Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

On the role of optimization algorithms in ownership-preserving data mining

Muhammad KAMRAN‡, Ehsan Ullah MUNIR

Department of Computer Science, COMSATS Institute of Information Technology, Wah Cantt 47010, Pakistan

E-mail: muhammad.kamran@ciitwah.edu.pk; ehsanmunir@comsats.edu.pk

Received Aug. 17, 2016; Revision accepted Feb. 14, 2017; Crosschecked Feb. 15, 2018

Abstract: Knowledge extraction from sensitive data often needs collaborative work. Statistical databases are generated from such data and shared among various stakeholders. In this context, the ownership protection of shared data becomes important. Watermarking is emerging to be a very effective tool for imposing ownership rights on various digital data formats. Watermarking of such datasets may bring distortions in the data. Consequently, the extracted knowledge may be inaccurate. These distortions are controlled by the usability constraints, which in turn limit the available bandwidth for watermarking. Large bandwidth ensures robustness; however, it may degrade the quality of the data. Such a situation can be resolved by optimizing the available bandwidth subject to the usability constraints. Optimization techniques, particularly bioinspired techniques, have become a preferred choice for solving such issues during the past few years. In this paper, we investigate the usability of various optimization schemes for identifying the maximum available bandwidth to achieve two objectives: (1) preserving the knowledge stored in the data; (2) maximizing the available bandwidth subject to the usability constraints to achieve maximum robustness. The first objective is achieved with a usability constraint model, which ensures that the knowledge is not compromised as a result of watermark embedding. The second objective is achieved by finding the maximum bandwidth subject to the usability constraints specified in the first objective. The performance of optimization schemes is evaluated using different metrics.

Key words: Information security; Optimization; Digital rights; Watermarking
https://doi.org/10.1631/FITEE.1601479 **CLC number:** TP309

1 Introduction

Statistical databases are constructed from sensitive data, such as medical datasets and email usage datasets, with the aim of (1) concealing the identity of the concerned entity and (2) ensuring that the knowledge is preserved. These databases are then shared among various stakeholders. Since such data is in digital format, it is possible to alter and illegally share (or sell) the data. Accordingly, the data owner (Alice) must enforce ownership rights on the shared data (Wylie and Mineau, 2003; Bertino et al., 2005; Shehab et al., 2008). This demands the insertion of some hidden information as the ownership information, so that an intruder (Mallory) cannot locate and subsequently alter the ownership information. (Throughout this paper, we will let Alice be the data owner and Mallory be the attacker.) In this context, watermarking has been used quite extensively, which embeds some hidden and imperceptible information in data without destroying the contents of the original dataset. The quality of the embedded watermark is quantified by the following two major factors:

1. It must be robust against all possible malicious attacks for the deterioration of the watermark.

2. A watermark must not damage the original dataset and the data usability must be ensured after watermark insertion.

So far, various watermarking techniques have

‡ Corresponding author
ORCID: Muhammad KAMRAN, http://orcid.org/0000-0002-6639-5688

been proposed for different data formats, such as video, images, audio, software, natural languages, text data (Wolfgang and Delp, 1996; Hartung and Kutter, 1999), relational databases (Agrawal and Kiernan, 2002; Atallah et al., 2002; Shehab et al., 2008; Kamran et al., 2013), and electronic medical records (Kamran and Farooq, 2012). However, watermarking of outsourced statistical datasets has not received much attention. Watermarking of such datasets poses a unique challenge, as insertion of watermark into a feature of the dataset may change the predictive ability of that feature. Consequently, the knowledge extracted from the dataset may be invalid. Therefore, watermarking must be optimized while recognizing some usability constraints. Our literature review also shows that optimization techniques, such as genetic algorithms (GA) (Holland, 1992), genetic programming (GP) (Engelbrecht, 2007), particle swarm optimization (PSO) (Kennedy and Eberhart, 1995), and mixed integer nonlinear programming (MINLP) (Grossmann and Kravanja, 1997) are good solvers for optimization problems. Accordingly, the major contributions in this paper are as follows:

1. Watermarking is modeled as a constraint optimization problem and GA, GP, PSO, and MINLP approaches are used to optimize watermark encoding.
2. Watermarking is optimized such that the predictive ability (classification potential) of a feature does not change after encoding.
3. The proposed methodologies are tested on a large number of datasets with varying usability constraints.
4. The robustness of the optimized watermark against various malicious attacks is validated.

2 Related work

Some work closely related to our proposed methodology is the watermarking of relational databases, which primarily deals with the watermarking of numeric features subject to usability constraints. In this context, techniques presented by Agrawal and Kiernan (2002) and Sion et al. (2004) require the presence of a primary key attribute to enforce ownership over the shared data. However, the aforementioned studies did not use any optimization scheme to maximize the available bandwidth.

Although statistical databases need to have a primary key as the focus, it is not to identify each instance distinctly. A number of recent techniques (Zhang et al., 2011; Khanduja and Verma, 2012; Rao et al., 2012; Iqbal et al., 2012; Wang and Gao, 2012) extend the work of Agrawal and Kiernan (2002) and embed a multibit watermark in selected least significant bits (LSBs).

A relevant relational database watermarking technique was proposed by Shehab et al. (2008). The authors modeled the findings of maximum tolerable alterations subject to usability constraints and used GA and pattern search (PS) for optimization. The conclusion that PS performed better than GA was obtained. However, the technique proposed by Shehab et al. (2008) requires the presence of a primary key attribute in the database, which may not be available in the statistical databases.

Recently, Khanduja et al. (2015) proposed a technique in which an image (other files such as audio can also be used for watermarking) was first used to create watermark bits and then embedded via a partition-based approach for watermarking relational databases. Other recent techniques are presented where histogram modulation (Franco-Contreras et al., 2014), semantic properties (Franco-Contreras and Coatrieux, 2015), and variant parts (Rani et al., 2016) of database attributes are used to control the data distortions during the watermarking of relational databases. However, they are not applicable to statistical databases and they do not examine the effectiveness of optimization schemes for ownership protection.

The focus of the above-mentioned techniques is the watermarking of relational databases, and the techniques almost require a primary key for watermarking. However, there is often (if not always) no primary key or any other unique feature in statistical databases. Moreover, the focus of our work is to investigate the performance of optimization schemes to find the maximum available bandwidth for watermarking so that (1) the maximum watermark robustness may be achieved and (2) the knowledge stored in the data is preserved.

3 Overview of the proposed approach

We use various optimization techniques to find the optimum tolerable alteration during watermark embedding. In our scheme the predictive ability of

each feature is considered. Fig. 1 depicts the main architecture of the proposed approach. In the first phase, the dataset is divided into n non-overlapping partitions using the classification potential of the features. Thereafter, watermarking is performed such that the classification potential of each feature presented in the dataset is preserved after the watermark is embedded. This step is conducted to make sure that the optimum watermark does not violate the usability constraints, which is the first objective of the optimization schemes. The formation of the watermark also accounts for the second objective of the optimization scheme: maximizing the available bandwidth to ensure the maximum watermark robustness. In the decoding phase, dataset partition is performed in the same fashion as in the encoding stage. Finally, the watermark is decoded using the decoding parameter and the majority voting scheme to minimize detection errors. An interested reader may find the example run of a watermark embedding and decoding algorithm proposed by Kamran and Farooq (2013). Before detailing our proposed approach, we revisit some of the preliminaries.

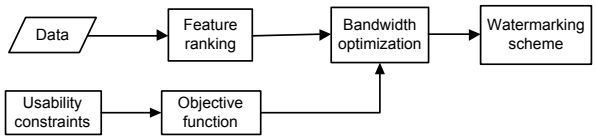

Fig. 1 Architecture of the proposed approach

Definition 1 (Entropy) If a feature α has m distinct values of the feature denoted by $v_1, v_2, \ldots, v_m$ with probabilities $p_1, p_2, \ldots, p_m$, then the smallest number of bits on average needed to transmit a stream of values of α is

$$-\sum_{j=1}^{m} p_j \log_2 p_j.$$

The negative sign is to make the whole quantity positive (because $\log_2$ of the fraction is a negative value). This formula is called Entropy (H).

Definition 2 (Conditional entropy) The conditional entropy $H(R|\alpha=v_j)$ of a feature α is the entropy of R among only those records where $\alpha=v_j$.

Definition 3 (Information gain) The information gain IG of an attribute α is the expected reduction in entropy caused by the partitioning records based on an attribute's value:

$$\mathrm{IG}(R \mid \alpha) = H(R) - H(R \mid \alpha). \tag{1}$$

Definition 4 (Classification potential) The classification potential $C_{P\alpha}$ of an attribute α measures the strength of attribute α to classify all of the records in R in comparison with other attributes:

$$C_{P_\alpha} = \mathrm{IG}(R \mid \alpha) \Big/ \sum_{n=1}^{A} \mathrm{IG}(R \mid n) \times 100, \tag{2}$$

where A denotes the total number of attributes in the dataset.

Definition 5 (Classification statistics) The learning statistics, C_S, is a tuple including the classification statistics (or accuracy) of a particular classification/clustering algorithm.

These statistics include TP, FP, TN, FN, DR, and FAR with

$$\mathrm{DR} = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \times 100\%, \tag{3}$$

$$\mathrm{FAR} = \frac{\mathrm{FT}}{\mathrm{FP+TN}} \times 100\%, \tag{4}$$

where TP, FP, TN, and FN denote the number of true positive instances, the number of false positive instances, the number of true negative instances, and the number of false negative instances, respectively.

We use the usability constraints model proposed by Kamran and Farooq (2013) to enforce usability constraints to find the optimum watermark using different optimization schemes. To make it convenient for the readers of this paper, we provide a brief discussion of these constraints here.

Definition 6 (Local usability constraints) The local usability constraint L_i is a tuple constituting information gain IG of feature X in a particular data group g_i:

$$L_i = \mathrm{IG}. \tag{5}$$

The local usability constraints are used when watermarking features in a group g_i (or partition) and they are enforced for that group only.

Definition 7 (Global usability constraints) Given a

dataset, global usability constraint G is a tuple that consists of a feature set produced by different feature selection schemes on the dataset.

In our case, the following feature selection schemes are used: information gain (IG), information gain ratio (IG_r), correlation-based feature selection (CFS), consistency-based feature selection (CBF), and principal component analysis (PCA).

$$G = (\mathrm{IG}, \mathrm{IG_r}, \mathrm{CFS}, \mathrm{CBF}, \mathrm{PCA}). \quad (6)$$

The global usability constraints are enforced both at a group (or partition) level and at the whole dataset level.

4 Proposed methodology

In this section, the details of the proposed approach are introduced.

4.1 Watermark encoding phase

The steps involved in the watermark encoding phase are as follows:

Step 1: The classification potential of each feature is calculated using IG so that no feature is omitted. The features are then ranked according to their classification potential in decreasing order. The data owner may use feature selection schemes of his/her own choice depending on the nature of the dataset. The rank of each feature is stored in a vector $\boldsymbol{R}$. Step 1 will be further detailed in Section 4.2.

Step 2: The ranks of each feature in $\boldsymbol{R}$ and a secret key κ are used to partition the dataset vertically, in n non-overlapping partitions $\{\mathrm{par}_0, \mathrm{par}_1, \ldots, \mathrm{par}_{n-1}\}$. We discuss step 2 further in Section 4.3.

Step 3: The watermark is optimized and embedded in this stage, which we detail in Section 4.4.

4.2 Feature ranking

This step is important because the predictive ability of any feature must not be affected after watermarking. We choose IG for ranking features because the classification accuracy of a dataset is highly dependent on the IG of a dataset. The ranks of all the features present in the dataset are stored in a vector $\boldsymbol{R}$. During this phase, we also use some other feature selection schemes (defined above), and note their statistics in matrix $\boldsymbol{S}$ so that these statistics may be compared with the same statistics of the watermarked data. It is vital for statistical databases to have correct values for all the relevant features because changing the value of any relevant feature may result in misclassification, which will waste the research efforts for the classification of the dataset. Therefore, while watermarking statistical databases, we must cater to the classification ability of a feature. Moreover, classification rules are sensitive to the values of the features while watermarking statistical databases. Therefore, the changes in the value of watermarked feature(s) should be minimized such that the classification rules are preserved.

4.3 Data partitioning

Statistical databases usually do not have a feature to uniquely identify every single instance (or record). Therefore, the techniques developed so far for partitioning datasets using a unique attribute cannot be applied here. Accordingly, the dataset D with a total of t features $A_0, A_1, \ldots, A_{t-1}$ is divided into n non-overlapping partitions $\{\mathrm{par}_0, \mathrm{par}_1, \ldots, \mathrm{par}_{n-1}\}$. The rank for each feature is used to partition the data vertically by placing the high-ranked attributes in the same partition. These data partitions are logical and are not separated from each other.

4.4 Watermark encoding

We model watermark embedding as a constrained optimization problem subject to two types of constraints: (1) local constraints placed on each partition par_i, denoted by L_i; (2) global constraints G on the overall dataset (Kamran and Farooq, 2013). Along with local and global constraints placed by the data owner, we ensure that the predictive ability of any feature is not disturbed at all as a consequence of watermark embedding. This scenario is handled by the optimization technique. We place local constraints on each partition separately to ensure that the watermarked features belong to the same partition as they did before watermarking. Although the global constraints are there to ensure the overall usability of data after watermark embedding, the objective function used for each optimization scheme is the same. Our optimization problem has two objectives, (1) minimization and (2) maximization, depending on the value of the bit present within the watermark. As our

watermark consists of 0 and 1 bits, our technique considers the constrained problem as a maximization problem when the bit is 0, and a minimization problem when the bit is 1. We use different objectives for different bits to ensure resilience at each inserted bit of the watermark. The objective function for maximization is

$$\max \phi(\text{par}_i+\Delta_i) \tag{7}$$

subject to G and L_i.

When the watermark bit is 1, the above function is used. However, the problem becomes the minimization problem. In the fitness function given above, the value of Δ_i reflects the manipulations made while watermarking the partition par_i.

We now discuss the different optimization schemes used in this study.

4.4.1 Particle swarm optimization algorithm

The particle swarm optimization (PSO) algorithm (Kennedy and Eberhart, 1995) is a stochastic technique developed for continuous optimization, inspired by the social behavior of bird flocks. The solutions to the problem are modeled as particles. Each particle has its 'velocity' and uses particle search for the optimum solutions in the search space. The fitness of each particle is evaluated using a 'fitness function'. The velocity of every particle is updated using two factors: (1) personal best position (autobiographical memory) and (2) global best position (publicized knowledge). After each fitness evaluation, the position of the particle is updated by adding the new velocity value to each component of the current position vector. The PSO algorithm has been implemented efficiently and effectively in function optimization, artificial neural network training, fuzzy system control, and other areas. Moreover, it has been proved better in many respects than evolutionary algorithms such as genetic algorithms, memetic algorithms, ant-colony systems, and shuffled frog-leaping (Elbeltagi et al., 2005).

In our implementation of the PSO algorithm, we map the statistics contained in Δ_i using a bit string having a length of l (the length of the watermark) as depicted in Fig. 2. Since our particle consists of 0 and 1 bits, the constrained problem can be considered as a maximization problem when the bit is 0, and a minimization problem when the bit is 1. The fitness function used is given in Eq. (7).

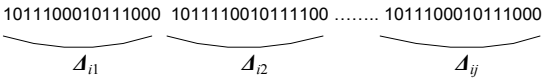

Fig. 2 The particle representation of particle swarm optimization

4.4.2 Genetic algorithm

The genetic algorithm (GA) is inspired by the intelligence model and the principles of biological evolution by the survival of the fittest. The populations of candidate solutions compete to evolve to become better solutions called 'chromosomes'. The GA is a heuristics-based method and was introduced by Holland (1992). A chromosome represents a candidate solution and consists of genes. The chromosome is generally represented as a string of 0's and 1's. Besides, other representation schemes are also used. The fitness of a chromosome is characterized by its fitness value, which is computed by a fitness function. The GA algorithm maintains a population of chromosomes and evolves to produce better solutions by using the GA operators, such as Selection, Crossover, and Mutation. GA encourages better solutions or chromosomes to evolve into the next generation and discourages, by decreasing their fitness value, the chromosomes that do not perform well. The fitness function is chosen according to the nature of the problem. Due to the heuristic nature, GA may not find the optimum solution, but the evolutionary nature makes it a very good candidate for various complex problems. We use GA to create the watermark (chromosome) for embedding in the partitioned dataset. In our implementation of the GA algorithm, the chromosome consists of mapping the feasible Δ_i's into a bit string consisting of 0's and 1's having a length l. Here again, the fitness function given in Eq. (7) is used to measure the fitness of a chromosome (watermark). The representation of the chromosome (watermark) is given in Fig. 3.

If a chromosome violates the usability constraints, it is penalized by decreasing its fitness value. The penalty function is given as

$$\partial_{\text{new}} = \partial_{\text{old}} / (1+\upsilon), \tag{8}$$

where ∂_{new} and ∂_{old} are the new and old fitness values of the chromosome, respectively, and υ is the number of violations of the usability constraints. If there is no violation, then according to Eq. (8), there is no change in the fitness value of the chromosome. Moreover, as the value of υ increases, the fitness of the chromosome decreases. The feasible solution set λ contains all of the chromosomes that satisfy the conditions specified in the usability constraints. Our chromosome representation consists of 0 and 1 bits. Therefore, our technique considers the constrained problem as a maximization problem when the bit is 0, and a minimization problem when the bit is 1. The algorithm stops when a chromosome is found with no violation of the usability constraints, or when a predefined number of generations have been tested. When GA terminates, the chromosome with the best fitness value from the feasible solution set λ is the watermark applied on D. The aforementioned process is applied on each partition that is selected from D for watermarking.

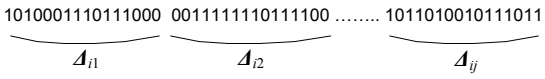

Fig. 3 The chromosome representation of the genetic algorithm

4.4.3 Genetic programming

Genetic programming (GP) (Engelbrecht, 2007) is a specialization of GAs. Each individual is a potential solution for the problem. The operators used in GP are the same as in GA, but there is a slight difference in how these operators are used. In GP, each individual, or chromosome, is represented using a tree structure. A tree consists of a function set and a terminal set. The function set constitutes all of the functions or the operators that can be applied to the elements of the terminal set. The terminal set contains the variables and constants. The functions applied to the terminal set may contain one or more arithmetic, mathematical, and Boolean functions. The elements of the terminal set are placed in the leaf nodes and the elements of the function set create the non-leaf nodes of the tree. The optimum solution found by GP is the evolved tree. In our case, each individual in GP represents a watermark. In a tree, we place the (0, 1) bits in the function set and the leaf nodes contain the values stored in par_i and Δ_i. Here again, our technique considers the constrained problem as a maximization problem when the bit is 0, and a minimization problem when the bit is 1, and uses Eq. (7) as the fitness function. We set the depth of the tree according to the watermark length l. The GP algorithm stops when it finds a solution (GP tree) with no violation of the usability constraints, or when it reaches a predefined number of iterations. The GP tree representation of the watermark is shown in Fig. 4.

4.4.4 Mixed integer nonlinear programming

Mixed integer nonlinear programming (MINLP) is mathematical programming involving continuous and discrete variables, a nonlinear objective function, and constraints. MINLPs have been applied to optimize constrained problems with applications in engineering, financial, and other scientific problems (Grossmann and Kravanja, 1997). Our motivation to use MINLP in our watermarking scheme came from its use in designing algorithms for constrained combinatorial nonlinear mathematical problems in various disciplines. In this study, we work with the outer approximation (OA) method proposed by Duran and Grossmann (1986). The OA method divides MINLP into the NLP subproblem and a master mixed integer program (MIP). An interested reader may refer to Grossmann and Kravanja (1997) for a survey of the MINLP techniques and their applications. We used LINDO (Schrage, 1991) software to implement MINLP for optimizing the nonlinear objective function given in Eq. (7). For this optimization technique, we had to fine-tune the usability constraints so that they can be modeled for MINLP.

4.4.5 Watermark embedding algorithm

The best watermark selected from the vector of optimum watermarks is embedded as the watermark in the partitioned dataset subject to the usability constraints G and L_i. To embed the watermark in the dataset, the contents (bits) of the watermark generated in step 2 (of Algorithm 1) are checked and the watermark is embedded using the mapping of the bit pattern. We consider only the numeric attributes to illustrate the watermark embedding procedure. Our technique inserts the watermark in selected features of each partition. Such features may be selected by the

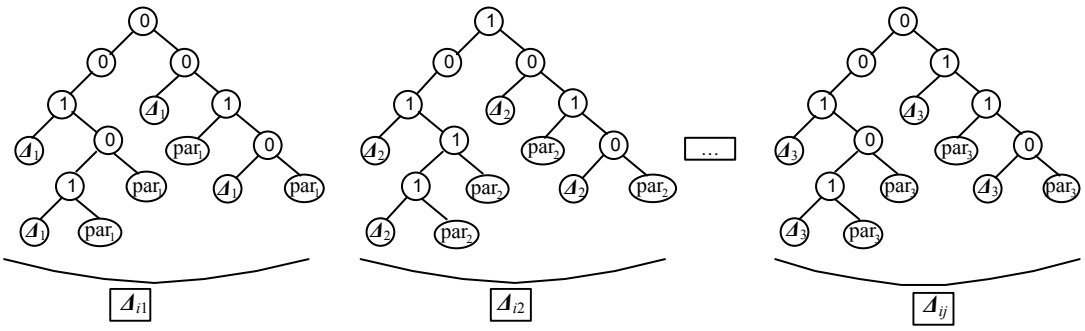

Fig. 4 The genetic programming tree representation

data owner. The percentage change in the value of the watermarked feature is stored in a vector $\boldsymbol{R}$ and used during the decoding process. The watermarking statistics, such as the matrix $\boldsymbol{\Delta}$, the number of partitions n, and the ranks matrix $\boldsymbol{R}$, are also saved during the watermarking embedding process to be used in the decoding stage. Algorithm 1 lists the steps involved in the watermark embedding stage.

Algorithm 1 Embedding watermark

Require: dataset D, number of partitions n, watermark $W=b_0, b_1, \ldots, b_{l-1}$, secret key κ.

Ensure: watermarked dataset D_{W}, watermarking statistics matrix $\boldsymbol{\Delta}$, and features' ranks matrix $\boldsymbol{R}$.

```
Rank features using Eq. (2) to store them in vector R
for b_i=0 to l−1 do
    for each partition 0 to n−1
        if b_i==0 then
            maximize ϕ(par_i+Δ_i) subject to G and L_i
            insert Δ_i into Δ
        end if
        if b_i==1 then
            minimize ϕ(par_i+Δ_i) subject to G and L_i
            insert Δ_i into Δ
        end if
        insert (par_i+Δ_i) into D_W
    end for
end for
return D_W
```

4.5 Watermark decoding

In the watermark decoding process, we check the existence of the embedded watermark in the dataset. We use a watermark decoding scheme that is based on the tolerance threshold ϑ. The tolerance threshold is defined as the amount of change in the data that does not affect the usability of the data. The steps involved in the decoding phase include:

Step 1: The altered data (data after attack) is first partitioned using the same steps as in the watermark encoding phase.

Step 2: For every numeric attribute of all the rows in partition par_i, the watermark bits are detected starting from the LSB and moving toward the MSB (most significant bit). The aforementioned process is carried out using a tolerance threshold.

Step 3: The bits are decoded using the majority voting scheme.

4.5.1 Tolerance threshold

We have used the tolerance threshold to detect the embedded watermark bits. Recall that our watermark embedding has two opposing objectives, maximization and minimization, based on the value of the encoded bit b. In our case, the alteration in the original value of a feature f belonging to the i^{th} partition and the j^{th} row is increased during the maximization process. Consequently, after watermarking, $\boldsymbol{\Delta}_{d(i,j)}$ is always greater than $\boldsymbol{\Delta}_{(i,j)}$, which was calculated during the embedding process. As a result, the value of ϑ (calculated by Eq. (10)) in this case would always be greater than zero, which means that for this particular bit, the optimization problem is a maximization problem. Therefore, we decode the embedded bit b as 0. Similarly, we can detect the embedded bit for the minimization problem to be 1.

The decoding statistics $\boldsymbol{\Delta}_{d(i,j)}$ are calculated by

$$\boldsymbol{\Delta}_{d(i,j)}=\aleph * D_{\mathrm{W}}, \tag{9}$$

where $\aleph$ is a secret parameter known only to the data owner. The value of the decoding parameter is then computed using the relation

$$\vartheta=\boldsymbol{\Delta}_{d(i,j)}-\boldsymbol{\Delta}_{(i,j)}. \tag{10}$$

The tolerance threshold ϑ is used to detect each bit of the embedded watermark. After a bit has been detected, $\Delta_{d(i,j)}$ is incorporated in the altered data. The above process is repeated to detect the next LSB. The watermark decoding steps are depicted in Algorithm 2. In our watermark decoding scheme, for every numeric attribute of all the rows in partition par_i, the watermark bits are detected starting from the LSB and moving toward the MSB. The bits are detected in this order because during the watermark embedding stage, the bits were embedded starting from the MSB to the LSB of the selected watermark.

Algorithm 2 Decoding watermark

Require: dataset D_W, data change matrix Δ, n, ε, κ, l.
Ensure: detected watermark W_D

```
par_0, par_1, …, par_n←DataPartitioning(D_W, R, κ, n)
dt(0, 2, …, l–1)←0
for each partition 0 to n−1
    DataAlterations←0
    for j=1 to size of partition do
        for b=0 to l−1 do
            Compute Δ_d(i,j) and ϑ using Eqs. (9) and (10), respectively
            if ϑ<0 then
                dt(j, b)←1
            else if ϑ>0 and ϑ<ℵ then
                dt(j, b)←0
            else // Data altered by the attacker
                DataAlterations←DataAlterations+1
                dt(j, b)←x
            end if
            // Update the data
            if dt(j, b)==0 then
                D_W(j)=D_W(j)−Δ_d(i,j)
            else
                D_W(j)=D_W(j)+Δ_d(i,j)
            end if
        end for
    end for
    W_D←mode(dt(1, 2, ..., l))
end for
return W_D
```

5 Experiments and results

We performed experiments on 24 different machine learning and data mining datasets available through the UCI Machine Learning Repository (These datasets can be downloaded from http://mlearn.ics.uci.edu/MLRepository.html). These biomedical and biomedicine datasets were carefully chosen from different domains so that we could test our technique for two-class datasets, multiclass datasets, high-dimensional datasets, datasets with missing values, imbalanced datasets, and datasets with a large number of instances. The experiments were performed on a computer with a 1.73 Core 2 processor and 1 GB of RAM. We set the usability constraints so that up to a±2% data change was made while preserving the data quality and the same feature selection scheme results for the datasets both before and after watermarking. Although the data owner can choose a desired watermark length, we performed our experiments with the watermark length l=16 bits. Some classification algorithms were also used to test the classification accuracies for both original and watermarked datasets. Each of the four optimization techniques discussed in this study was tested on each dataset. We also compared the performance of each optimization technique by computing the time taken by each of the techniques for watermark embedding. The robustness of the watermark was also tested under various scenarios for corrupting the watermark.

5.1 Watermark imperceptibility and data quality

In this section, we show that once D is watermarked with a scheme, the watermark not only remains imperceptible but also preserves the classification potential of all the watermarked features. We show the aforementioned feature selection schemes IG, IG_r, CFS, CBF, and PCA on the watermarked dataset. Tables 1–5 show the outputs of the feature selection schemes (IG, IG_r, CFS, CBF, and PCA) when applied on the watermarked dataset. If the optimization scheme, during its execution, did not show any improvement for the predefined number of consecutive iterations, then the algorithm was stopped. Tables 1–5 show that PSO, GA, and GP were all able to find the optimum solution (watermark) where the results of the feature selection schemes were preserved. However, MINLP prematurely converged to some local optima in the solution space for some datasets, especially for the datasets having a large number of features and instances. We believe that the aforementioned phenomenon is observed due to the learning behavior of nature-inspired algorithms (PSO, GA, and GP).

Table 1 Effect of watermarking on the output of information gain (IG)

Dataset	Output			
	D_1	D_2	D_3	D_4
Ann-Thyroid	P	P	P	**NP**
BreastCancer	P	P	P	P
BreastCancerDiagnostic	P	P	P	**NP**
BreastCancerPrognostic	P	P	P	**NP**
Cleveland-Heart	P	P	P	P
ContraceptiveMethod	P	P	P	P
Dermatology	P	P	P	P
Echocardiogram	P	P	P	P
E-Coli	P	P	P	P
Haberman'sSurvival	P	P	P	P
Hepatitis	P	P	P	P
HungarianHeart	P	P	P	P
HyperThyroid	P	P	P	**NP**
Hypo-Thyroid	P	P	P	P
LiverDisorders	P	P	P	P
LymphNodes	P	P	P	P
MammographicMasses	P	P	P	P
NewThyroid	P	P	P	P
PimaIndiansDiabetes	P	P	P	P
Sick	P	P	P	**NP**
StatlogHeart	P	P	P	P
SwitzerlandHeart	P	P	P	P
Thyroid0387	P	P	P	**NP**
VA-Heart	P	P	P	P

D_1, D_2, D_3, and D_4 refer to the data watermarked using PSO, GA, GP, and MINLP, respectively. P means that the output of the feature selection scheme is preserved, whereas NP refers to the output that is not preserved

5.2 Classification-preserving characteristic of the proposed approach

We also tested the effect of the watermarking scheme on the classification accuracy of five carefully selected classifiers from various paradigms. Tables 6 and 7 show that the classification accuracy is also preserved for all of the classifiers with a slight exception for repeated incremental pruning to produce error reduction (RIPPER) (JRIP), where the classification accuracy for watermarked data differs negligibly for some datasets. Tables 6 and 7 show that the overall classification accuracy is also preserved using the proposed approach while watermarking the machine learning datasets.

5.3 Which optimization technique is the fastest?

The time taken by each of the optimization techniques for watermark embedding is given in Table 8. The results are the mean time taken by the algorithms per record when each algorithm was run 20 times. All four techniques took more time for larger datasets and datasets with more features, because the feature selection schemes took more computation time for such a dataset. MINLP was the quickest among all the optimization techniques; however, it converged prematurely to a local optimum and improved the solution quality for several consecutive iterations. Among the three nature-inspired optimization algorithms used in this study, PSO was the fastest, and GA, GP took more time due to their evolutionary nature.

Table 2 Effect of the watermarking scheme on the output of information gain ratio (IG_r)

Dataset	Output			
	D_1	D_2	D_3	D_4
Ann-Thyroid	P	P	P	**NP**
BreastCancer	P	P	P	P
BreastCancerDiagnostic	P	P	P	**NP**
BreastCancerPrognostic	P	P	P	**NP**
Cleveland-Heart	P	P	P	P
ContraceptiveMethod	P	P	P	P
Dermatology	P	P	P	P
Echocardiogram	P	P	P	P
E-Coli	P	P	P	P
Haberman'sSurvival	P	P	P	P
Hepatitis	P	P	P	P
HungarianHeart	P	P	P	P
HyperThyroid	P	P	P	**NP**
Hypo-Thyroid	P	P	P	P
LiverDisorders	P	P	P	P
LymphNodes	P	P	P	P
MammographicMasses	P	P	P	P
NewThyroid	P	P	P	P
PimaIndiansDiabetes	P	P	P	P
Sick	P	P	P	**NP**
StatlogHeart	P	P	P	P
SwitzerlandHeart	P	P	P	P
Thyroid0387	P	P	P	**NP**
VA-Heart	P	P	P	P

The meanings of D_1, D_2, D_3, D_4, P, and NP are the same as those in Table 1

5.4 Robustness

Suppose Alice's watermarked data (D_W) has j instances and she marks her dataset with a watermark W. Mallory wants to corrupt the watermark, but he

does not know anything about D or the secret parameters used for watermarking. Therefore, Mallory is facing a dilemma of corrupting the watermark while keeping the data quality intact. However, he does not have any knowledge about the original data and the inserted watermark. This situation makes it very difficult for Mallory to remove the watermark. Therefore, Mallory can randomly alter the dataset to corrupt the watermark. In general, Mallory may attack the watermark by altering the dataset in three ways:

1. Insert new instance(s);
2. Delete existing instance(s);
3. Modify the existing instance(s) by changing the value(s) of any feature(s).

Consider that Mallory inserts a new instance in D_{W}. The parameter DataAlterations used in our decoding phase helps detect the alterations made by Mallory as a result of inserting a new instance. The probability of detecting such alterations is

$$P(\text{insert}) = 1 - 1/2^{\ln(j+1)}. \tag{11}$$

The second option for Mallory to corrupt the watermark is to delete some random instance(s). Suppose that Mallory deletes a randomly chosen instance. Again, the parameter DataAlterations used in the decoding phase helps detect the instance deleted by Mallory. The probability of detecting such alterations is

$$P(\text{delete}) = 1 - 1/2^{\ln(j-1)}. \tag{12}$$

Let Mallory alter a randomly selected instance by modifying a feature f. Since Mallory does not know anything about the original data, he may violate the usability constraints when modifying the value of the feature. The parameter DataAlterations detects

Table 3 Effect of the watermarking scheme on the output of correlation-based feature selection (CFS)

Dataset	Output			
	D_1	D_2	D_3	D_4
Ann-Thyroid	P	P	P	**NP**
BreastCancer	P	P	P	P
BreastCancerDiagnostic	P	P	P	**NP**
BreastCancerPrognostic	P	P	P	**NP**
Cleveland-Heart	P	P	P	P
ContraceptiveMethod	P	P	P	**NP**
Dermatology	P	P	P	P
Echocardiogram	P	P	P	P
E-Coli	P	P	P	P
Haberman'sSurvival	P	P	P	**NP**
Hepatitis	P	P	P	P
HungarianHeart	P	P	P	P
HyperThyroid	P	P	P	**NP**
Hypo-Thyroid	P	P	P	**NP**
LiverDisorders	P	P	P	P
LymphNodes	P	P	P	P
MammographicMasses	P	P	P	**NP**
NewThyroid	P	P	P	P
PimaIndiansDiabetes	P	P	P	P
Sick	P	P	P	**NP**
StatlogHeart	P	P	P	P
SwitzerlandHeart	P	P	P	P
Thyroid0387	P	P	P	**NP**
VA-Heart	P	P	P	P

The meanings of D_1, D_2, D_3, D_4, P, and NP are the same as those in Table 1

Table 4 Effect of the watermarking scheme on the output of consistency-based feature selection (CBF)

Dataset	Output			
	D_1	D_2	D_3	D_4
Ann-Thyroid	P	P	P	**NP**
BreastCancer	P	P	P	P
BreastCancerDiagnostic	P	P	P	**NP**
BreastCancerPrognostic	P	P	P	**NP**
Cleveland-Heart	P	P	P	P
ContraceptiveMethod	P	P	P	P
Dermatology	P	P	P	P
Echocardiogram	P	P	P	P
E-Coli	P	P	P	P
Haberman'sSurvival	P	P	P	P
Hepatitis	P	P	P	**NP**
HungarianHeart	P	P	P	P
HyperThyroid	P	P	P	**NP**
Hypo-Thyroid	P	P	P	P
LiverDisorders	P	P	P	**NP**
LymphNodes	P	P	P	P
MammographicMasses	P	P	P	P
NewThyroid	P	P	P	P
PimaIndiansDiabetes	P	P	P	**NP**
Sick	P	P	P	**NP**
StatlogHeart	P	P	P	P
SwitzerlandHeart	P	P	P	P
Thyroid0387	P	P	P	**NP**
VA-Heart	P	P	P	**NP**

The meanings of D_1, D_2, D_3, D_4, P, and NP are the same as those in Table 1

the alterations made by Mallory. The probability of detecting such alterations in our detection scheme is

$$P(\text{modify}) = 1 - 1/2^{\ln j}. \quad (13)$$

The proposed method makes the watermarking scheme robust, and the probability that alterations are made by the attacker is dependent on j and the total number of instances. The more instances there are in a dataset, the more difficult it will be for the attacker to corrupt the embedded watermark after using the proposed method for bandwidth optimization.

Table 5 Effect of the watermarking scheme on the output of principal component analysis (PCA)

Dataset	Output			
	D_1	D_2	D_3	D_4
Ann-Thyroid	P	P	P	**NP**
BreastCancer	P	P	P	P
BreastCancerDiagnostic	P	P	P	**NP**
BreastCancerPrognostic	P	P	P	**NP**
Cleveland-Heart	P	P	P	P
ContraceptiveMethod	P	P	P	P
Dermatology	P	P	P	**NP**
Echocardiogram	P	P	P	P
E-Coli	P	P	P	P
Haberman'sSurvival	P	P	P	P
Hepatitis	P	P	P	P
HungarianHeart	P	P	P	P
HyperThyroid	P	P	P	**NP**
Hypo-Thyroid	P	P	P	P
LiverDisorders	P	P	P	P
LymphNodes	P	P	P	P
MammographicMasses	P	P	P	P
NewThyroid	P	P	P	P
PimaIndiansDiabetes	P	P	P	**NP**
Sick	P	P	P	**NP**
StatlogHeart	P	P	P	P
SwitzerlandHeart	P	P	P	P
Thyroid0387	**P**	**P**	**P**	**NP**
VA-Heart	**P**	**P**	**P**	**P**

The meanings of D_1, D_2, D_3, D_4, P, and NP are the same as those in Table 1

6 Conclusions

In this paper, the appropriateness of using optimization schemes for ownership-preserving data mining has been examined. For this purpose, the insertion of a watermark has been modeled as an optimization problem with these objectives: (1) preserving the classification potential of high-ranking features, (2) identifying the maximum available bandwidth while ensuring the usability constraints, and (3) maximizing the watermark robustness by using the available bandwidth. Four different optimization schemes were tested to achieve these objectives. The performance of these schemes was evaluated using different measures. Nature-inspired schemes were found to perform better with more computation time. However, in most cases, watermarking is performed offline and the data owner can afford to compromise the computation time to achieve better results. To our best knowledge, this is the first effort to investigate the use of optimization schemes for ownership-preserving data mining when dealing with statistical databases. In the future, we would like to investigate other optimization schemes such as ant colony optimization (ACO), memetic algorithms, and other classical optimization schemes.

References

Agrawal R, Kiernan J, 2002. Watermarking relational databases. Proc 28th Int Conf on Very Large Databases, p.155-166. https://doi.org/10.1016/B978-155860869-6/50022-6

Atallah MJ, Raskin V, Hempelmann CF, et al., 2002. Natural language watermarking and tamper-proofing. Int Workshop on Information Hiding, p.196-212. https://doi.org/10.1007/3-540-36415-3_13

Bertino E, Ooi BC, Yang Y, et al., 2005. Privacy and ownership preserving of outsourced medical data. Proc 21st Int Conf on Data Engineering, p.521-532. https://doi.org/10.1109/ICDE.2005.111

Duran MA, Grossmann IE, 1986. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Math Programm*, 36(3):307-339. https://doi.org/10.1007/BF02592064

Elbeltagi E, Hegazy T, Grierson D, 2005. Comparison among five evolutionary-based optimization algorithms. *Adv Eng Inform*, 19(1):43-53. https://doi.org/10.1016/j.aei.2005.01.004

Engelbrecht AP, 2007. Computational Intelligence: an Introduction (2nd Ed.). Wiley, New York.

Franco-Contreras J, Coatrieux G, 2015. Robust watermarking of relational databases with ontology-guided distortion control. *IEEE Trans Inform Forens Secur*, 10(9):1939-1952. https://doi.org/10.1109/TIFS.2015.2439962

Franco-Contreras J, Coatrieux G, Cuppens F, et al., 2014. Robust lossless watermarking of relational databases based on circular histogram modulation. *IEEE Trans Inform Forens Secur*, 9(3):397-410. https://doi.org/10.1109/TIFS.2013.2294240

Table 6 Classification accuracy of different classifiers when classifying the original data

Dataset	Classification accuracy (%)					
	J48	SMO	NB	IBk	JRIP	Mean
Ann-Thyroid	99.69	93.79	95.42	94.12	99.53	96.51
BreastCancer	94.56	96.71	95.99	96.99	95.42	95.93
BreastCancerDiagnostic	92.97	97.89	92.62	97.19	93.67	94.87
BreastCancerPrognostic	73.74	75.76	67.17	73.23	75.76	73.13
Cleveland-Heart	55.45	59.08	55.45	59.08	53.14	56.44
ContraceptiveMethod	52.14	48.20	50.78	48.47	52.41	50.40
Dermatology	93.99	95.35	97.27	95.63	86.88	93.82
Echocardiogram	90.84	86.26	87.02	86.26	90.84	88.24
E-Coli	84.23	87.20	85.12	86.01	81.25	84.76
Haberman'sSurvival	71.89	73.53	74.84	69.28	72.22	72.35
Hepatitis	81.94	87.10	83.22	85.16	76.77	82.84
HungarianHeart	68.71	68.02	65.31	66.33	64.63	66.60
HyperThyroid	98.94	97.77	95.39	97.83	98.49	97.68
Hypo-Thyroid	99.24	97.44	97.91	97.28	99.24	98.22
LiverDisorders	68.70	58.26	55.36	59.13	64.64	61.22
LymphNodes	77.03	86.49	83.11	83.78	76.35	81.35
MammographicMasses	82.73	78.88	83.14	80.12	83.25	81.62
NewThyroid	92.09	89.77	96.74	93.95	93.02	93.11
PimaIndiansDiabetes	73.83	77.34	76.30	73.18	75.13	75.16
Sick	98.75	93.86	92.68	95.96	98.21	95.89
StatlogHeart	76.67	82.59	84.81	81.11	77.04	80.44
SwitzerlandHeart	29.27	39.02	35.77	30.89	39.84	34.96
Thyroid0387	95.76	77.77	78.42	81.81	93.93	85.54
VA-Heart	34.00	35.00	34.00	32.00	30.00	33.00
Mean	78.63	78.46	77.66	77.70	77.99	

Table 7 Classification accuracy of different classifiers when classifying watermarked data

Dataset	Classification accuracy (%)					
	J48	SMO	NB	IBk	JRIP	Mean
Ann-Thyroid	99.69	93.79	95.42	94.12	99.54	96.51
BreastCancer	94.56	96.71	95.99	96.99	95.42	95.93
BreastCancerDiagnostic	92.97	97.89	92.62	97.19	93.66	94.87
BreastCancerPrognostic	73.74	75.76	67.17	73.23	75.76	73.13
Cleveland-Heart	55.45	59.08	55.45	59.08	53.15	56.44
ContraceptiveMethod	52.14	48.20	50.78	48.47	52.44	**50.41**
Dermatology	93.99	95.35	97.27	95.63	86.89	**93.83**
Echocardiogram	90.84	86.26	87.02	86.26	90.86	**88.25**
E-Coli	84.23	87.20	85.12	86.01	81.25	84.76
Haberman'sSurvival	71.89	73.53	74.84	69.28	72.23	72.35
Hepatitis	81.94	87.10	83.22	85.16	76.77	82.84
HungarianHeart	68.71	68.02	65.31	66.33	64.63	66.60
HyperThyroid	98.94	97.77	95.39	97.83	98.49	97.68
Hypo-Thyroid	99.24	97.44	97.91	97.28	99.23	98.22
LiverDisorders	68.70	58.26	55.36	59.13	64.64	61.22
LymphNodes	77.03	86.49	83.11	83.78	76.35	81.35
MammographicMasses	82.73	78.88	83.14	80.12	83.25	81.62
NewThyroid	92.09	89.77	96.74	93.95	93.02	93.11
PimaIndiansDiabetes	73.83	77.34	76.30	73.18	75.13	75.16
Sick	98.75	93.86	92.68	95.96	98.21	95.89
StatlogHeart	76.67	82.59	84.81	81.11	77.04	80.44
SwitzerlandHeart	29.27	39.02	35.77	30.89	39.84	34.96
Thyroid0387	95.76	77.77	78.42	81.81	93.93	85.54
VA-Heart	34.00	35.00	34.00	32.00	30.00	33.00
Mean	78.63	78.46	77.66	77.70	77.99	

Table 8 Time taken by different optimization schemes for watermark embedding

Dataset*	Time (s)			
	PSO	GA	GP	MINLP
Ann-Thyroid	0.0180	0.4320	0.0540	0.01440
BreastCancer	0.0096	0.2304	0.0288	0.00768
BreastCancerDiagnostic	0.0101	0.2424	0.0303	0.00808
BreastCancerPrognostic	0.0101	0.2424	0.0303	0.00808
Cleveland-Heart	0.0098	0.2352	0.0294	0.00784
ContraceptiveMethod	0.0099	0.2376	0.0297	0.00792
Dermatology	0.0096	0.2304	0.0288	0.00768
Echocardiogram	0.0098	0.2352	0.0294	0.00784
E-Coli	0.0098	0.2352	0.0294	0.00784
Haberman'sSurvival	0.0096	0.2304	0.0288	0.00768
Hepatitis	0.0098	0.2352	0.0294	0.00784
HungarianHeart	0.0098	0.2352	0.0294	0.00784
HyperThyroid	0.0101	0.2424	0.0303	0.00808
Hypo-Thyroid	0.0100	0.2400	0.0300	0.00800
LiverDisorders	0.0098	0.2352	0.0294	0.00784
LymphNodes	0.0096	0.2304	0.0288	0.00768
MammographicMasses	0.0096	0.2304	0.0288	0.00768
NewThyroid	0.0097	0.2328	0.0291	0.00776
PimaIndiansDiabetes	0.0098	0.2352	0.0294	0.00784
Sick	0.0099	0.2376	0.0297	0.00792
StatlogHeart	0.0098	0.2352	0.0294	0.00784
SwitzerlandHeart	0.0098	0.2352	0.0294	0.00784
Thyroid0387	0.0102	0.2448	0.0306	0.00816
VA-Heart	0.0096	0.2304	0.0288	0.00768
Mean	0.0101	0.2435	0.0304	0.00810

* These datasets can be downloaded from http://mlearn.ics.uci.edu/MLRepository.html

Grossmann IE, Kravanja Z, 1997. Mixed-integer nonlinear programming: a survey of algorithms and applications. In: Biegler LT, Coleman TF, Conn AR, et al. (Eds.), Large-Scale Optimization with Applications. Part II: Optimal Design and Control. Springer, New York, NY, p.73-100. https://doi.org/10.1007/978-1-4612-1960-6_5

Hartung F, Kutter M, 1999. Multimedia watermarking techniques. *Proc IEEE*, 87(7):1079-1107. https://doi.org/10.1109/5.771066

Holland, JH, 1992. Adaptation in Natural and Artificial Systems: an Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. MIT Press, Cambridge, MA.

Iqbal S, Rauf A, Mahfooz S, et al., 2012. Self-constructing fragile watermark algorithm for relational database integrity proof. *World Appl Sci J*, 19(9):1273-1277. https://doi.org/10.5829/idosi.wasj.2012.19.09.1865

Kamran M, Farooq M, 2012. An information-preserving watermarking scheme for right protection of EMR systems. *IEEE Trans Knowl Data Eng*, 24(11):1950-1962. https://doi.org/10.1109/TKDE.2011.223

Kamran M, Farooq M, 2013. A formal usability constraints model for watermarking of outsourced datasets. *IEEE Trans Inform Forens Secur*, 8(6):1061-1072. https://doi.org/10.1109/TIFS.2013.2259234

Kamran M, Suhail S, Farooq M, 2013. A robust, distortion minimizing technique for watermarking relational databases using once-for-all usability constraints. *IEEE Trans Knowl Data Eng*, 25(12):2694-2707. https://doi.org/10.1109/TKDE.2012.227

Kennedy J, Eberhart R, 1995. Particle swarm optimization. Proc IEEE Int Conf on Neural Networks, p.1942-1948. https://doi.org/10.1109/ICNN.1995.488968

Khanduja V, Verma OP, 2012. Identification and proof of ownership by watermarking relational databases. *Int J Inform Electron Eng*, 2(2):274-277. https://doi.org/10.7763/IJIEE.2012.V2.97

Khanduja V, Chakraverty S, Verma OP, 2015. Watermarking categorical data: algorithm and robustness analysis. *Def Sci J*, 65(3):226-232. https://doi.org/10.14429/dsj.65.8444

Rani S, Kachhap P, Haider R, 2016. Dataflow analysis-based

approach of database watermarking. In: Chaki R, Cortesi A, Saeed K (Eds.), Advanced Computing and Systems for Security, Volume 2. Springer, New Delhi, p.153-171. https://doi.org/10.1007/978-81-322-2653-6_11

Rao UP, Patel DR, Vikani PM, 2012. Relational database watermarking for ownership protection. *Proc Technol*, 6:988-995. https://doi.org/10.1016/j.protcy.2012.10.120

Schrage LE, 1991. LINDO: an Optimization Modeling System (4th Ed.). Scientific Press, South San Francisco, CA.

Shehab M, Bertino E, Ghafoor A, 2008. Watermarking relational databases using optimization-based techniques. *IEEE Trans Knowl Data Eng*, 20(1):116-129. https://doi.org/10.1109/TKDE.2007.190668

Sion R, Atallah M, Prabhakar S, 2004. Rights protection for relational data. *IEEE Trans Knowl Data Eng*, 16(12): 1509-1525. https://doi.org/10.1109/TKDE.2004.94

Wang YM, Gao YX, 2012. The digital watermarking algorithm of the relational database based on the effective bits of numerical field. World Automation Congress, p.1-4.

Wolfgang RB, Delp EJ, 1996. A watermark for digital images. Proc 3rd IEEE Int Conf on Image Processing, p.219-222. https://doi.org/10.1109/ICIP.1996.560423

Wylie JE, Mineau GP, 2003. Biomedical databases: protecting privacy and promoting research. *Trends Biotechnol*, 21(3):113-116. https://doi.org/10.1016/S0167-7799(02)00039-2

Zhang LZ, Gao W, Jiang N, et al., 2011. Relational databases watermarking for textual and numerical data. Int Conf on Mechatronic Science, Electric Engineering and Computer, p.1633-1636. https://doi.org/10.1109/MEC.2011.6025791