Information Sciences 179 (2009) 1663–1677

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Towards the evaluation of time series protection methods

Jordi Nin *, Vicenç Torra

IIIA, Artificial Intelligence Research Institute CSIC, Spanish National Research Council Campus UAB s/n, 08193 Bellaterra, Catalonia, Spain

ARTICLE INFO

Article history:
Received 6 February 2008
Received in revised form 20 January 2009
Accepted 21 January 2009

Keywords:
Statistical disclosure control
Information loss measures
Time series privacy
Privacy in statistical databases

ABSTRACT

The goal of statistical disclosure control (SDC) is to modify statistical data so that it can be published without releasing confidential information that may be linked to specific respondents. The challenge for SDC is to achieve this variation with minimum loss of the detail and accuracy sought by final users. There are many approaches to evaluate the quality of a protection method. However, all these measures are only applicable to numerical or categorical attributes.

In this paper, we present some recent results about time series protection and re-identification. We propose a complete framework to evaluate time series protection methods. We also present some empirical results to show how our framework works.

1. Introduction

A common practice in many organizations is to manage large volumes of confidential data. In many cases, such data contains valuable statistical information that third parties might require for performing certain analyses. Nevertheless, when data has to be transferred to such third parties, privacy becomes an essential issue. In this situation, it is necessary to release data, such that preserving certain statistics, do not reveal confidential information. This is a usual problem, for instance, in the National Statistics Offices.

Privacy Preserving Data Mining (PPDM) [4] is a closely related research line to SDC where privacy is also involved. PPDM tackles the problem of developing data mining techniques in which the privacy of the individuals is preserved. In a very similar way to SDC, PPDM modifies individual data records in such a way that the results of a mining process are (almost) the same as those obtained when using real data. Therefore, PPDM methods seek to protect the anonymity of data owners when a data analysis technique is computed on the protected data set. See [40,53] for recent examples of PPDM techniques.

Privacy and anonymity technologies rise as a very important issue in the Internet scenario, where thousands of data users access a large variety of information, see [25] for an up-to-date survey about data access in the Internet.

During the past few years, special efforts have been made to develop a wide range of protection methods. Good surveys about data SDC methods can be found in the literature [1,11]. However, all the research presented in these surveys focus on the anonymization of numerical and categorical data.

However, in the real world, an increasing percentage of the released information has an implicit or explicit time component. This is the case of e.g. income or stock prices. Similarly, data accumulation through consecutive years (e.g. economical data from companies or census data from individuals) can also be considered from this point of view. Standard protection methods have been designed for non-temporal attributes and they ignore many key questions regarding time series as

* Corresponding author.
E-mail addresses: jnin@iiia.csic.es (J. Nin), vtorra@iiia.csic.es (V. Torra).

0020-0255/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2009.01.024

e.g. time series normalization, preservation of time information. In general, methods ignore the standard uses specific to time series as e.g. forecasting or tendency analysis.

The main goal of protection methods, regardless of the type of data to be protected, is to minimize both the *disclosure risk* (DR) and the *information loss* (IL) of the protected released data. Disclosure risk measures the capacity of an intruder to obtain some information about the original data set from the protected one, while information loss measures the reduction of the statistical utility of the protected microdata with respect to the original one. However, they are contradictory objectives: when one of them decreases the other one increases. Finding the optimal combination of these two measures becomes a difficult and challenging task.

Few approaches have been used to calculate information loss. In [10], the authors calculate the average difference between some statistics computed on both the original and the protected microdata. A probabilistic variation of this measure was presented in [28] to ensure that the information loss value is always inside the interval $[0,1]$. A different approach was presented in [6], where some measures (accuracy, completeness and consistency) are calculated over the protected data to evaluate the information loss. Nevertheless, all these measures are only useful when data does not present temporal characteristics.

The computation of disclosure risk is based on several works as e.g. [10,39,52] on *record linkage* methods [45,50,51]. Such methods try to model a situation where an intruder gets the protected data set while at the same time he has access to some records of the original data set obtained from other sources. The goal of such methods (i.e. of the intruder) is to link the original records with the corresponding records in the protected data set. Of course, the more records can be linked (by means of these record linkage methods), the more disclosure risk has the employed protection method. Some examples of standard record linkage methods are distance based ones and probabilistic ones. Again, neither distance based nor probabilistic record linkage have been designed to link time series. Of course, other approaches can be considered, such as *k*-anonymity [44], *p*-sensitive [48] or *l*-diversity [26]. However, by using record linkage methods a data holder knows in an empirical way the amount of records with a high risk of re-identification and he can apply counter measures on such records to reduce their disclosure risk. Note also that disclosure risk computation is very dependent on the protected data. For instance in [41], authors consider a specific disclosure risk measure for tabular data when tables of counts are released.

In order to quantify the trade-off between information loss and disclosure risk, these measures are combined to obtain a global value about a specific protection method, this value weighs the relationship between IL and DR. One approach was presented in [10], where the authors combine both information loss and disclosure risk in a *Score* using the arithmetic mean. A different approach is the R-U (risk-utility) maps [16–18]. This measure shows in a graphical way the relationship between any numerical measure of statistical disclosure risk (R) and any numerical measure of data utility (U). The key point in all these combinations is that IL and DR measures have to properly represent the utility and the risk of the protected data. This goal can be only achieved with measures that take into account the data type of the data being protected. Therefore, standard IL and DR measures are unsuitable for the evaluation of time series.

In [31], we developed a method for time series protection. Such method is based on microaggregation [12], that presents a good performance for standard numerical data with respect to information loss and disclosure risk measures as shown in [11]. Microaggregation is one of the standard methods for database protection commonly used in National Statistical Offices (see e.g. [19]).

The basic implementation of microaggregation works as follows [11,12,38]: given a data set with *a* attributes, small clusters of at least *k* elements (records) are built and each original record is replaced with the centroid of the cluster to which the record belongs to. A certain level of privacy is ensured since *k* records have an identical protected value (*k-anonymity* [37,43,44]).

However, when *a* is large, the statistical utility of the basic microaggregation technique is diminished, especially if the attributes are not highly correlated [2]. This is so because the larger the number of attributes, the larger the distance between the original records in the data set and their corresponding centroids. Therefore, a lot of information on the original data is lost when the protected microdata file is released. To solve this drawback, statistical agencies apply the following natural strategy: data set is split into smaller blocks of attributes, and microaggregation is independently applied to each block. This way, the information loss decreases, at the cost of decreasing the achieved level of privacy since the property of *k*-anonymity is not ensured, as we see later on in this paper. This kind of microaggregation methods are known as *multivariate microaggregation* methods. Another important drawback of this type of methods is that finding the optimal multivariate microaggregation (i.e. finding the clusters that minimize the sum of square errors) is NP-hard [35].

Microaggregation requires the definition of a distance on the data. For standard data, the usual distance is the Euclidean distance. In the case of time series, several distances on time series can be considered. We have used two different distances: short time series distance [49] and Euclidean distance.

In [32], we present a detailed study about the application of time series distances to record linkage methods. The main conclusion of [32] is that an intruder can disclose more information if he exploits the fact that the protected information consists of time series. In this case, the intruder can take profit from the time information stored in the time series.

To the best of our knowledge, neither information loss nor disclosure risk measures are described for the case of time series protection. In this paper, we propose a group of information loss measures designed for time series protection evaluation. Such measures consider the main uses of time series, e.g. forecasting and autocorrelation analysis. We also propose the use of the record linkage methods presented in [32] as the most straightforward way to compute the disclosure risk.

Finally, we propose to combine both IL and DR measures in a final score using the arithmetic mean. This approach is also used in [10].

1.1. Organization of the paper

In Section 2, we review some basic concepts related to time series statistics, distances and forecasting. In Section 3, we discuss some aspects related to protection methods in general and microaggregation in particular. Sections 4 and 5 are devoted, respectively to study information loss and disclosure risk measures for time series. Section 6 discusses about the combination of information loss and disclosure risk to provide a general evaluation of the time series protection methods. Section 7 presents some results of our framework using real data. Finally, Section 8 draws some conclusions and describes some lines for future work.

2. Preliminaries

2.1. Time series

Numerical time series are defined by pairs $\{(x_i, t_i)\}$ for $i = 1, \ldots, n$ where t_i corresponds to the temporal variable and x_i is the numerical variable that depends on time (dependent variable). Consequently, $t_{i+1} > t_i$. Income, stock prices and sport statistics are examples of time series, as they depend on time.

We can define in the same way ordinal or categorical time series replacing x_t with a categorical or ordinal variable. Weather forecast (e.g. sunny, cloudy, raining) and restaurant categories (e.g. one Michelin star, two Michelin stars, three Michelin stars) are examples of categorical and ordinal time series respectively. In this work, we will only consider numerical time series.

In this work, we will adopt the following assumptions: time series are discrete, the observations are made at fixed time intervals and all time series have the same initial time t_0. Under these assumptions, it is possible to simplify the notation disregarding the temporal variable. Therefore, hereinafter, our notation for a time series will be $(x_1, \ldots, x_n)$.

Certain time series statistics have been defined. In this work we will use the two most common ones: the *time series mean* and the *autocorrelation function*. The reason for this selection is that both statistics are involved in the ARMA and ARIMA processes [7], two well-known processes for time series modeling. Both statistics are defined as follows [8]:

- **Time series mean.** It is defined by

$$\mu = \frac{1}{n}\sum_{i=0}^{n} x_i,$$

where n corresponds to the number of elements of the time series.

- **Autocorrelation function (ACF).** It describes the correlation between the process at different times. For discrete time series of length n, it is defined by

$$R(j) = \frac{\sum_{i=1}^{n-k}(x_i - \mu)(x_{i+j} - \mu)}{(n-k)\sigma^2},$$

where n corresponds to the number of elements of the time series and i and $i + j$ are the initial elements for computing the correlation. It is usual to use $i = 0$ with j being a given shift.

2.2. Time series distances

In the literature we can find a large number of distances for time series. See [9,29,49] for more details.

Here, we only describe the distances used in this paper for computing the disclosure risk.

- **Euclidean distance (EU).** It is defined as

$$d_{EU}(x, v) = \sqrt[2]{\sum_{k=1}^{n}(x_k - v_k)^2}.$$

- **Short time series distance (STS).** It was defined in [29] as the square root of the sum of the slope squared differences. Formally, it is defined as follows:

$$d_{STS}(x, v) = \sqrt[2]{\sum_{k=1}^{n}\left(\frac{v_{k+1} - v_k}{t_{k+1} - t_k} - \frac{x_{k+1} - x_k}{t_{k+1} - t_k}\right)^2}.$$

2.3. Time series forecasting

Forecasting is a process that uses a set of historical values to predict an outcome. It is commonly used in time series to predict future values of a given time series. Good surveys on forecasting are [5,36]. We explain herein five well-known forecasting models widely used in real applications.

All forecasting models estimate future values using the previous elements of time series. For instance, given a time series $(x_1, \dots, x_n)$, we can estimate the value x_{n+1}. In this case, $(x_1, \dots, x_n)$ are independent values of the forecasting model, whereas x_{n+1} is the dependent one. This process can be repeated using x_{n+2} as the dependent value and adding the *estimated* x_{n+1} value to the independent ones.

2.3.1. Simple exponential smoothing forecasting model

This is a very popular model used to produce smoothed time series. Simple exponential smoothing (*SESF*) assigns exponentially decreasing weights as the observations get older. In other words, recent observations are given relatively more weight in forecasting than the older ones.

2.3.2. Double exponential smoothing forecasting model

The double exponential smoothing (*DESF*), also known as Holt exponential smoothing, is a refinement of the previous one after adding a component which includes any trend in data. Simple exponential smoothing models work better with data with no trend or seasonality components. For this reason, when the data exhibits either an increasing or decreasing trend over time, simple exponential smoothing forecasts tend to fall behind observations. Double exponential smoothing has been designed to address this type of time series by considering the existing trends in the data.

2.3.3. Linear regression forecasting model

This is a regression model (*RM*) where a dependent variable y is expressed in terms of an independent variable x and a random term ϵ as follows:

$$y = \beta_0 + \beta_1 x + \varepsilon,$$

where β_0 is the intercept ('constant' term) and β_1 is the parameter of the independent variable. This model can be used for forecasting, if x refers to previous values of the variable and y to the ones to be forecasted.

2.3.4. Multiple linear regression forecasting model

This is an extension of the linear regression model. In this case, there is a dependent variable y and several independent variables $x_i, i = 1, \dots, p$, and a random term ϵ. The model (*MLRF*) is as follows:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \varepsilon,$$

where β_0 is the intercept ('constant' term), the β_i are the respective parameters of independent variables, and p is the number of parameters to be estimated in the linear regression.

2.3.5. Polynomial regression forecasting model

The linear regression forecasting model (a first-order polynomial) can be extended to higher orders. The polynomial regression model (*PRM*) $y_i = \alpha_0 + \alpha_1 x_i + \alpha_2 x_i^2 + \cdots + \alpha_m x_i^m + \varepsilon_i$ $(i = 1, 2, \dots, n)$ is a system of polynomial equations of order m with coefficients $\{\alpha_0, \dots, \alpha_m\}$. This model can be expressed by using a data matrix X, a target vector $\vec{y}$ and a parameter vector $\vec{\alpha}$. The ith rows of X and $\vec{y}$ contain the x and y values for the ith data sample. In this way, the model can be written as a system of linear equations:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^m \\ 1 & x_2 & x_2^2 & \dots & x_2^m \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^m \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix},$$

thus, using pure matrix notation is, as aforementioned,

$$Y = \mathbf{X}\vec{\alpha} + \varepsilon.$$

Given $\mathbf{X}$ and Y, the vector of polynomial coefficients is determined using the following expression.

$$\widehat{\vec{\alpha}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T Y.$$

3. Protection methods

A data set X can be seen as a matrix with n rows (*records*) and a columns (*attributes*). Each row contains the values of the attributes for each individual. Each attribute can be classified as *identifier*, when they unambiguously identify the individual (e.g. its passport number), or *quasi-identifier* otherwise (e.g. its age). Note, however, that some combinations of quasi-identifier attributes can identify an individual depending on the scenario. Quasi-identifier attributes can be divided in two classes: *confidential* and *non-confidential* depending on whether they contain private information. For instance, the zip code is non-confidential, whereas the salary can be a confidential attribute.

Normally, before releasing a data set with confidential attributes, a protection method ρ is applied, leading to a protected data set X'. Indeed, we will assume the following typical scenario: (i) identifier attributes in X are either removed or encrypted, therefore we will write $X = X_{nc}||X_c$; (ii) confidential quasi-identifier attributes X_c are not modified, so we have $X'_c = X_c$; (iii) the protection method itself is applied to non-confidential quasi-identifier attributes in order to preserve the privacy of the individuals whose confidential data is being released. Therefore, we have $X'_{nc} = \rho(X_{nc})$.

This scenario, used for the first time in [11] to compare several protection methods, has also been adopted in other works [33,34,47,51,52].

3.1. Microaggregation

As introduced in Section 1, microaggregation ensures *k-anonymity* by building small clusters of at least k elements and replacing the original values with the centroid of the cluster to which the record belongs to.

Of course, there are other ways to achieve k-anonymity. For instance, in [3] authors present a clustering technique where the released data set preserves k-anonymity, as in basic microaggregation. In other solutions, such as those presented in [20], the data holder chooses different subsets of attributes ensuring k-anonymity for each of these subsets separately, similar to multivariate microaggregation.

We have seen that, in order to solve the information loss problem of the basic microaggregation method, *multivariate microaggregation* is used at the cost of increasing the disclosure risk. Specifically, after dividing attributes into different blocks and applying the basic microaggregation technique to each block separately, the k records which fall in the same cluster for the first block of attributes, may fall in a different cluster for any of the other blocks of attributes. So, the resulting protected records will not be equal and no k-anonymity is ensured.

In order to preserve information loss as low as possible, microaggregation methods try to minimize the total sum of distances between all the elements to be protected and the centroid of the cluster where an element belongs to, i.e. minimize the total Sum of Square Errors (SSE):

$$\mathrm{SSE} = \sum_{i=1}^{c} \sum_{x_{ij} \in C_i} (x_{ij} - \bar{x}_i)^T (x_{ij} - \bar{x}_i),$$

where c is the total number of clusters, C_i is the ith cluster and $\bar{x}_i$ is the centroid of C_i. The restriction is $|C_i| \geqslant k$, for all $i = 1, \ldots, c$. In general, the larger value of k the lower the disclosure risk. Therefore, in order to parametrize microaggregation methods, k has to be as large as possible without compromising the statistical utility of the protected information.

The rationale of this process is to make the protected data as similar as possible to the original one. In any case, methods should provide clusters with at least k elements. As introduced before, finding the optimal multivariate microaggregation has been proven to be an NP-Hard problem. For this reason, heuristic methods have been proposed in the literature.

For our work, we will adapt for time series protection one of the most used methods for heuristic multivariate microaggregation: the MDAV (Maximum Distance to Average Vector) algorithm.

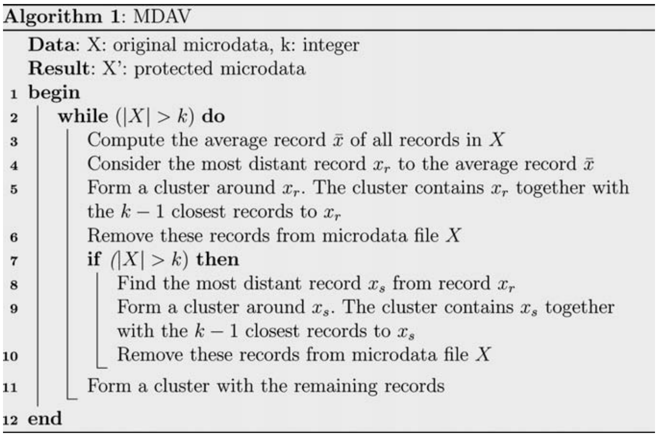
3.2. MDAV microaggregation

The MDAV (Maximum Distance to Average Vector) algorithm[27] is an heuristic algorithm for clustering records in a data set X so that each cluster is constrained to contain at least k records. The MDAV algorithm is described in Algorithm 1.

MDAV generic algorithm can be instantiated for different data types, using appropriate definitions for distance and average. More specifically, for its application to a particular type of data, we need to formulate what the *most distant record* means, and which the *closest records* of a given record are. Additionally, we need to define *the average record* of a set of records. This average record is needed both in step (3) and to mask the original data (5 and 9). Recall that we need to build a representative for each cluster and then replace each original record with the corresponding representative.

In [12], this method is applied to numerical data, using the Euclidean distance for computing the distance between records and the arithmetic mean to compute the average. In [13,46], this method was extended to categorical data using appropriate functions.

In [14], MDAV was extended to be p-sensitive k-anonymity [48] when the intruder only knows data coming from one microaggregated group.



```
Algorithm 1: MDAV
  Data: X: original microdata, k: integer
  Result: X': protected microdata
begin
    while ($|X| > k$) do
        Compute the average record $\bar{x}$ of all records in $X$
        Consider the most distant record $x_r$ to the average record $\bar{x}$
        Form a cluster around $x_r$. The cluster contains $x_r$ together with
        the $k - 1$ closest records to $x_r$
        Remove these records from microdata file $X$
        if ($|X| > k$) then
            Find the most distant record $x_s$ from record $x_r$
            Form a cluster around $x_s$. The cluster contains $x_s$ together
            with the $k - 1$ closest records to $x_s$
            Remove these records from microdata file $X$
        Form a cluster with the remaining records
end
```

3.3. Time series microaggregation

To specialize the MDAV algorithm for time series we need to establish which distance and which average function will be used. We propose to implement the general MDAV algorithm described in Algorithm 1 with the following parameterizations:

- **Distance functions.** We propose the use of Euclidean and STS distances: $d_{EU}(x, v)$ and $d_{STS}(x, v)$ as defined in Section 2.1.
- **Average.** We propose to use a kind of arithmetic mean. Such mean has been defined component-wise. That is, given the set $X = \{x^j\}_{j=1,\dots,J}$ with time series x^j for $j = 1, \dots, J$, each one with x^j_k, we define the average series $\tilde{x}_k$ by $\tilde{x}_k = (1/J)\sum_{j=1,\dots,J} x^j_k$.

With these definitions, the average record $\tilde{x}$ in the MDAV algorithm is the average of all records (time series) in X.

The two distance functions considered (Euclidean and STS distances) lead to different results when combined with the microaggregation algorithm. While the Euclidean distance makes clusters based on the distance between data components, the STS distance makes clusters based on the shape of the time series. This is illustrated in the following example.

Example 1. Fig. 1 (left) represents 4 series to be microaggregated. The results of microaggregating these 4 series into 2 clusters using either Euclidean or STS distances are given, respectively, in middle and right chart of Fig. 1. It can be observed that the Euclidean distance gathers together the nearest series even in case they have different shapes (and, thus, the outcomes are just lines but that mainly keep the original values). In contrast, the STS distance gathers series according to shapes (and, thus, the outcomes keep such shapes but not the original position of the series).

In this example, we have used point-wise average for computing the representative of each cluster.

According to this, in the step of selecting the distance function, we have the opportunity to model how the microaggregation procedure makes the clusters and decide which information is the most important to be kept in the final protected model.

Note that our time series microaggregation enjoys k-anonymity as in the case of MDAV when the intruder has only access to one time series. Note also that, if we are interested in protecting time series by ensuring the p-sensitive k-anonymity, we can use the improved version of MDAV presented in [14] but using the distance functions and averaging method described in this paper.

In the following we will use *eu*-microaggregation to denote the microaggregation based on the Euclidean distance and *sts*-microaggregation to denote the microaggregation based on the STS distance.

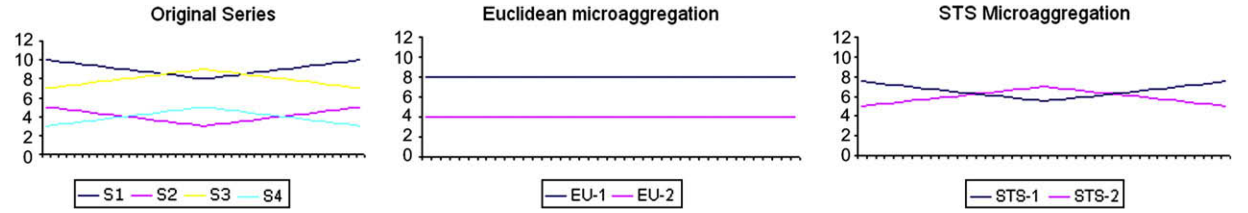


Fig. 1. Graphical representation of distance function selection.

4. Time series information loss measures

Strictly speaking, information loss depends on the data uses to be supported by the protected data. However, potential data uses are very diverse and it could be even hard to identify them in case of release of data. It is thus desirable for the data protector to be able to measure information loss in a generic way. Information loss measures should reflect how much perturbation is added by a given protection method. The amount of information loss measured in this generic way should roughly correspond to the amount of information loss for a reasonable range of data uses. When one defines the measure from a set of components, we need such components to cover (almost) all the possible data uses of a generic user.

In our scenario for time series protection, information loss components have to cover a broad variety of uses, ranging from the statistical analysis to forecasting. For this reason, we divide the information loss components into three different categories:

- IL_1. Measures related to statistical analysis. Such measures, as the average or the autocorrelation function, cover part of the typical statistical analysis like ARMA or ARIMA processes [7].
- IL_2. Measures related to the differences among original and protected time series. It is clear that information loss increases if protected elements are 'far' (dissimilar) from the original ones.
- IL_3. Measures related to forecasting. As forecasting is one of the most common uses of time series, we can say that the statistical information is preserved when the forecast from protected time series is similar to the forecast using the original data.

4.1. Information loss computation

We have defined the general information loss in terms of the three different components described above IL_1, IL_2 and IL_3:

$$IL = \frac{IL_1 + IL_2 + IL_3}{3}.$$

We formally define IL_1, IL_2 and IL_3 below. These three measures are calculated using the differences between values obtained from the original and the protected data. It is possible to define such differences in different ways (e.g. by using the mean square error or the mean absolute error). However, as we want to obtain a value in the $[0,1]$ interval, we define the IL_i measures as mean variations dividing the differences by the largest value (original or protected) to ensure that the result is always inside the rank $[0,1]$. We denote γ by the original statistic value and γ' by the same statistic computed in the protected data set.

- IL_1. It is defined as the average of the difference between the time series means and the autocorrelation functions of both original and protected time series. Formally, IL_1 is computed using the formula

  $$IL_1 = \frac{IL_{1.1} + IL_{1.2}}{2},$$

  where $IL_{1.1}$ and $IL_{1.2}$ correspond, respectively, to

  $$IL_{1.1} = \frac{\sum_{i=1}^{s} \frac{(|\mu_i| - |\mu'_i|)}{\mathrm{Max}(|\mu_i|, |\mu'_i|)}}{s},$$

  $$IL_{1.2} = \frac{1}{4} \sum_{i=0,n/4,n/2,3n/4} \left(\frac{\sum_{i=1}^{s} \frac{|(R_i| - |R'_i|)|)}{Max(|R_i|, |R'_i|)}}{s} \right),$$

  where s is the number of series in the data set and n is the number of elements (length) of the time series.
- IL_2. It is defined in terms of the absolute differences between original and protected time series elements

  $$IL_2 = \frac{\sum_{i=1}^{s \times n} \frac{|x_i - x'_i|}{\mathrm{Max}(|x_i|, |x'_i|)}}{s \times n}.$$

- IL_3. It is defined using the differences between different forecasting models for the $n+1$, $n+2$ and $n+3$ values

  $$IL_3 = \frac{\sum_{m \in FM} \frac{\sum_{i=1}^{3} \frac{|x_{n+i} - x'_{n+i}|}{\mathrm{Max}(x_{n+i}, x'_{n+i})}}{3}}{5},$$

  where FM is the set containing all the forecasting models described in Section 2.3. Then, FM is defined as $FM = \{SESF, DESF, RF, MLRF, PRF\}$. In this measure, we can add other forecasting models, such as non linear forecasting models [21] or neural networks based models [22], depending on the content of the data set.Note that all information loss measures are divided by the maximum between the original and the protected statistic, therefore all these measures are ranked in the $[0,1]$ interval.

5. Time series disclosure risk measures

Section 4 discusses ways to measure the information loss caused by protection methods in time series. However, as we have explained in the preliminaries, the quality assessment of a protection method should not be restricted to its information loss, but it should also include a measure of its disclosure risk. The best method is the one that optimizes the trade-off between both magnitudes. Consider the following extreme cases as examples of this trade-off:

- If masking consists of encrypting the original data, no disclosure is possible, but no information at all is released (maximum information loss, minimum disclosure risk).
- If no masking is performed and the original data is released, users can perform fully accurate computations, but disclosure of individual respondent data is complete (minimum information loss, maximum disclosure risk).

Once the modified data set X' is released, everybody can see its content. Let us assume, following the scenario described in Section 3, that $X' = X'_{nc}||X_c$. Let us assume now that an intruder has access to some other data set $Y = Y_{id}||Y_{nc}$ which includes the identifier and the non-confidential quasi-identifier attributes of some of the individuals whose data is in X'. If this intruder is able to correctly link a record $y_{nc} \in Y_{nc}$ (which is also in X'_{nc}) with the corresponding protected record $x'_{nc} \in X'_{nc}$, then he will know that the matching (not modified) confidential information x_c belongs to the individual with identifier y_{id}, breaking therefore the privacy of this individual.

According to this scenario, disclosure risk measures have to be in accordance with the difficulty for an intruder of linking the protected data with the original data. In this work, we propose the use of the record linkage methods presented in [32] that were specially designed for time series re-identification.

We also measure the disclosure risk in a scenario where the intruder has no access to an external data set. In this case, we assume that the intruder tries to infer the original values from the ones protected. We model this situation using the *interval disclosure*. In this approach, an interval is considered around each protected value. Then, when one original value falls in the interval defined around the corresponding protected value, we assume that the intruder obtains a value of enough quality to break the privacy of the data respondent.

5.1. Time series normalization

It is usual to normalize data sets before applying record linkage methods. This is so to avoid the scale problems of raw data. The following two alternatives are usually considered:

- **Ranging.** Raw data is translated into the $[0,1]$ interval using this expression $x' = \frac{(x-\min(a))}{(\max(a)-\min(a))}$, where x is the original value and $\max(a)$ and $\min(a)$ are the maximum and minimum values for the corresponding attribute a.
- **Standardization.** Raw data is normalized by translating the mean to be equal zero and the standard deviation to be equal one. That is, $x' = \frac{(x-\mu_a)}{S_a}$, where μ_a and S_a are, respectively, the mean and the standard deviation of the corresponding attribute a.

This kind of pre-processing, when applied separately for each component of the time series, causes the loss of the temporal information of the time series. For this reason, we apply another type of normalization using all the elements included into the time series. In this work we had used the following normalization

$$x'_i = \frac{(x_i - \mu_x)}{S_x},$$

where μ_x and S_x are the mean and the standard deviation of the elements of the corresponding time series.

Now, we illustrate with a clear example (that uses the index of prices for some food products) the impact of the normalization of the time series, comparing the normalization by component (each component treated as an attribute) and the normalization of the time series as a whole. The example reveals that the normalization by component distorts completely the shape of the time series.

Example 2. Let us consider the price index of four different foods in nine years. We can observe in Table 1 the original raw values and their tendency during the period 1993–2001 and in Tables 2 and 3, respectively, the normalized data values after standard (component-wise) and time series (data altogether) normalization.

Fig. 2 shows that different normalizations produce different outcomes and that the standard component-wise normalization causes important divergences on the tendency of the time series between the original time series and the normalized one. For example, in the case of bread, when comparing charts (a) with (b), we observe that in the original data bread price tendency was to increase every year but that after normalization bread price has a decreasing tendency. This is a negative effect of the normalization over the data.

To avoid this effect of component-wise normalization, we propose the use of specific normalization procedures for time series: normalization of all the series.

Table 1
Data extracted from Spanish National Statistics Institute.

	Index of prices								
	1993	1994	1995	1996	1997	1998	1999	2000	2001
Bread	106.5	110.3	114.9	117.9	119.3	121	122.2	124.1	129
Oil	102.7	119.8	147.8	178.7	130.8	116.2	133.6	123.5	114.4
Vegetables	95.6	101.9	110.8	116.4	114.2	119	124.6	126.4	133.9
Potatoes	101.1	133.6	162.8	123.8	121.3	140.4	149.8	148.6	177.6

Table 2
Data normalized with the standard component-wise procedure.

	Index of prices								
	1993	1994	1995	1996	1997	1998	1999	2000	2001
Bread	1.00	0.26	0.08	0.02	0.31	0.20	0.00	0.02	0.23
Oil	0.65	0.56	0.71	1.00	1.00	0.00	0.41	0.00	0.00
Vegetables	0.00	0.00	0.00	0.00	0.00	0.12	0.09	0.12	0.31
Potatoes	0.50	1.00	1.00	0.12	0.43	1.00	1.00	1.00	1.00

Table 3
Data normalized with the time series procedure.

	Index of prices								
	1993	1994	1995	1996	1997	1998	1999	2000	2001
Bread	−1.00	−0.81	−0.57	−0.42	−0.34	−0.26	−0.19	−0.10	0.16
Oil	−1.20	−0.32	1.13	2.72	0.25	−0.50	0.39	−0.13	−0.60
Vegetables	−1.57	−1.24	−0.78	−0.49	−0.61	−0.36	−0.07	0.02	0.41
Potatoes	−1.28	0.39	1.90	−0.11	−0.24	0.75	1.23	1.17	2.66

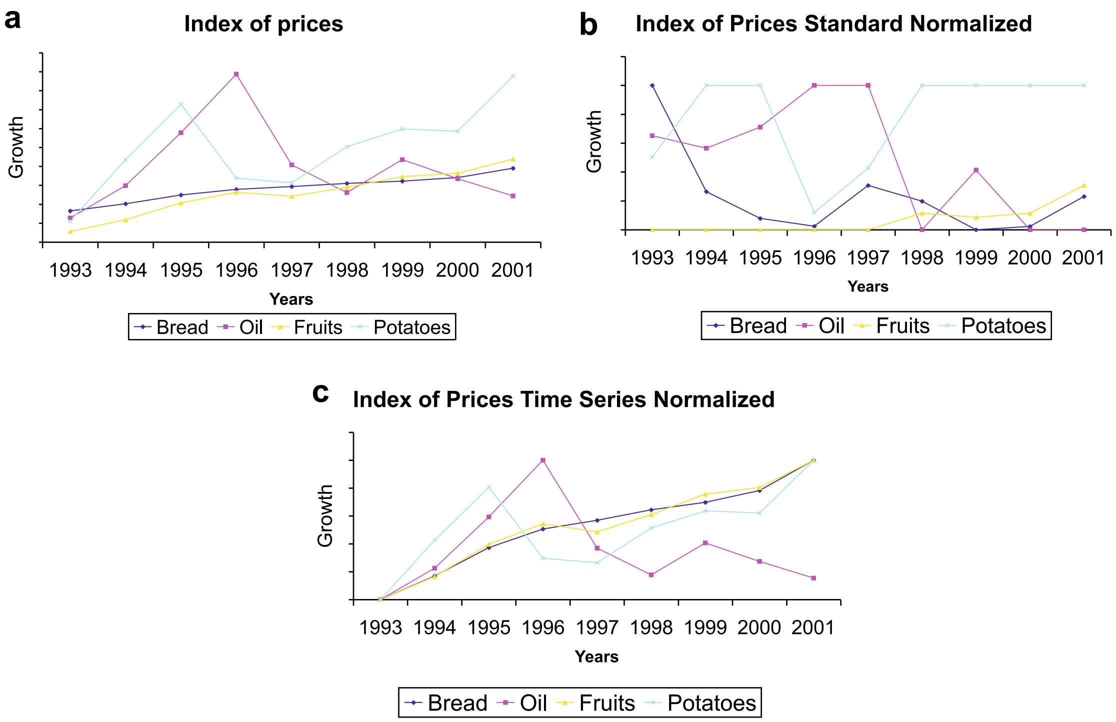
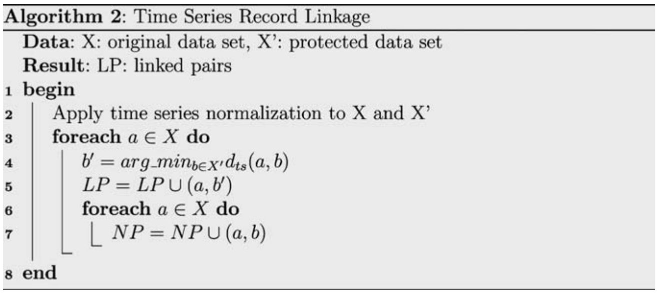


Fig. 2. Graphical representation of the effects of time series normalization, (a) represents the original data without normalization, (b) represents normalized data with independent normalization, (c) represents normalized data with time series normalization.

5.2. Time series re-identification

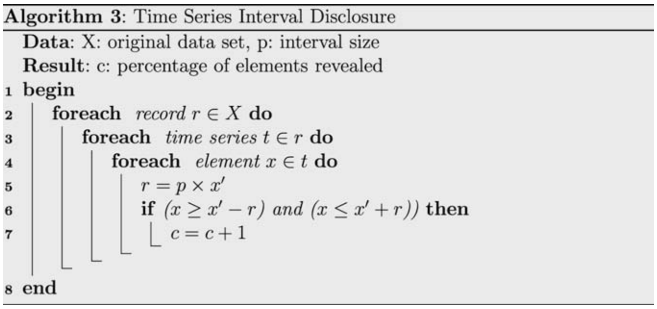
The time series record linkage presented in this section is based on the standard distance based record linkage (DB-RL). Recall that DB-RL method can be applied when a distance between pairs of records (one in the original data set and the other in the protected data set) can be defined. Then, every protected record is linked to the closest original one. When the data is numerical (DB-RL standard), it is usual to use the Euclidean distance (after normalizing the whole data set). In our case with time series, we use the normalization explained above and the distances presented in Section 2.1. That is, the Euclidean distance and the STS distance. Formally, time series record linkage is described in Algorithm 2, where $d_{ts}(a,b)$ is defined in terms of a given distance d_{x_i} for each time series x_i.



```
Algorithm 2: Time Series Record Linkage
Data: X: original data set, X': protected data set
Result: LP: linked pairs
begin
    Apply time series normalization to X and X'
    foreach $a \in X$ do
        $b' = arg\_min_{b \in X'} d_{ts}(a,b)$
        $LP = LP \cup (a,b')$
        foreach $a \in X$ do
            $NP = NP \cup (a,b)$
end
```

5.3. Time series interval disclosure

When the intruder has no access to any external data source, he can try to approximate original values assuming that they are in a finite interval around the protected value. To measure the risk of this approach, we apply the Algorithm 3 where p is a parameter defined by the user which defines the interval length.



```
Algorithm 3: Time Series Interval Disclosure
Data: X: original data set, p: interval size
Result: c: percentage of elements revealed
begin
    foreach record $r \in X$ do
        foreach time series $t \in r$ do
            foreach element $x \in t$ do
                $r = p \times x'$
                if $(x \geq x' - r)$ and $(x \leq x' + r))$ then
                    $c = c + 1$
end
```

Normally, the parameter p is defined by using a percentage of difference of an element. For example, with $p = 10\%$, if the element is equal to 10, the corresponding interval will be $[9,11]$. This measure is very similar to the measure presented in [4], where authors quantify disclosure risk measuring how closely the original values of a protected attribute can be estimated.

5.4. The computation of the DR measures

Considering the two scenarios presented above, it is possible to compute the final disclosure risk as:

$$DR = \frac{DR_1 + DR_2}{2},$$

where DR_1 and DR_2 summarize the re-identification risk and the interval disclosure risk, respectively.

DR_1 is computed by maximizing the percentage of records correctly linked by the intruder using different time series distances. Here, we want to evaluate the worst scenario, therefore we should consider that the intruder always uses the best record linkage method in each case. As in our case we consider *EULD* (Euclidean distance linkage disclosure) and *STSLD* (Short time series linkage disclosure), we compute DR_1 using the formula

$$DR_1 = \mathrm{Max}(EULD, STSLD)$$

where *EULD* and *STSLD* are the average percentage of records correctly linked using time series record linkage with Euclidean and STS distance when the intruder knows different numbers of time series (from 1 to all).

DR_2 is computed as the interval disclosure using different values for the parameter p, in our case p ranges from 1% to 10%

$$DR_2 = \frac{\sum_{p=0.01}^{0.1} ID_p}{10}.$$

6. Final trade-off evaluation

As we said in the introduction, information loss and disclosure risk are combined to obtain a global value about a specific protection method, this value weighs the relationship between information loss and disclosure risk. There are different approaches to achieve this goal, some of them use a graphical representation, for instance R-U maps [17], others are analytical. For example, in [10], the authors combine both information loss and disclosure risk in a *score* using the arithmetic mean.

As we are more interested in presenting a final result in an analytical way, we use this latter approach. Therefore, the final evaluation of a time series protection method uses the following *score*:

$$score = \frac{IL + DR}{2},$$

where IL is the overall information loss measure and DR is the overall disclosure risk measure. The rationale of this measure is to give equal weight to information loss and to disclosure risk. As they are contradictory goals, it is impossible to reduce both at the same time, then the arithmetic mean shows us which protection method and parameters are more suitable for a given data set. Note that the lower the score, the best the protection method.

7. Experiments

7.1. Data protection

To analyze empirically our framework and to evaluate the time series microaggregation method we have protected some real data sets that can be obtained freely from different data sources. Firstly, we have used a file from [24] (the so-called forecasters) with 3003 time series of different lengths (between 14 and 64 elements). We have re-sampled all time series to 10 elements to covert them into the same length. Secondly, we have used the Stock Exchange information of the thirty five most important Spanish companies. These companies are ranked in the so-called Ibex35 stock market. We have downloaded the information about prices from June, 21st 2005 to April, 28th 2006 from [42]. And finally, we have used data information about all football teams of the nine most important European domestic leagues from [23]. As said above, the information about these three testbeds is publicly available. Data details are given in Table 4.

We have considered all time series as quasi-identifiers. Therefore, we need to apply a protection method (microaggregation) to all of them. For the disclosure risk computation we assume that the intruder has access to all time series. The lack of confidential attributes does not affect our calculation since confidential attributes are not protected, information loss is zero for confidential attributes and they are disregarded in the disclosure risk computation.

We have protected the original data with the time series microaggregation method described in Section 3. We have applied this method with $k \in \{2, 3, 6, 9, 12\}$.

To measure the disclosure risk when the intruder has access to more than one time series, we have split original time series into n smaller time series. After that, we have applied the time series microaggregation method to these new data sets. In this way, we study the scenario where time series microaggregation does not preserve k-anonymity as in the case of standard multivariate microaggregation. We detail now this transformation for each file.

- **Forecasters problem.** We have split the original time series into $n \in \{1, 2\}$ time series. So, in this case we have two different data sets, one with one time series and another one with two time series.
- **Ibex35 problem.** We have split the original time series into $n \in \{2, 4, 20\}$ time series. So, in this case, we have three different data sets with 4, 8 and 40 time series.
- **Football problem.** In this case, no conversion is done because the original data set already consisted of eight time series.

7.2. Results

We have computed the score and its components for the three data sets presented before. It is possible to divide the overhead produced by such computations into two different parts. On the one hand, one should consider the overhead produced by the calculation of the information loss measures and, on the other hand, the cost of the computation related to the disclosure risk measures. The main overhead related to information loss measures is produced by the computation of IL_3 where three forecasted values have to be computed for each time series. This overhead could be very large if we select a very complex forecasting model or we apply such measure to a very large time series. However, this is not our case. We have selected five forecasting models with a low computational cost. In the largest example, Ibex35, time series have 220 points and

Table 4
Details of time series examples.

Forecasters	Records	3003
	Number of time series	1
	Time series length	10
	Number of elements	10
	Series description	Financial information
Ibex35	Records	35
	Number of time series	2
	Time series length	220
	Number of elements	440
	Series description	Financial information, volume transactions
Football	Records	176
	Number of time series	8
	Time series length	25
	Number of elements	200
	Series description	Years, FIFA points, Leage position, goals for goals against, matches won, matches drew, matches lost

information loss measures are computed in less than 20 s in a desktop computer. On the contrary, the main overhead of disclosure risk measures is related to the number of records (n) considered in the record linkage process. It is well-know that record linkage has a complexity equal to $O(n^2)$, however, this cost can be reduced to $O(bn)$ by using a blocking technique [30], where b is the size of the block used. Therefore, we believe that the overhead produced by the computation of the score is affordable for large data sets in a desktop computer.

Tables 5–7 present the score and its components for the forecaster, football and ibex35 data set, respectively. Columns one to three present the IL_i components and column four shows the overall IL value. From these columns we can infer that IL increases when k increases. E.g. in the forecaster problem protected with *eu*-microaggregation, *IL* values range from 6.78 to 15.89. This behavior is consistent with the usual results for general microaggregation methods.

We can also infer from IL_1 (column one) that time series microaggregation preserve the time series mean and autocorrelation function. See, for example, the forecasters data set in Table 5, where for all the microaggregation configurations, IL_1 is always 0.00. It is known that general microaggregation preserves the average when applied to numerical attributes. Therefore, it is not surprising that time series microaggregation also preserves time series mean, when applied to time series.

Comparing *eu*-microaggregation and *sts*-microaggregation with the same k and number of series, it can be observed that (in general) *IL* is lower for the *eu*-microaggregation. However, in a few cases, *sts*-microaggregation obtains a lower *IL*. For instance, in the forecasters data set with two time series and $k = 2$, *IL* for *eu*-microaggregation is equal to 18.57 while for *sts*-microaggregation is equal to 17.11.

Columns five and six present the *EULD* and *STSLD*. From these two columns it is clear that re-identification risk decreases when k increases. The same happens with *ID* and the overall *DR* (columns seven and eight). Then, we can say that parameter k is inversely proportional to disclosure risk.

Table 5
Score and its components in the forecasters data set. Forecasters.*i*.*d*-*k* corresponds to microaggregation using distance *d* (Euclidean or STS) with *i* series and parameter *k*.

	i	*k*	IL_1	IL_2	IL_3	*IL*	*EULD*	*STSLD*	*ID*	*DR*	*Score*
Forecasters.*i*.eu-*k*	1	2	0.00	6.33	6.78	4.37	42.79	33.37	40.32	41.56	22.96
	1	3	0.00	8.32	9.27	5.86	25.67	17.32	39.20	32.44	19.15
	1	6	0.00	11.00	12.56	7.85	10.62	6.03	37.63	24.13	15.99
	1	9	0.00	12.57	14.20	8.92	7.19	3.33	36.52	21.86	15.39
	1	12	0.00	13.63	15.89	9.84	5.49	2.56	35.80	20.65	15.24
	2	2	0.00	26.27	29.44	18.57	28.37	22.56	28.41	28.39	23.48
	2	3	0.00	27.76	31.84	19.87	15.55	11.72	26.43	20.99	20.43
	2	6	0.00	26.12	32.30	19.47	7.74	5.11	26.42	17.08	18.28
	2	9	0.00	25.09	31.74	18.95	6.44	3.50	26.97	16.71	17.83
	2	12	0.00	24.41	30.34	18.25	5.44	2.56	27.76	16.60	17.43
Forecasters.*i*.sts-*k*	1	2	0.00	10.16	7.88	6.01	30.67	42.12	38. 14	40.13	23.07
	1	3	0.00	12.72	10.10	7.61	17.65	25.61	36.60	31.11	19.36
	1	6	0.00	16.53	13.39	9.97	7.16	10.92	34.05	22.49	16.23
	1	9	0.00	18.51	15.46	11.32	4.83	7.49	32.70	20.10	15.71
	1	12	0.00	20.41	17.10	12.50	3.26	5.19	31.39	18.29	15.40
	2	2	0.00	23.69	27.63	17.11	20.63	30.82	29.63	30.23	23.67
	2	3	0.00	26.01	29.40	18.47	11.39	17.62	27.80	22.71	20.59
	2	6	0.00	28.68	31.31	19.99	5.81	8.04	26.05	17.05	18.52
	2	9	0.00	30.30	32.78	21.03	3.70	5.61	24.93	15.27	18.15
	2	12	0.00	31.69	34.17	21.95	2.86	4.56	24.19	14.38	18.16

Table 6
Score and its components in the football data set. football.*i*.*d*.*k* corresponds to microaggregation using distance *d* (Euclidean or STS) with *i* series and parameter *k*.

	i	*k*	IL_1	IL_2	IL_3	*IL*	*EULD*	*STSLD*	*ID*	*DR*	*Score*
Football.*i*.eu-*k*	8	2	0.11	44.55	44.25	29.63	84.16	84.37	19.94	52. 16	40.89
	8	3	0.15	45.38	45.51	30.35	78.91	75.14	19.73	49.32	39.84
	8	6	0.17	45.39	45.75	30.44	65.27	54.26	19.49	42.38	36.41
	8	9	0.4	45.44	46.37	30.74	54.97	34.38	19.61	37.29	34.02
	8	12	0.22	45.43	45.20	30.28	50.28	27.91	19.44	34.86	32.57
Football.*i*.sts-*k*	8	2	0.1	46.66	45.41	30.72	71.66	83.59	19. 74	51.67	41.19
	8	3	0.16	49.08	48.43	32.56	56.82	77.77	17.68	47.73	40.14
	8	6	0.25	50.84	50.13	33.74	31.75	58.17	15.79	36.98	35.36
	8	9	0.31	51.9	48.94	33.72	22.23	41.12	14.82	27.97	30.85
	8	12	0.34	52.63	49.79	34.25	14.35	33.95	14.74	24.35	29.30

Table 7
Score and its components in the ibex35 data set. ibex35.*i*.*d*.*k* corresponds to microaggregation using distance *d* (Euclidean or STS) with *i* series and parameter *k*.

	i	*k*	IL_1	IL_2	IL_3	*IL*	*EULD*	*STSLD*	*ID*	*DR*	*Score*
Ibex35.*i*.eu-*k*	2	2	0	29.82	31.51	20.44	57.14	52.86	13.54	35.34	27. 89
	2	3	0.00	36.82	40.05	25.62	35.71	35.71	10.57	23.14	24.38
	2	6	0.01	43.61	44.18	29.27	14.29	18.57	8.69	13.63	21.45
	2	9	0.02	44.4	52.81	32.41	5.71	10.00	7.47	8.74	20.57
	2	12	0.03	49.34	52.54	33.97	7.14	4.29	6.74	6.94	20.46
	4	2	0.00	31.85	32.12	21.32	60.00	60.71	12.50	36.61	28.96
	4	3	0.00	37.44	37.28	24.91	45.71	42.14	10.06	27.89	26.40
	4	6	0.01	42.61	44.96	29.19	21.43	22.86	8.83	15.85	22.52
	4	9	0.01	44.29	47.63	30.64	11.43	14.29	7.14	10.72	20.68
	4	12	0.01	49.9	53.63	34.51	8.57	7.14	6.78	7.68	21.09
	20	2	0.00	32.78	35.27	22.68	67.86	70.57	11.89	41.23	31.96
	20	3	0.00	37.71	41.09	26.27	48.71	48.14	9.76	29.24	27.75
	20	6	0.00	42.73	45.59	29.44	26.43	24.71	8.72	17.58	23.51
	20	9	0.00	44.97	49.81	31.59	12.29	16.71	6.65	11.68	21.64
	20	12	0.00	50.63	52.83	34.49	8.29	7.71	7.09	7.69	21.09
Ibex35.*i*.sts-*k*	2	2	0.02	45.94	47.57	31.18	20.00	51.43	6.81	29. 12	30.15
	2	3	0.04	47.87	49.53	32.48	15.71	28.57	6.13	17.35	24.92
	2	6	0.07	58.98	62.03	40.36	10.00	11.43	4.83	8.13	24.25
	2	9	0.18	57.09	64.54	40.60	1.43	2.86	3.56	3.21	21.91
	2	12	0.17	58.04	61.52	39.91	4.29	2.86	5.77	5.03	22.47
	4	2	0.02	45.61	45.72	30.45	20.71	52.86	7.94	30.40	30.43
	4	3	0.03	51.69	50.39	34.04	15	34.29	4.94	19.62	26.83
	4	6	0.07	56.5	57.91	38.16	10	16.43	5.08	10.76	24.46
	4	9	0.08	57.44	57.11	38.21	7.86	7.86	4.75	6.31	22.26
	4	12	0.13	57.88	57.72	38.58	5.71	4.29	5.16	5.44	22.01
	20	2	0.00	46.47	47.28	31.25	19.57	60.29	8.24	34.27	32.76
	20	3	0.01	51.55	53.66	35.08	14.43	43.43	5.55	24.49	29.79
	20	6	0.02	57.51	59.55	39.03	11.43	22.43	4.19	13.31	26.17
	20	9	0.03	57.06	60.42	39.17	7.86	13.86	4.7	9.28	24.23
	20	12	0.02	57.25	60.06	39.11	6	12	5.46	8.73	23.92

In general, the greatest re-identification risk for a given microaggregation (*eu* and *sts*) occurs when the same distance is used in the time series record linkage. For instance, in the football data set configurations with $k = 6$, *EULD* for *eu*-microaggregation is 65.54 while *STSLD* is 54.26. In contrast, using *sts*-microaggregation the largest re-identification risk is *STSLD* (58.17).

If one compares the *ID* of both microaggregation methods, in general, *sts*-microaggregation achieves lower values. For instance, comparing in the football data set both microaggregation methods with $k = 12$, *eu*-microaggregation obtains 19.44 while *sts*-microaggregation only 14.74.

The last column of each table shows the overall score. It can be observed that the score is very data set dependent. However, (in general) with small values of *k* the best scores are obtained by *eu*-microaggregation (e.g. forecasters.1.eu.2 is equal to 22.96 and forecasters.1.sts.2 is equal to 23.48). On the other hand, with large values of *k* the best scores are obtained by *sts*-microaggregation (e.g. football.8.eu.12 is equal to 32.57 and football.8.sts.12 is equal to 29.30). This fact could be a clue for a data holder. For instance, if he is interested in protecting his data set with a very low disclosure risk, i.e. a large *k*, he should choose to protect his data set with *sts*-microaggregation because its trade-off between information loss and disclosure risk is better than *ue*-microaggregation when high levels of privacy are required. On the contrary, if the data holder is interested in releasing a protected data set with a high statistical utility he should use *eu*-microaggregation instead of *sts*-microaggregation.

To sum up, the score presented here is a good tool to evaluate different aspects of a given protection method, including the evaluation of the privacy provided by a protection method in different scenarios or the statistical utility of released data for a wide range of mining analysis. Moreover, as we have explained before, our score can help a data holder to choose a certain protection method depending on his preferences.

8. Conclusions and future work

In this paper, we have presented a new framework for evaluating time series protection methods. We have introduced a set of information loss and disclosure risk measures for time series. Such measures cover all common uses of time series. We have presented some results analysing an extension of microaggregation for time series.

As future work, we plan to study new information loss and disclosure risk measures for categorical and ordinal time series protection methods. Therefore, further extending this framework to other types of data.

Acknowledgements

Partial support by the Spanish MEC (projects ARES – CONSOLIDER INGENIO 2010 CSD2007-00004 – and eAEGIS – TSI2007-65406-C03-02)) and Generalitat de Catalunya (Grant 2005-SGR-00093) is acknowledged. Jordi Nin wants to thank the Spanish National Research Council (CSIC) for his I3P grant.

References

[1] N.R. Adam, J.C. Wortmann, Security-control for statistical databases: a comparative study, ACM Computing Surveys 21 (1989) 515–556.

[2] C. Aggarwal, On *k*-anonymity and the curse of dimensionality, in: Proceedings of the 31st International Conference on Very Large Databases, 2005, pp. 901–909.

[3] G. Aggarwal, T. Feder, K. Kenthapadi, S. Khuller, R. Panigrahy, D. Thomas, A. Zhu, Achieving anonymity via clustering, in: Proceedings of the 25th ACM Symposium on Principles of Databases Systems, 2006, pp. 153–162.

[4] R. Agrawal, S. Ramakrishnan, Privacy-preserving data mining, SIGMOD Records 29 (2) (2000) 439–450.

[5] J.S. Armstrong, Principles of Forecasting: A Handbook for Researchers and Practitioners, Kluwer Academic Publishers, Norwell, Massachusetts, 2001, ISBN 0-7923-7930-6.

[6] E. Bertino, I. Nai Fovino, L. Parasiliti Provenza, A framework for evaluating privacy preserving data mining algorithms, Data Mining and Knowledge Discovery, vol. 11, (No. 2), Springer, 2005.

[7] G. Box, G. Jenkins, Time series analysis, Forecasting and Control, Holden-Day, Incorporated, 1990.

[8] P. Brockwell, R. Davis, Introduction to Time Series and Forecasting, Springer Texts in Statistics, 2002, ISBN 978-0-387-95351-9.

[9] S. Chu, E. Keogh, D. Hart, M. Pazzani, Iterative Deepening dynamic time warping for time series, in: The Second SIAM International Conference on Data Mining Chicago, USA, April 11–13, 2002.

[10] J. Domingo-Ferrer, V. Torra, Disclosure control methods and information loss for microdata, 2001, pp. 91–110 of [15].

[11] J. Domingo-Ferrer, V. Torra, A quantitative comparison of disclosure control methods for microdata, 2001, pp. 111–133 of [15].

[12] J. Domingo-Ferrer, J.M. Mateo-Sanz, Practical data-oriented microaggregation for statistical disclosure control, IEEE Transactions on Knowledge and Data Engineering 14 (2002) 189–201.

[13] J. Domingo-Ferrer, V. Torra, Ordinal, continuous and heterogeneous *k*-anonymity through microaggregation, Data Mining and Knowledge Discovery 11 (2005) 195–212.

[14] J. Domingo-Ferrer, F. Sebé, A. Solanas, Microaggregation Heuristics for P-sensitive K-anonymity, in: Proceedings of Joint UNECE/Eurostat Work Session on Statistical Data Confidentiality, 2007.

[15] P. Doyle, J. Lane, J. Theeuwes, L. Zayatz (Eds.), Confidentiality, Disclosure, and Data Access: Theory and Practical Applications for Statistical Agencies, Elsevier Science, 2001.

[16] G. Duncan, S. Fienberg, R. Krishnam, R. Padman, S. Roehrig, Disclosure limitation methods and information loss for tabular data, 2001, pp. 135–166 of [15].

[17] G. Duncan, S. Keller-McNulty, S. Stokes, Disclosure risk vs. data utility: the R-U confidentiality map, Technical Report Number 121 of National Institute of Statistical Sciences, 2001. <www.niss.org>.

[18] G. Duncan, S. Keller-McNulty, S. Stokes, Database security and confidentiality: examining disclosure risk vs. data utility through the R-U confidentiality map, Technical Report Number 142 of National Institute of Statistical Sciences, 2004. <www.niss.org>.

[19] F. Felso, J. Theeuwes, G.G. Wagner, Disclosure limitation methods in use: results of a survey, 2001, pp. 17–42 of [15].

[20] B. Fung, K. Wang, P. Yu, Top-down specialization for information and privacy preservation, in: Proceedings of the 21st IEEE International Conference on Data Engineering, 2005, pp. 205–216.

[21] C. Gomes da Silva, Time series forecasting with a non-linear model and the scatter search meta-heuristic, Information Sciences 178 (16) (2008) 3288–3299.

[22] C. Hamzaçebi, Improving artificial neural networks performance in seasonal time series forecasting, Information Sciences 178 (23) (2008) 4550–4559.

[23] Fotball statistics web. <http://www.histora.org/>.

[24] International institute of forecasters. <http://www.forecasters.org/>.

[25] J. Lane, P. Heus, T. Mulcahy, Data access in a cyber world: making use of cyber infrastructure, Transactions on Data Privacy 1 (1) (2008) 2–16.

[26] A. Machanavajjhala, J. Gehrke, D. Kifer, M. Venkitasubramaniam, l-Diversity: privacy beyond *k*-anonymity, in: 22nd IEEE International Conference on Data Engineering, 2006.

[27] J.M. Mateo-Sanz, J. Domingo-Ferrer, A method for data-oriented multivariate microaggregation, Statistical Data Protection for Official Publications of the European Communities (1999) 89–99.

[28] J.M. Mateo-Sanz, J. Domingo-Ferrer, F. Sebé, Probabilistic information loss measures in confidentiality protection of continuous microdata, Data Mining and Knowledge Discovery, vol. 11, (No. 2), Springer, 2005.

[29] C.S. Möller-Levet, F. Klawonn, K.-H. Cho, O. Wolkenhauer, Fuzzy clustering of short time series and unevenly distributed sampling points, in: Proceedings of the Fifth International Symposium on Intelligent Data Analysis, Berlin, Germany, August 28–30, 2003.

[30] H. Newcombe, Record linking: the design of efficient systems for linking records into individuals and family histories, American Journal of Human Genetics (1967).

[31] J. Nin, V. Torra, Extending microaggregation procedures for time series protection, Lecture Notes in Artificial Intelligence 4259 (2006) 899–908.

[32] J. Nin, V. Torra, Distance based re-identification for time series, analysis of distances, Lecture Notes in Artificial Intelligence 4302 (2006) 205–216.

[33] J. Nin, J. Herranz, V. Torra, Rethinking rank swapping to decrease disclosure risk, Data and Knowledge Engineering 64 (1) (2008) 346–364.

[34] J. Nin, J. Herranz, V. Torra, How to group attributes in multivariate microaggregation, International Journal on Uncertainty, Fuzziness and Knowledge-Based Systems 16 (1) (2008) 121–138.

[35] A. Oganian, J. Domingo-Ferrer, On the complexity of optimal microaggregation for statistical disclosure control, Statistical Journal of United Nations Economic Commission for Europe 18 (4) (2000) 345–354.

[36] N. Rescher, Predicting the Future: An Introduction to the Theory of Forecasting, State University of New York Press, 1998, ISBN 0791435539.

[37] P. Samarati, L. Sweeney, Protecting privacy when disclosing information: *k*-anonymity and its enforcement through generalization and suppression, SRI Intl. Tech. Rep, 1998.

[38] G. Sande, Exact and approximate methods for data directed microaggregation in one or more dimensions, International Journal of Uncertainity Fuzziness and Knowledge-Based Systems 10 (5) (2002) 459–476.

[39] Sebé, J. Domingo-Ferrer, J.M. Mateo-Sanz, V. Torra, Post-masking optimization of the tradeoff between information loss and disclosure risk in masked microdata sets, Inference Control in Statistical Databases, Lecture Notes in Computer Science 2316 (2002) 187–196.

[40] D. Shah, S. Zhong, Two methods for privacy preserving data mining with malicious participants, Information Sciences 177 (23) (2008) 5468–5483.

[41] D. Smith, M. Elliot, A measure of disclosure risk for tables of counts, Transactions on Data Privacy 1 (2008) 34–52.

[42] Stock Exchange web, Sabadell Bank. <http://www.bsmarkets.com/>.

[43] L. Sweeney, Achieving *k*-anonymity privacy protection using generalization and suppression, International Journal of Uncertainity Fuzziness and Knowledge-Based Systems 10 (5) (2002) 571–588.

[44] L. Sweeney, *k*-anonymity: a model for protecting privacy, International Journal of Uncertainity Fuzziness and Knowledge-Based Systems 10 (5) (2002) 557–570.

[45] V. Torra, J. Domingo-Ferrer, Record Linkage Methods for Multidatabase Data Mining, Information Fusion in Data Mining, Springer, 2003.

[46] V. Torra, Microaggregation for categorical variables: a median based approach, Proc, Privacy in Statistical Databases (PSD 2004). Lecture Notes in Computer Science 3050 (2004) 162–174.

[47] V. Torra, J. Abowd, M. Domingo-Ferrer, Using Mahalanobis distance-based record linkage for disclosure risk assessment, Privacy in Statistical Databases 2006, Lecture Notes in Computer Science, vol. 4302, Springer-Verlag, 2006. pp. 233–242.

[48] T. Truta and B. Vinay (2006) Privacy protection: p-sensitive k-anonymity property. In 2nd International Workshop on Privacy Data Management PDM 2006, page 94, Berlin Heidelberg. IEEE Computer Society.

[49] T. Warren Liao, Clustering of time series data – a survey, Pattern Recognition 38 (2005) 1857–1874.

[50] W.E. Winkler, Data cleaning methods, in: Proceedings of the SIGKDD 2003, Washington, 2003.

[51] W.E. Winkler, Re-identification methods for masked microdata, Privacy in Statistical Databases 2004, Lecture Notes in Computer Science, vol. 3050, Springer-Verlag, 2004. pp. 216–230.

[52] W. Yancey, W. Winkler, R. Creecy, Disclosure risk assessment in perturbative microdata protection, Inference Control in Statistical Databases: From Theory to Practice, Lecture Notes in Computer Science, vol. 2316, Springer-Verlag, 2002. pp. 135-152.

[53] S. Zhong, Privacy-preserving algorithms for distributed mining of frequent itemsets, Information Sciences 177 (2) (2007) 490–503.