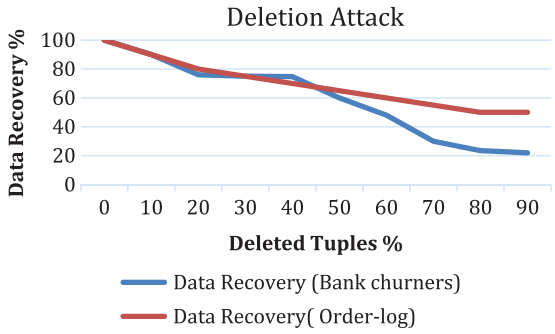


Figure 9: Deletion attack

Table 3: Watermark detection rates of data after insertion, alteration, and deletion attacks

Range	Data insertion attack		Data alteration attack		Data deletion attack	
	Order log	Bank churners	Order log	Bank churners	Order log	Bank churners
10%	95%	90%	95%	90%	90%	90%
20%	90%	80%	90%	80%	80%	80%
30%	85%	80%	85%	80%	75%	75%
40%	80%	70%	80%	70%	70%	74.7%
50%	75%	67%	75%	67%	65%	60%
60%	70%	66%	70%	66%	60%	48%
70%	65%	62%	65%	62%	55%	30%
80%	60%	60%	60%	60%	50%	23.5%
90%	55%	58%	55%	58%	50%	22%

After comparing the two datasets, we find the data detection rate decreases with the increase in the percentage of attacks in all cases of attacks at ratios that range from 10% to 90%. The first data set (order log) indicates that the detection rate reached 55%. However, in the second dataset (bank churners), the detection rate can reach 58%, even at an attack rate of 90%. Therefore, the proposed algorithm can recover more than half of the watermark in all addition and alteration attacks cases. In the case of deletion attacks, however, the detection rates achieved for the first and second datasets are 50% and 22%, respectively, at a deletion rate of 90%.

6 Conclusion and Future Works

Currently, scholarly attention is mainly focused on database watermarking because it can solve the problems of ownership proofing and copyright protection in the process of sharing databases. The reversible watermarking technique can retrieve the original data and fully preserve data quality. This study presented a novel robust and reversible watermarking method for non-numeric relational databases. The experimental results demonstrate that the proposed method minimizes distortion and promotes watermarking robustness. The method used the grouping and the majority voting

mechanism; thus, it can outperform other methods in detecting the majority of watermark information and recovering a large portion of data in all types of attacks regardless of the size of the tested data. Conversely, the results suggest that despite exposing the database to an insertion attack at 50%, 75% of the watermark is retained. Thus, the proposed algorithm can recover more than half of the embedded watermark in all addition and alteration attacks cases, retain watermark information, and restore data completely. Future directions of this research can be conducted on applying watermarking techniques for securing confidential data on cloud computing. The main process is to perform digital watermarking using encryption techniques to maintain the privacy of cloud services and authentication of cloud users.

Funding Statement: The authors received no specific funding for this study.

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

References

[1] H. Deng, Z. Qin, Q. Wu, Z. Guan, R. Deng *et al.,* "Identity-based encryption transformation for flexible sharing of encrypted data in public cloud," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3168–3180, 2020.

[2] Y. Zhang, J. Yu, R. Hao, C. Wang and K. Ren, "Enabling efficient user revocation in identity-based cloud storage auditing for shared Big data," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 3, pp. 608–619, 2020.

[3] M. Lázaro, M. Olliaro, A. Cortesi and C. Feregrino, "Semantic-driven watermarking of relational textual databases," *Expert Systems with Applications,* vol. 167. Elsevier, pp. 1–23, 2020.

[4] K. H. El Drandaly, W. Khedr, A. Mostafa and I. Mohamed, "A digital watermarking for relational database: state of Art techniques," *International Journal of Advanced Science and Technology,* Elsevier, vol. 29, pp. 870–883, 2020.

[5] R. Agrawal and J. Kiernan, "Watermarking relational databases," in *Proc. of the 28th international conference on Very Large Data Bases*, Hong Kong, China, VLDB Endowment, pp. 155–166, 2002.

[6] S. Bhattacharya and A. Cortesi, "A distortion free watermark framework for relational databases," in *ICSOFT*, Sofia, Bulgaria, (2), pp. 229–234, 2009.

[7] S. Iftikhar, M. Kamran and Z. Anwar, "RRW- robust and reversible watermarking technique for relational data," *IEEE Transactions on Knowledge & Data Engineering*, vol. 27, no. 4, pp. 1132–1145, 2015.

[8] Y. Wu and F. Y. Shih, "Genetic algorithm based methodology for breaking the steganalytic systems," *IEEE Transactions on Systems Man & Cybernetics Part B Cybernetics, a Publication of the IEEE Systems Man & Cybernetics Society*, vol. 36, no. 1, pp. 24–31, 2006.

[9] M. B. Imamoglu, M. Ulutas and G. Ulutas, "A new reversible database watermarking approach with firefly optimization algorithm," *Mathematical Problems in Engineering*, vol. 2017, no. 2, pp. 1–14, 2017.

[10] D. Hu, D. Zhao and S. Zheng, "A new robust approach for reversible database watermarking with distortion control," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 6, pp. 1024–1037, 2019.

[11] J. Franco-Contreras and G. Coatrieux, "Databases traceability by means of watermarking with optimized detection," *Digital Forensics and Watermarking*, vol. 10082, pp. 343–357, 2016.

[12] H. Tufail, K. Zafar and A. R. Baig, "Relational database security using digital watermarking and evolutionary techniques," *Computational Intelligence, Wiley*, vol. 34, no. 4, pp. 693–716, 2019.

[13] J. Lian, "A new reversible database watermarking approach with ant colony optimization algorithm," in *3rd Int. Symposium on Big Data and Applied Statistics*, Vancouver, Canada, vol. 1616, pp. 10–12, 2020.

[14] A. Al-Haj and A. Odeh, "Robust and blind watermarking of relational database systems," *Journal of Computer Science*, vol. 4, no. 12, pp. 1024–1029, 2008.

[15] R. Sion, M. Atallah and S. Prabhakar, "Rights protection for categorical data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 7, pp. 912–926, 2015.
[16] R. Sion, "Proving ownership over categorical data," in *IEEE Int. Conf. on Data Engineering*, Boston, MA, USA, pp. 1–12, 2004.
[17] D. Hanyurwimfura, Y. Liu and Z. Liu, "Text format based relational database watermarking for non-numeric data," in *IEEE Int. Conf. on Computer Design & Applications*, Qinhuangdao, China, pp. 312–316, 2010.
[18] S. Melkundi and C. Chandankhede, "A robust technique for relational database watermarking and verification," in *IEEE Int. Conf. on Communication, Information & Computing Technology*, Mumbai, India, pp. 1–7, 2015.
[19] V. Khanduja, A. Khandelwal and A. Madharaia, "A robust watermarking approach for non-numeric relational database," in *IEEE Int. Conf. on Communication*, Mumbai, India, pp. 1–5, 2012.
[20] R. Bedi, A. Thengade and V. Wadhai, "A new watermarking approach for non-numeric relational database," *International Journal of Computer Applications*, vol. 13, pp. 37–40, 2011.
[21] U. Khadam, M. Iqbal, M. Azam, S. Khalid, S. Rho *et al.,* "Digital watermarking technique for text document protection using data mining analysis," *IEEE Access, Special Section of Data Mining for Internet of Things*, vol. 7, pp. 64955–64965, 2019.
[22] W. Li, J. Yan and Z. Zhang, "Relational database watermarking based on Chinese word segmentation and word embedding," in *IEEE Int. Conf. on Computer Communications and Networks (ICCCN)*, Honolulu, HI, USA, pp. 1–6, 2020.
[23] Z. Ni, Y. Shi, N. Ansari and W. Su, "Reversible data hiding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 354–362, 2006.
[24] G. Xuan, Q. Yao, C. Yang, J. Gao, P. Chai *et al.,* "Lossless data hiding using histogram shifting method based on integer wavelets," in *Int. Workshop on Digital Watermarking*, Jeju Islan, Korea, Springer, pp. 323–332, 2006.
[25] C. C. Lin, W. L. Tai and C. C. Chang, "Multilevel reversible data hiding based on histogram modification of difference images," *Pattern Recognition,* Elsevier, vol. 41, no. 12, pp. 3582–3591, 2008.
[26] W. L. Tai, C. M. Yeh and C. C. Chang, "Reversible data hiding based on histogram modification of pixel differences," *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 19, no. 6, pp. 906–910, 2009.
[27] D. G. Yeo, H. Y. Lee and B. M. Kim, "High capacity reversible watermarking using differential histogram shifting and predicted error compensation," *Journal of Electronic Imaging*, vol. 20, no. 1, pp. 40–43, 2011.
[28] H. T. Wu and J. Huang, "Reversible image watermarking on prediction errors by efficient histogram modification," *Signal Processing*, vol. 92, no. 12, pp. 3000–3009, 2012.
[29] L. Luo, Z. Chen, M. Chen, X. Zeng and Z. Xiong, "Reversible image watermarking using interpolation technique," *IEEE Transactions on Information Forensics & Security*, vol. 5, no. 1, pp. 187–193, 2010.
[30] X. Gao, L. An, Y. Yuan, D. Tao and X. Li, "Lossless data embedding using generalized statistical quantity histogram," *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 21, no. 8, pp. 1061–1070, 2011.
[31] X. Li, W. Zhang, X. Gui and B. Yang, "A novel reversible data hiding scheme based on Two-dimensional difference-histogram modification," *IEEE Transactions on Information Forensics & Security*, vol. 8, no. 7, pp. 1091–1100, 2013.
[32] B. Yang, M. Schmucker, C. Busch, X. Niu and S. Sun, "Approaching optimal value expansion for reversible watermarking," *ACM Workshop on Multimedia and Security*, pp. 95–102, 2005.
[33] G. Xuan, Y. Q. Shi, P. Chai, X. Cui, Z. Ni *et al.,* "Optimum histogram pair based image lossless data embedding," in *Int. Workshop on Digital Watermarking*, Springer, pp. 264–278, 2008.
[34] J. Wang, J. Ni, X. Zhang and Y. Q. Shi, "Rate and distortion optimization for reversible data hiding using multiple histogram shifting," *IEEE Transactions on Cybernetics*, vol. 47, no. 2, pp. 315, 2017.

[35] A. S. Alghamdi, S. Naz, A. Saeed, E. Al Solami, M. Kamran *et al.,* "A novel database watermarking technique using blockchain as trusted third party," *Computers, Material and Continua (CMC)*, vol. 70, no. 1, pp. 1585–1601, 2022.

[36] R. A. Ahmed, M. E. Shehaba, S. Morsy and N. Mekawie, "Performance study of classification algorithms for consumer online shopping attitudes and behavior using data mining," in *Fifth Int. Conf. on Communication Systems and Network Technologies*, (IEEE), 2015.