Journal of King Saud University – Computer and Information Sciences (2015) 27, 297–304

King Saud University

Journal of King Saud University – Computer and Information Sciences

www.ksu.edu.sa
www.sciencedirect.com

An anonymization technique using intersected decision trees

Sam Fletcher *, Md Zahidul Islam

Center for Research in Complex Systems (CRiCS), School of Computing and Mathematics, Charles Sturt University, Bathurst, NSW 2795, Australia

Received 22 February 2014; revised 25 April 2014; accepted 4 June 2014
Available online 19 June 2015

KEYWORDS

Privacy preserving data mining;
Decision tree;
Anonymization;
Data mining;
Data quality

Abstract Data mining plays an important role in analyzing the massive amount of data collected in today's world. However, due to the public's rising awareness of privacy and lack of trust in organizations, suitable Privacy Preserving Data Mining (PPDM) techniques have become vital. A PPDM technique provides individual privacy while allowing useful data mining. We present a novel noise addition technique called Forest Framework, two novel data quality evaluation techniques called EDUDS and EDUSC, and a security evaluation technique called SERS. Forest Framework builds a decision forest from a dataset and preserves all the patterns (logic rules) of the forest while adding noise to the dataset. We compare Forest Framework to its predecessor, Framework, and another established technique, GADP. Our comparison is done using our three evaluation criteria, as well as Prediction Accuracy. Our experimental results demonstrate the success of our proposed extensions to Framework and the usefulness of our evaluation criteria.

1. Introduction

As technology has advanced, so has the ability to gather massive amounts of data. Data Mining plays an important role in data collection, pre-processing, integration and pattern extraction from the collected data. Due to the wide use of data mining, it is important to consider the ramifications. The most prominent ramification is perhaps the breach of individual privacy. This public awareness of privacy and lack of trust in organizations (Arnott, 2011) may introduce additional complexity to data collection. As a result, organizations may not be able to fully enjoy the benefits of data mining. Privacy Preserving Data Mining (PPDM) techniques have therefore become vital. A PPDM technique provides individual privacy while allowing useful data mining on a dataset. Typical PPDM techniques include noise addition to a dataset, data swapping, aggregation and masking (Brankovic et al., 2007; Dankar and Eman, 2012; Adam and Worthmann, 1989; Farkas and Jajodia, 2002). The two main aims of the PPDM techniques are high security and high data quality/utility.

In this paper, we propose a novel technique called Forest Framework as a modification of an existing technique called Framework (Islam and Brankovic, 2011). Forest Framework

* Corresponding author.
E-mail addresses: safletcher@csu.edu.au (S. Fletcher), zislam@csu.edu.au (M.Z. Islam).
Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.jksuci.2014.06.015

adds noise to a dataset in such a way that many existing patterns of the unperturbed dataset are preserved many more than is possible with Framework. We also propose two novel data quality evaluation techniques called "Evaluation of Data Utility using Domain Similarity" (EDUDS) and "Evaluation of Data Utility using Splitting Criteria" (EDUSC). Additionally, we propose a security analysis technique called "Security Evaluation using Record Similarity" (SERS). We carry out experiments on five natural datasets available from UCI Machine Learning Repository (Bache and Lichman, 2013). Our experimental results indicate the effectiveness of all our techniques. The organization of the paper is as follows: Section 2 provides a relevant background study; Section 3 presents Forest Framework; Section 4 presents EDUDS; Section 5 presents EDUSC; Section 6 presents SERS; and experimental results are presented in Section 7. Section 8 gives concluding remarks and avenues for future research.

We consider a dataset as a two dimensional table where rows represent the records and columns represent attributes. Each attribute can be numerical or categorical. Out of the attributes, one is a class attribute. Fig. 1 shows an example decision forest having two trees T_1 and T_2 obtained from a dataset. Decision tree algorithms iteratively discover which attribute best explains the class attribute for the given segment of records defined by the preceding attribute splits (e.g. $A > 7$ in Fig. 1) (Quinlan, 1993; Quinlan, 1996). In this example, each of the trees has three leaves L_1, L_2, and L_3. The leaves of a tree divide the dataset into mutually exclusive horizontal segments of records.

2. Background study

There are many privacy preserving data mining techniques in the literature, ranging from output privacy (Wang and Liu, 2011) to categorical noise addition (Giggins, 2012) to differential privacy (Friedman and Schuster, 2010), to many others discussed in surveys (Brankovic et al., 2007; Adam and Worthmann, 1989; Farkas and Jajodia, 2002; Wu et al., 2010). Framework – one such privacy preservation technique – was proposed in 2011 (Islam and Brankovic, 2011). It first builds a decision tree from an original dataset in order to learn the existing patterns (logic rules) of the dataset. It then adds noise to all attributes (both numerical and categorical) of a dataset in a way that preserves patterns discovered by the decision tree built from the original dataset. The basic idea of Framework is to add noise to the value of a numerical attribute of a record, but in such a way that the perturbed value falls within the range that satisfies the logic rule of the leaf where the record originally belongs to. That is, if a record in an unperturbed dataset falls in a leaf of the tree obtained from the original dataset, then Framework adds noise in such a way that the record still falls in the same leaf of the tree even after noise addition.

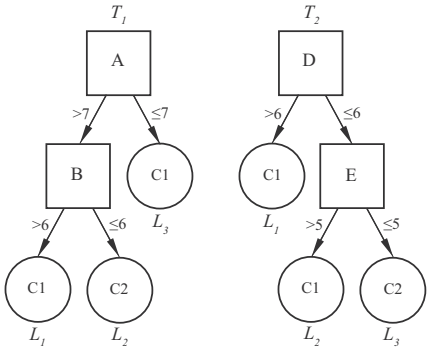

Fig. 1 An example decision forest with two trees.

For a categorical attribute, it first discovers which values are similar. Using a user-defined probability it then changes each value to another value having high similarity with the original value. For a class attribute, it shuffles the class values of the records belonging to the same leaf in such a way so that the distribution of class values among the records remains the same.

Two main aims of noise addition techniques are to perturb a dataset in order to preserve individual privacy and maintain high utility in the perturbed dataset. Measuring data utility is a challenging task (Ntoutsi et al., 2008; Osei-Bryson, 2004). Generally, the quality of a perturbed dataset is measured through the Prediction Accuracy of a decision tree, built from the perturbed dataset, while it is applied on an unperturbed testing dataset (Islam and Brankovic, 2011; Ray et al., 2011). It has also been shown that an assessment of data quality provided by a comparison of Prediction Accuracy may differ from an assessment provided by a comparison of decision tree similarity (Islam, 2007; Lim et al., 2000). The utility of a perturbed dataset is sometimes evaluated through the similarity of decision trees, the accuracy of the trees, and statistical properties such as mean and correlation matrix (Islam and Brankovic, 2011). Finding a suitable technique to compare the similarity of two trees can be a challenge.

General Additive Data Perturbation (GADP) perturbs only those attributes which are deemed confidential by a user, thus allowing data quality to remain as high as possible for the non-confidential attributes (Muralidhar et al., 1999). However, it takes all attributes into account when perturbing confidential attributes, and thus maintains all correlations among the attributes of a dataset. Modifications of GADP such as CGADP and EGADP have been proposed in order to preserve statistical parameters in datasets having non-multivariate normal distribution or small number of records (Sarathy et al., 2002; Muralidhar and Sarathy, 2005).

We will also be using the same benchmark perturbation technique used in the original paper: Random Technique (RT) (Islam and Brankovic, 2011). RT is a random noise addition technique which is used as a means for evaluating the effectiveness of other perturbation and evaluation techniques. It adds uniform noise to all attributes indiscriminately.

3. Our perturbation technique: forest framework

Forest Framework is a modification of an existing technique called Framework (Islam and Brankovic, 2011), with an aim to better preserve the original data quality in a perturbed dataset. Unlike Framework, it first builds a decision forest (Islam and Giggins, 2011) from an unperturbed dataset in order to learn the existing patterns (logic rules) of the dataset. The construction of a forest allows for far more patterns to be discovered, and therefore preserved. In our experiments we will be using the SysFor forest-building algorithm (Islam and Giggins, 2011), which harnesses the popular C4.5 tree building algorithm (Quinlan, 1993; Quinlan, 1996). Forest Framework

then adds noise to the attribute values of a record in such a way that the record still falls in the same leaf for each tree, even after the noise addition.

We first present its steps and then give a more detailed description of the steps.

Step 1: Build a decision forest.
Step 2: Intersect the leaves of each tree with the leaves of every other tree.

For each record, **DO:**
Step 3: Find the intersection that it falls in.
Step 4: Add noise to all numerical attributes.
Step 5: Add noise to all categorical attributes (including the class).
End **DO**.

Step 1: Build a decision forest. Many datasets have natural class attributes. For example, a patient dataset may have a class attribute called *Diagnosis*. If a dataset does not have any natural class attribute, a user can consider any suitable categorical attribute as the class attribute. In the absence of a categorical attribute a user can discretize a numerical attribute into a suitable number of categories. We then apply an existing decision forest algorithm such as SysFor or Credal (Islam and Giggins, 2011; Abellán and Masegosa, 2010) in order to build a decision forest F having a user defined number of decision trees, $F = \{T_1, T_2, \ldots T_f\}$.

Step 2: Intersect the leaves of each tree with the leaves of every other tree. Forest Framework then detects intersections of all leaves of the trees. As mentioned for our Fig. 1 example, the leaves of a tree divide the dataset into mutually exclusive horizontal segments. However, the leaves of two trees represent overlapping segments. Therefore, we can produce an intersection of the leaves of the trees. For example in Fig. 1, the intersection of L_1 of T_1 and L_1 of T_2 is the segment where all records have $A > 7, B > 6$ and $D > 6$.

If the logic rule for a leaf in Tree T_1 is: $A > 5 \wedge B > 3 \wedge E > 9 \rightarrow C_1$ and the logic rule for a leaf in T_2 is: $A \leqslant 8 \wedge B > 7 \rightarrow C_1$ then the intersection of the leaves (logic rules) is: $A[R : 6, 8] \wedge B > 7 \wedge E > 9$; meaning that the value of attribute A is within a range 6 and 8 inclusively, B is greater than 7, and E is greater than 9. We call the range of values of an attribute within a segment/intersection as a "sub-domain" of the attribute. Note that we use only non-class attributes in building the intersections.

Step 3: For each record, find the intersection that it falls in. Each record of a dataset is then assigned to the intersection that it falls in – i.e. the intersection for which the conditional values match the attribute values of the record. A record can only fall in one and only one intersection.

Step 4: For each record, add noise to all numerical attributes. Forest Framework then adds noise to each numerical attribute. Let v be the original value of attribute A, v' be the perturbed value of the attribute, and ϵ be the noise value then $v' = v + \epsilon$. The noise value ϵ is generated from a Gaussian distribution having mean μ and variance σ^2. We also apply a wrap-around approach (Islam and Brankovic, 2011) to maintain the range (sub-domain) of an attribute within an intersection. For example, we maintain the range/domain of attribute A within the intersection introduced in Step 2 as $A[R : 6, 8]$.

Step 5: For each record, add noise to all categorical attributes (including the class attribute). We swap the values for each categorical attribute among the records belonging to an intersection. Let the domain of the categorical values for attribute X be $\{X_1, X_2, \ldots, X_s\}$, where s is the sub-domain size of the attribute within an intersection. Let N_i be the number of records having value X_i. We first change the X values of all records within the intersection to null, and then randomly choose N_i number of records and assign the value $X_i; \forall i$. Therefore, the number of records N_i having a value $X_i; \forall i$ remains the same both before and after noise addition. However, with some probability, each categorical value for each individual record is altered.

The time complexity of Forest Framework is as follows. Step 1 has the same time complexity as the decision forest algorithm chosen by the user. Step 2 depends on the total number of leaves from all trees, which depends on the size of the trees, which in turn is largely dependent on the chosen decision forest algorithm. In ordinary circumstances it will have an insignificant time complexity. Step 3 is $O(n)$, where n is the number of records. Step 4 and 5 combined is $O(nm)$, where m is the number of attributes. Thus the overall time complexity of Forest Framework excluding Step 1 is $O(nm)$, which reduces to $O(n)$ assuming $n \gg m$.

4. Evaluation of Data Utility using Domain Similarity (EDUDS)

We now introduce our first data quality evaluation technique: "Evaluation of Data Utility using Domain Similarity" (EDUDS). The technique measures the similarity of an original and a perturbed dataset by evaluating the similarity of the forests built from the original and the perturbed dataset. If two forests – one built from an original dataset and the other one built from a perturbed dataset – are found to be similar then the two datasets are also considered to be similar. The technique estimates the similarity of two forests by estimating the similarity between each pair of intersections; one intersection of the pair belonging to the original dataset and the other intersection belonging to a perturbed dataset. It is worth noting that EDUDS provides us with a single metric for similarity comparison, unlike the approach taken in (Islam and Brankovic, 2011).

EDUDS estimates the similarity of the segments based on each record from an original dataset, D_O. If a record $R_i \in D_O$ falls in (belongs to) the intersections I_O and I_P then the technique measures the similarity of the intersections I_O and I_P, where I_O, and I_P are an intersection in the original dataset and an intersection in the perturbed dataset, respectively. To compare two intersections, EDUDS compares the sub-domains of each attribute to check how closely they match.

Sub-domains of numerical attributes have 2 parts: the lower limit and the upper limit. The difference between the lower limit in an intersection (I_O) in the original dataset and the lower limit in the corresponding intersection (I_P) in a perturbed dataset can easily be calculated. The same process can also be repeated for the upper limits of the segments. By adding these differences together, we can quantify the similarity between an attribute's sub-domains. This value is then normalized by dividing it by the total domain size of the attribute as shown

in Eq. 1 and Table 1. Any attributes not tested by either the original or perturbed intersections are discounted.

$$Diff_a^r = \frac{|LDom_{O,a}^r - LDom_{P,a}^r| + |HDom_{O,a}^r - HDom_{P,a}^r|}{domain_a^r} \quad (1)$$

$$Diff_a^r = \begin{cases} 1 & \text{if} Test(I_{O,a}^r) \vee Test(I_{P,a}^r) = 1 \\ 0 & \text{if} Test(I_{O,a}^r) \wedge Test(I_{P,a}^r) = 1 \\ Discounted & \text{Otherwise} \end{cases} \quad (2)$$

$$SDS_{O,P} = \frac{1}{n}\sum_{r=0}^{n}\frac{1}{k}\sum_{a=0}^{k}(1 - Diff_a^r) \quad (3)$$

In the case of a categorical attribute, the similarity of two intersections is calculated in the same way, except that if one intersection tests the attribute and the other does not then the normalized sub-domain difference is considered to be 1, as shown in Eq. 2. The normalized sub-domain difference is considered to be 0 if the attribute is tested by both segments. Moreover, the attribute is discounted if it is not tested by the logic rules of either segment. As shown in Eq. 3, the process is then repeated for every attribute of the record, and for every record in the dataset.

Not only does the technique have the advantage of using forests, but it can also work equally well with single trees. To the best of our knowledge, none of the existing evaluation techniques attempt to compare forests. It can also often be challenging to find the best matching rules from the original and perturbed trees, such as is required by an existing technique (Islam and Brankovic, 2011). EDUDS handles this problem by calculating similarity on a per-record basis. Using EDUDS on a pair of forests (instead of a pair of trees) can provide a better evaluation of the similarity between two datasets, as forests typically extract a larger spread of logic rules.

5. Evaluation of Data Utility using Splitting Criteria (EDUSC)

Our second proposed data quality evaluation technique is named "Evaluation of Data Utility using Splitting Criteria" (EDUSC), and it can also evaluate data quality using single trees or forests. It can use any (normalized) splitting criteria such as Gain Ratio (Quinlan, 1993), Information Gain and Gini Index (Abellán and Masegosa, 2010). We now describe EDUSC using forests and Gain Ratios, as an example.

We consider that splitting criteria (used in decision tree building), such as the Gain Ratios of attributes, are a strong property of a data segment. Therefore, we compare the segments of a perturbed dataset with the segments of an original dataset through the comparison of Gain Ratios. For all the segments, of an original dataset, that are naturally obtained through a tree, EDUSC explores the corresponding data segments in a perturbed dataset. It does so by artificially/forcefully building a decision forest from a perturbed dataset, where each tree (of the forest) is exactly the same as the corresponding tree in the forest that is naturally built from the original data set. An artificial/forced tree has exactly the same nodes and splitting points as the corresponding natural tree. EDUSC then calculates the Gain Ratios of every attribute within each forced segment/node of the perturbed dataset. Gain Ratios are calculated for leaf nodes as well.

The data segments are compared by the differences of the Gain Ratios for each attribute – provided that at least one of the pair of Gain Ratios for the attribute is over a threshold. This threshold is the same as the user defined minimum Gain Ratio used in building a decision forest by SysFor (Islam and Giggins, 2011).

If $G_{O,s}^a$ and $G_{P,s}^a$ are the Gain Ratios of attribute a in the original (O) and perturbed (P) data segments s then we calculate the similarity of the data segments in terms of attribute a as $1 - |G_{O,s}^a - G_{P,s}^a|$. We repeat this process for all applicable attributes and for all data segments of a pair of trees t (one obtained naturally from the original and the other one obtained forcefully from the perturbed dataset) as shown in Eq. 4 and Table 2.

$$Sim^t = 1 - \frac{1}{S_t}\sum_{s=0}^{S_t}\frac{1}{k * total}\sum_{a=1}^{k}|G_{o,s}^a - G_{p,s}^a| * affected_{O,s} \quad (4)$$

We then repeat the process of calculating similarity according to Eq. 4, for all corresponding pairs of trees of the two forests.

Table 1 Notation table for EDUDS.

Symbol	Meaning
k	Total number of attributes counted
$domain_a^r$	Domain size of attribute a for the segment where the record r falls in in the original dataset
r	A record
$Diff_a^r$	Sub-domain difference based on attribute a for a record r
n	Total number of records
$Test(I_{X,a}^r)$	Returns 1 if attribute a is tested in the segment where the record r falls in, in dataset X; otherwise 0
$LDom_{X,a}^r$	Lower limit of attribute a for the segment where the record r falls in, in dataset X
$HDom_{X,a}^r$	Higher limit of attribute a for the segment where the record r falls in, in dataset X
$SDS_{O,P}$	Sub-domain similarity of the original and the perturbed dataset

Table 2 Notation table for EDUSC.

Symbol	Meaning
S_t	Total number of nodes/segments in the tth tree
$G_{P,s}^a$	Gain ratio of attribute a in segment s of perturbed dataset
s	A node
$affected_{O,s}$	Number of records in the segment s of the original dataset
k	Total number of attributes counted
$total$	Total number of records in original dataset
a	An attribute
Sim^t	Similarity based on the segments of a natural tree t obtained from the original dataset and a forced tree obtained from the perturbed dataset
T_O	Total number of trees in a forest built from the original dataset
Sim	Similarity based on all natural trees obtained from original dataset and forced trees obtained from perturbed dataset
$G_{O,s}^a$	Gain ratio of attribute a in segment s of original dataset

Using all the natural trees obtained from the original dataset and the forced trees obtained from the perturbed dataset, a similarity of two datasets is then calculated by averaging the similarities for each pair of trees, as shown in Eq. 5.

$$Sim = \frac{1}{T_O}\sum_{t=0}^{T_O} Sim^t \tag{5}$$

6. Security Evaluation using Record Similarity (SERS)

Following an existing approach (Islam and Brankovic, 2011), we propose a novel security analysis called SERS. Let D^O and D^P be an original and a perturbed dataset, respectively. Let $R_i^O \in D^O$ and $R_i^P \in D^P$ be the ith original and the ith perturbed record, respectively. Also consider that $R_{i,j}^O$ is the jth attribute value of the ith record in the original dataset. Moreover, let $|R_i^O|$ be the number of attributes and $|R_{i,j}^O|$ be the domain size of the jth attribute. If the first N attributes are numerical and the next C attributes are categorical, then we calculate the distance between R_i^O and R_k^P as follows.

$$d_{i,k} = \frac{\sum_{j=1}^{N} \frac{|R_{i,j}^O - R_{k,j}^P|}{|R_{i,j}^O|} + \sum_{j=N+1}^{N+C} d(R_{i,j}^O, R_{k,j}^P)}{|R_i^O|} \tag{6}$$

where $d(R_{i,j}^O, R_{k,j}^P)$ is equal to 0 if $R_{i,j}^O = R_{k,j}^P$; otherwise 1. The similarity of the ith original record and the kth perturbed record is defined as $S_{i,k} = 1 - d_{i,k}$. We then normalize the similarity values as follows, where $|D^O|$ is the number of records.

$$\widetilde{S}_{i,k} = \frac{S_{i,k}}{\sum_{l=1}^{|D^O|} S_{i,l}} \tag{7}$$

Using the normalized similarity $\widetilde{S}_{i,k}$ we then calculate the entropy for each original record as follows. The higher entropy indicates more uncertainty (i.e. security) in record re-identification for the ith original record, which is the target record of an intruder.

$$H_i = -\sum_{k=1}^{|D^O|} \widetilde{S}_{i,k} \log_2 \widetilde{S}_{i,k} \tag{8}$$

The average entropy for all original records ($H = \sum_{i=1}^{|D^O|} \frac{H_i}{|D^O|}$) is then considered as the overall security of the perturbed dataset.

7. Experimental results

We implement our technique, Forest Framework, and compare it to its predecessor, Framework (Islam and Brankovic, 2011). We also compare it to GADP (Muralidhar et al., 1999) and Random Technique (RT) (Islam and Brankovic, 2011) in order to evaluate EDUDS, EDUSC and SERS. The techniques are applied on five real life datasets (see Table 3) that are publicly available from UCI Machine Learning Repository (Bache and Lichman, 2013). All datasets have multivariate normal distribution, and none of the datasets have any missing values.

In our experiment, the SysFor algorithm (Islam and Giggins, 2011) is used to build decision forests. It allows for 6 parameters to be user-defined. For all experiments, we keep 5 of them the same: the minimum gain ratio is 0.01; the confidence factor for pruning is 0.25; the goodness threshold is 0.50; the separation threshold is 0.30; and the number of trees constructed when building a forest is 3. The values for minimum gain ratio, confidence and separation are the values recommended by the authors (Quinlan, 1993; Islam and Giggins, 2011), and were kept at the default values in order to demonstrate the effectiveness of Forest Framework in an ordinary scenario. Goodness threshold was increased from the default 0.30 to 0.50 in order to better guarantee the uniqueness of each of the trees (Islam and Giggins, 2011), thus preventing the logic rules of each tree from being too similar when intersecting them. A major advantage of SysFor is that it discovers the best logic rules in the dataset with a substantially smaller number of trees than competing techniques such as Random Forest (Breiman, 2001), and so the number of trees constructed can be reliably set at 3. We recommend keeping this value low so as to keep the number of intersections manageable. The remaining parameter – minimum number of records in each leaf (min. leaf size) – was altered depending on the size of the dataset, and is presented in Table 3. In order to avoid over-fitting (Quinlan, 1993) it is undesirable to have very small leaves, however datasets with less records will naturally require smaller minimums than larger datasets. For example, the minimum leaf size for Abalone is larger than the entire Wine dataset.

We perturb each dataset by using 4 techniques: Forest Framework (FF), Framework (F), GADP, and Random Technique (RT). For each dataset a 10-fold cross-validation approach is used. That is, for each dataset, we prepare 10 training $\{TR_1, TR_2, \ldots, TR_{10}\}$ and 10 testing $\{TS_1, TS_2, \ldots, TS_{10}\}$ datasets in such a way that each record is included in a testing dataset once. We then perturb each training dataset 10 times using a technique and thereby produce 10 perturbed datasets from each training dataset for each technique.

We build decision forests from each original training dataset (TR_i) and each of the ten corresponding perturbed datasets $\{P_1^i, P_2^i, \ldots, P_{10}^i\}$. The decision forests are then evaluated using Prediction Accuracy on TS_i, EDUDS and EDUSC. The average accuracy of all ten decision forests built from $P_j^i; \forall j$ is considered as the accuracy of the ith fold. These same decision forests are then compared to the decision forest built

Table 3 Details of the datasets.

Name	Records	Numerical attributes	Categorical attributes	Class size	Minimum leaf size
Wine	178	12	1	3	10
WBC	683	9	1	2	45
Credit	653	6	10	2	30
CMC	1473	2	8	3	60
Abalone	4177	7	2	28	180

from TR_i, using EDUDS and EDUSC. The perturbed datasets are also compared against the original datasets using SERS. Similar tests are carried out for all 10 folds and then averaged. Therefore, for each perturbation technique we use 100 (i.e. 10×10) tests for each evaluation criteria.

Fig. 2 shows the average Prediction Accuracy of the perturbation techniques on the datasets, using Credal Voting (Abellán and Masegosa, 2010) to consolidate the predictions of each of the decision trees into a single prediction. The experimental results indicate that the accuracies of the forests built from datasets perturbed by Forest Framework (FF) are higher than those of Framework (F) and GADP for all datasets except Credit, and are clearly higher than those of RT for all datasets. "*Original*" in Fig. 2 refers to the accuracy of the forest built from the original training dataset and applied on the original testing dataset. Abalone reports a much lower *Original* Prediction Accuracy due to the large number of class values – in general it is much easier to predict one value out of two or three possibilities than it is to predict one value out of 28 possibilities.

Fig. 3 shows the similarity, between the original and perturbed datasets, using the EDUDS technique. Forest Framework performs better than the other three techniques for all datasets, except Abalone. We can see that when evaluating the information quality of perturbed datasets, Prediction Accuracy fails to distinguish between Framework and Forest Framework for four out of five datasets. However the EDUDS results demonstrate that there is a difference in the structure of the decision trees created by the perturbed datasets when comparing Framework and Forest Framework, for three of the datasets. Additionally, EDUDS shows a sharp decline in the similarity between decision trees when using GADP to perturb data, for all datasets. Meanwhile several datasets report GADP having similar Prediction Accuracy results to Framework and Forest Framework. These empirical results support what EDUDS aims to achieve in theory – identify when the logic rules have remained close to their original domains after perturbation. Framework and Forest Framework both aim to preserve the domains, while GADP and RT have no such capability, and this is seen in Fig. 3 by the larger difference between the former and latter groups than the distance within them.

The results of EDUSC are presented in Fig. 4. Forest Framework performs better than the other three techniques for all datasets, as expected. We can also see that EDUSC does not punish GADP in the same way that EDUDS does, since GADP aims to preserve the information quality of the dataset in different ways. As expected, EDUSC shows that GADP can

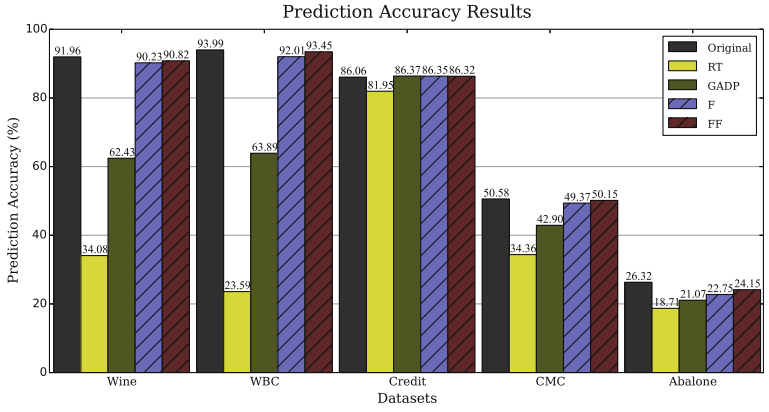

Fig. 2 Prediction Accuracy of the classifiers made from datasets perturbed with the different techniques. The two striped columns signify Framework and its extension, Forest Framework.

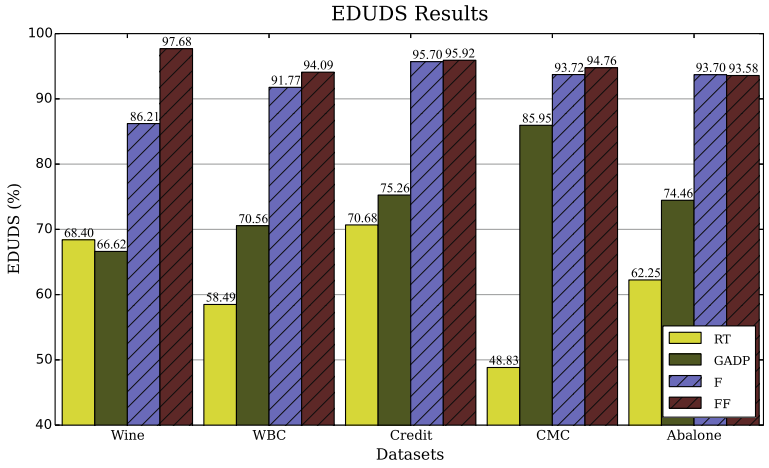

Fig. 3 EDUDS results of the classifiers made from datasets perturbed with the different techniques, compared to their original datasets. The two striped columns signify Framework and its extension, Forest Framework.

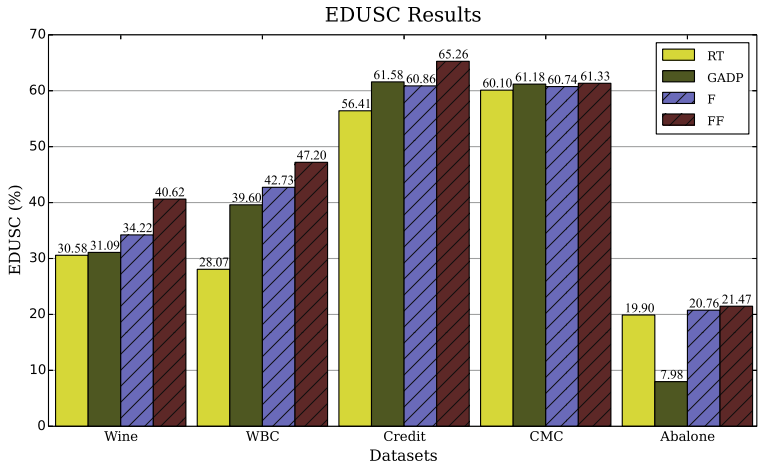

Fig. 4 EDUSC results of the classifiers made from datasets perturbed with the different techniques, compared to their original datasets. The two striped columns signify Framework and its extension, Forest Framework.

Table 4 SERS results for the perturbed datasets. Lower values indicate less entropy, and thus less security.

	Wine	WBC	Credit	CMC	Abalone
RT	7.317	9.568	9.165	10.309	11.859
F	7.315	9.248	9.174	10.333	11.860
FF	7.314	9.243	9.176	10.327	11.860
GADP	7.316	9.241	9.171	10.329	11.836

maintain the quality of the splitting criteria values reasonably well, but not as well as Framework in most cases and Forest Framework in all cases. Interestingly RT does not fall as much as expected, indicating that it is perhaps not as destructive of the patterns as first thought. This theory is supported by the occasionally high Prediction Accuracy results reported by RT. It is worth noting that EDUDS and EDUSC provide us with a single value for similarity comparison, unlike some of the measures used in the original Framework paper (Islam and Brankovic, 2011).

If a data miner only intended to use decision trees as a "black box", where they did not care about the patterns themselves, Prediction Accuracy might be enough. However one of the major strengths of decision trees is their ability to provide humanly readable patterns discovered in the data, thus providing analysts with valuable information for making decisions and guiding further exploration. By being able to detect changes in the structure of the created decision trees, EDUDS and EDUSC provide additional valuable information to the privacy preserving data miner.

Table 4 presents our security evaluation based on the results of SERS, where the security level achieved by FF is comparable with the other techniques, including RT. Therefore, FF preserves significantly better data quality than other perturbation techniques at the cost of only slightly worse security.

8. Conclusion

We present a novel noise addition technique called Forest Framework, two novel data quality evaluation techniques called EDUDS and EDUSC, and a novel security evaluation technique called SERS. We compare Forest Framework with three existing techniques (Framework, GADP and RT) based on four evaluation criteria – Prediction Accuracy, EDUDS, EDUSC and SERS. We carry out 100 tests for each technique on each dataset, following a 10-fold cross validation approach. Our experimental results indicate that Forest Framework preserves data quality better than the other three perturbation techniques.

EDUDS and EDUSC provide us with a mechanism to evaluate data quality through the similarity analysis of datasets in a natural way using tree similarity, instead of the prediction accuracies of the trees. Prediction Accuracy is not always considered to be a good evaluation of data quality (Islam and Brankovic, 2011; Islam, 2007; Lim et al., 2000).

Moreover, our experimental results indicate that EDUDS and EDUSC successfully identify RT (which is designed to be a low quality technique for comparison purposes only) as inferior to FF for all datasets. In fact, RT performs the worst for all datasets overall (see Fig. 3 and Fig. 4). EDUDS and EDUSC also show close results for the two comparable techniques, FF and F, as expected. This substantiates the effectiveness of EDUDS and EDUSC in capturing changes in data quality that would be otherwise missed. We plan to carry out more experiments to explore the impact of varying amounts of noise on data quality and security.

Acknowledgment

Many thanks to A/Prof Ljiljana Brankovic for her time during a discussion of the techniques and results.

References

Abellán, J., Masegosa, A., 2010. An ensemble method using credal decision trees. Eur. J. Oper. Res. 205, 218–226.

Adam, N., Worthmann, J., 1989. Security-control methods for statistical databases: a comparative study. ACM Comput. Surv. (CSUR) 21.

Arnott, C., 2011. Internet Privacy Research. Technical Report 1. The University of Queensland, Brisbane.

Bache, K., Lichman, M., 2013. UCI Machine Learning Repository. URL <http://archive.ics.uci.edu/ml/>.

Brankovic, L., Islam, M.Z., Giggins, H., 2007. Privacy-preserving data mining. In: Petkovic, M., Jonker, W. (Eds.), Security, Privacy and Trust in Modern Data Management, Springer, chapter 11, pp. 151–166.

Breiman, L., 2001. Random forests. Mach. Learning, 1–35.

Dankar, F.K., Eman, K.E., 2012. The application of differential privacy to health data. In: 5th International Workshop on Privacy and Anonymity in the Information Society. ACM, Berlin, Germany, pp. 158–166.

Farkas, C., Jajodia, S., 2002. The inference problem: a survey. ACM SIGKDD Explor. Newsletter 4, 6–11.

Friedman, A., Schuster, A., 2010. Data mining with differential privacy. In: 16th SIGKDD Conference on Knowledge Discovery and Data Mining. ACM, Washington, DC, USA, pp. 493–502.

Giggins, H., 2012. VICUS: a noise addition technique for categorical data. In: Zhao, Y., Li, J., Kennedy, P., Christen, P. (Eds.), Tenth Australasian Data Mining Conference-Volume 134. Australian Computer Society Inc., Sydney, Australia, pp. 139–148.

Islam, M.Z., 2007. Privacy Preservation in Data Mining Through Noise Addition. Ph.D. thesis. University of Newcastle, Newcastle.

Islam, M.Z., Brankovic, L., 2011. Privacy preserving data mining: a noise addition framework using a novel clustering technique. Knowledge-Based Syst. 24, 1214–1223.

Islam, M.Z., Giggins, H., 2011. Knowledge discovery through SysFor: a systematically developed forest of multiple decision trees. In: Ninth Australasian Data Mining Conference-Volume 121. Australian Computer Society Inc., Ballarat, Australia, pp. 195–204.

Lim, T., Loh, W., Shih, Y., 2000. A comparison of prediction accuracy, complexity and training time of thirty-three old and new classification algorithms. Mach. Learning 40, 203–228.

Muralidhar, K., Sarathy, R., 2005. An enhanced data perturbation approach for small data sets. Decis. Sci. 36, 513–529.

Muralidhar, K., Parsa, R., Sarathy, R., 1999. A general additive data perturbation method for database security. Manage. Sci. 45.

Ntoutsi, I., Kalousis, A., Theodoridis, Y., 2008. A general framework for estimating similarity of datasets and decision trees: exploring semantic similarity of decision trees. In: 2008 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, Atlanta, Georgia, United States, pp. 810–821.

Osei-Bryson, K.M., 2004. Evaluation of decision trees: a multi-criteria approach. Comput. Oper. Res. 31, 1933–1945.

Quinlan, J.R., 1993. C4.5: Programs for Machine Learning. 1st ed., Morgan Kaufmann.

Quinlan, J.R., 1996. Improved use of continuous attributes in C4.5. J. Artif. Intell. Res. 4, 77–90.

Ray, S., Nizam, M., Das, S., Fung, B., 2011. Verification of data pattern for interactive privacy preservation model. In: Proceedings of the 2011 ACM Symposium on Applied Computing. ACM, pp. 1716–1723.

Sarathy, R., Muralidhar, K., Parsa, R., 2002. Perturbing nonnormal confidential attributes: the copula approach. Manage. Sci. 48, 1613–1627.

Wang, T., Liu, L., 2011. Output privacy in data mining. ACM Transact. Database Syst. (TODS) 36, 1–34.

Wu, X., Ying, X., Liu, K., Chen, L., 2010. A Survey of Privacy-preservation of Graphs and Social Networks. Managing and Mining Graph Data.