# Noise Addition for Protecting Privacy in Data Mining

*Md. Zahidul Islam and Ljiljana Brankovic*

School of Electrical Engineering and Computer Science, The University of Newcastle, Callaghan, NSW 2308, Australia
E-mail: zahid@cs.newcastle.edu.au; lbrankov@cs.newcastle.edu.au

## 1 Introduction

In recent years advances in technology facilitated collection and storage of vast amount of data. Many organizations, including large and small businesses, and hospitals and government bodies rely on data for day-to-day operation as well as marketing, planning and research purposes. Examples include criminal records used by law enforcement and national security agencies, medical records used for treatment and research purposes and shopping records used for marketing and enhancing business strategies. The benefits of the information extracted from such data can hardly be overestimated. For example, we are all witnessing huge progress made in the human genetic project bringing new promises of previously unimaginable treatments such as gene therapy.

However, simultaneously with the data explosion there is an uprise of anxiety about the confidentiality of delicate personal information open to potential misuses. This is not necessarily limited to data as sensitive as medical and genetic records mentioned above. Other personal information, although not as vulnerable as health records, is also considered to be confidential and as such is open to malicious exploitation. For example, detailed credit card records can be used to monitor personal habits.

The IBM Multinational Consumer Privacy Survey performed in 1999 in Germany, USA and UK illustrates public concern about privacy [6]. Most consumers (80%) feel that "consumers have lost all control over how personal information is collected and used by companies." The majority of consumers (94%) are concerned about the possible misuse of their personal information. This survey also shows that, when it comes to the confidence that their personal information is properly handled, consumers have most trust in health care providers and banks and the least trust in credit card agencies and Internet companies.

Personal data are typically collected with the consent (presumed or otherwise) of the subject. It seems that the main public concern comes from so-called secondary use of personal information without the consent of the subject, that is, any use other than the one for which the data were originally collected. In other words, consumers feel strongly that their personal information should not be sold to other organizations without their prior consent. Indeed, the above-mentioned survey shows that over 50% of respondents have asked a company not to sell their information.

The main concern of collectors and owners of personal records is that public apprehension about privacy may result in difficulties in obtaining truthful information from individuals. Additionally, privacy concerns may lead to future laws and regulations that will restrict and constrain such data collection. In this paper we argue that it is possible to provide confidentiality of individual records and still preserve the usefulness of data for research and planning purposes.

We first note that removing names and other unique identifiers is not enough to ensure the confidentiality of personal records. Privacy invasion is possible whenever a record can be uniquely identified by using a combination of other attributes. For example, an individual, who is the only one with certain characteristics, say age of 25 and salary of 45000, may be uniquely identified and all other characteristics may be learned from other attributes in that record. Thus better techniques are needed to ensure privacy and a lot of work has been already done in the area of statistical databases (see, for example, [10,12,13]). In this paper we focus on records used for building decision trees and we develop techniques for adding noise to class and other attributes, using various probability distributions. We show that decision trees built from the perturbed records are the same or very similar to the trees built from the original records.

## 2 Previous Work

There are numerous research papers describing various noise addition techniques for protecting privacy in statistical databases [1]. These techniques roughly fall into two broad categories, probability distribution and fixed data perturbation. The probability distribution techniques replace the original values in the database by another set of values drawn from the same distribution. On the other hand, fixed data perturbation techniques add noise to values of the attributes. Typically the mean of the

added noise will be zero so as not introduce bias to the attributes. Nevertheless, most of these techniques still suffer from bias introduced to relationships between different attributes. Tendick and Matlof [12,13] introduced methods that correct the bias problem. They also measured the security by the coefficient of determination (squared correlation coefficient). This measures the proportion of variance of the attribute that a malicious database user can learn from perturbed attribute. If the determination coefficient is close to zero the security is high and if it is close to one the security is low. Thus the security is inversely proportional to the variance of the added noise. On the other hand, the large variance of noise is undesirable as it affects the variance of the attribute. Consequently a balance is needed between these two conflicting requirements.

In the context of data mining it is important not only to preserve statistical parameters such as means and variance but also to preserve patterns that exist in the database. Thus the balance between the pattern preservation and the security is needed. Various approaches have been taken to solve this problem. If a database is shared between several parties in such a way that all shares have the same schema but contain different records (horizontal partitioning of a database), parties may be unwilling to disclose any information about their share of the database including association rules that only hold locally for their share. On the other hand, all parties are interested in obtaining association rules that hold globally. The solutions to this scenario are suggested in [6,9]. For the scenario in which a database is partitioned vertically in two pieces, that is, if the two parties contain different portions of the same records, a solution is proposed in [4]. A very different approach is taken in [11] where a security method, in multi relational association rules mining, that relies on access control is described. Yet another technique based on noise addition [2] perturbs the original values in the database and then reconstruct the distributions of the original data values. The classifiers are built on these reconstructed distributions and then experimentally compared with the classifiers built on the original data.

In the technique proposed in [5] the noise is added to the class rather than other attributes in the data set. As the class is typically a categorical attribute containing just two different values, the noise was added by changing the class in a small number of records. This was achieved by randomly shuffling the values of the class in the heterogeneous leaves. It was experimentally shown that the decision trees built on the perturbed data are very similar to the decision trees built on the original data. In this paper we perturb attributes other than the class in such a way to preserve the patterns discovered by the original tree. Additionally, we repeat experiments from [5] using three different perturbation methods as described in the next section.

## 3 Our Work

Classification is one of the most important tasks in data mining. The training data consist of a set of pre-classified cases (records) where each case consists of several attributes and a class. The process of building a classifier takes as input a set of pre-classified cases and produces as output a classifier, which is then used to assign a class to new cases. During the process of building a classifier, data are typically divided into training set, used for building the classifier, and the testing set, used for evaluating it. There are various classifiers including decision trees, Bayesian classifiers and neural networks.

In this paper we concentrate on classification by decision trees. Each node of the tree tests a value of an attribute or a collection of attributes. Depending on the outcome of the test and labels on the edges from that node, one of the edges is taken and the subsequent test is performed in the next node. If the attribute tested in the node is numerical then typically there are two edges from the node. One of the edges is labelled ">c" while the other edge is labelled "<=c", where c is a constant from the domain of that attribute. If, on the other hand, the attribute is categorical then there are typically a few edges from the node, each labelled by a category from the attribute domain. Each leaf of the tree has a class associated with it. In homogeneous leaves the leaf class appears in all the cases from the training set that belong to that leaf. In heterogeneous leaves majority of the cases from the training set belong to the leaf class and only a few belong to another class.

When the tree is used to classify a new case, the case is first tested in the root of the tree, one of the edges from the root is taken and the new test is performed in the subsequent node. This procedure is repeated until the case arrives in one of the leaves of the tree and the leaf class is assigned to the case.

An example of a decision tree is shown in Figure 1. The tree was built on the "Boston Housing Price" dataset which is available from the UCI Machine Learning Repository at http://www.ics.uci.edu/~mlearn/MLRepository.html. Each case corresponds to a suburb and attributes include "average number of rooms per dwelling" and "percentage lower income earners". The class refers to the median house price in a suburb and has two values, "top 20%" and "bottom 80%". The root node tests the attribute "av rooms per dwelling". This is a numerical attribute and the left edge from the root denotes the values greater than 7.007 while the right edge denotes values less than or equal to 7.007. If the left edge is followed we

arrive in a heterogeneous leaf containing 33 cases from the training set where all but one belong to the "top 20%". The reader is invited to inspect the Figure 1 for more details.

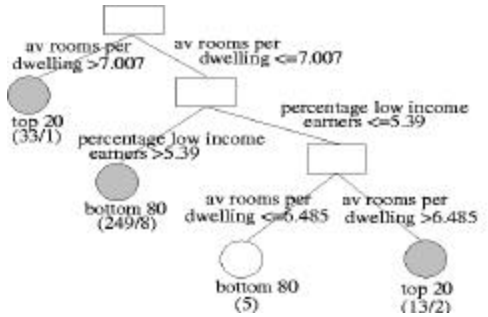


FIGURE 1. An example of a decision tree. Squares represent internal nodes, unshaded circles represent homogeneous leaves and shaded circles represent heterogeneous leaves.

Typically, the class attribute is confidential and should not be learnt by users of the data set. On the other hand, data miners need to have access to the data set in order to build a classifier. If they can identify a particular case in the training set then they can learn the confidential class of that case. Our goal is to perturb the collected data in such a way so that it hides the individual information and at the same time it preserves the patterns for the miners. We argue that data miners do not need to have access to 100% accurate data in order to construct classifiers. In fact, that is never the case because of the natural noise occurring in the data.

We consider adding noise in two different ways. Firstly, we follow [5] and add noise to the class attribute. If a user of a data set can uniquely identify a case in the set and wants to learn the class of the case, they will have some uncertainty about the class because of the added noise. Secondly, we add noise to attributes other than the class. This prevents a user from identifying a case with certainty and thus from learning the class. In both cases our goal is to hide individual information while preserving the original patterns in the data set. We evaluate the quality of our methods by using two indicators, data quality and security level.

We measure data quality by the similarity between the tree produced from the original data and a tree produced from the perturbed data. We note that we do not evaluate the precision of trees obtained from perturbed data sets. We rather evaluate trees by comparing the logic rules associated with them.

We measure the security by the uncertainty that users have in estimating the confidential class of a case. We present results in this section.

*3.1 Adding noise to the class.* Following [5], we add noise to the class attribute by perturbing the class of the cases in heterogeneous leaves of the original decision tree. The cases from homogeneous leaves of the original tree remain unchanged. A perturbed data set is what is released to data miners.

We use the following notation:

***H*** - the number of heterogeneous leaves

$\boldsymbol{m_k}$ - the number of majority cases in the kth heterogeneous leaf, 1<=k<=H

$\boldsymbol{n_k}$ - the number of minority cases in the kth heterogeneous leaf, 1<=k<=H

***E(N)*** – the expected number of changed classes

We applied three different noise addition techniques, which we call *Random, Probabilistic* and *All Leaves Probabilistic.* In each of these techniques the same amount of noise is added where the amount of noise is measured by the expected number of changed classes *E(N)*:

$$E(N) = \sum_{k}^{H} \frac{2m_k n_k}{m_k + n_k}.$$

In *Random technique*, in each heterogeneous leaf $k$ all $n_k$ cases with minority class are first converted into majority class. Then $n_k$ cases are randomly selected from the leaf and are changed to minority class. This change is made in all heterogeneous leaves. Let

$$p_k^i = \frac{\binom{n_k}{i}\binom{m_k}{n_k - i}}{\binom{m_k + n_k}{n_k}}$$

be the probability that in leaf $k$, $i$ minority classes will remain unchanged. Then the expected number of changed classes is

$$E(N) = 2\sum_{k=1}^{H}\sum_{i=1}^{n_k} p_k^i (n_k - i) = \sum_{k}^{H} \frac{2m_k n_k}{m_k + n_k}.$$

However, the probability that a class has been perturbed is not uniformly distributed over all the cases in the heterogeneous leaves. The cases that have minority class in the perturbed data set are perturbed with the probability $m_k / (m_k + n_k)$, while the cases with majority class in the perturbed data set are perturbed with the probability $n_k / (m_k + n_k)$. An intruder's best strategy is to assume that the cases in the $k^{th}$ heterogeneous leaf belong to the majority class with probability $m_k / (m_k + n_k)$. In other words, an intruder has no way of identifying cases, which originally belonged to the minority class. Thus the security of these cases is very high. On the other hand, security of cases that belong to the majority class is very low while the security of cases that belong to homogeneous leaves is zero.

We argue that if a case belongs to a homogeneous leaf then the value of the class attribute is consistent with a strong pattern identified by the decision tree, and it is very difficult to hide it. For example, if it is a common rule of a country that all citizens must retire by the age of 60 and after that they receive a fixed amount of living assistance from

the government, then there is nothing to hide about the monthly salary/income of a person who is over 60 years of age. In this case the pattern is very strong and it is likely to be commonly known. Moreover, there will be no difference between the confidentiality of the cases from the training set and any other cases as the very fact that the age is over 60 determines the salary.

In *Probabilistic technique*, all cases with minority class in a heterogeneous leaf are changed to cases of majority class. Then the class of all cases in the heterogeneous leaf $k$ are changed to minority class with a probability $p_k = n_k / (m_k + n_k)$ and hence

$$E(N) = \sum_{k}^{H} \frac{2m_k n_k}{m_k + n_k}.$$

Although the expected number of changed classes is the same as in the *Random technique*, the security is slightly higher as the intruder does not know the exact probability that a given case belongs to the majority class. For the leaf $k$, this probability is drawn from the binomial distribution with the mean $m = 2m_k n_k / (m_k + n_k)$. As we shall see in the next section, our experiments indicate that the data quality is also slightly worse than in the *Random technique*.

In *All Leaves Probabilistic technique* we perturb all the cases of the dataset, instead of just the cases within heterogeneous leaves. We use this method as a simulation of a natural noise occurring in the class attribute. We compare our other techniques to this one in order to evaluate how good they are in respect to pattern preservation.

We change the class of all cases in the data set with the probability

$$p = N_{Total} \sum_{k}^{H} \frac{2m_k n_k}{m_k + n_k}$$

where $N_{Total}$ is the total number of cases in the data set. The expected number of changed classes is

$$E(N) = \sum_{k}^{H} \frac{2m_k n_k}{m_k + n_k}.$$

We again measure security by the probability that a class value in the perturbed file is the same as the corresponding value in the original file. This probability is now uniformly distributed over all cases in the data set and is equal to

$$p = N_{Total} \sum_{k}^{H} \frac{2m_k n_k}{m_k + n_k}.$$

We recall that the security of cases in homogeneous leaves in the previous two techniques is zero while the security of cases with minority class in the original data set is very high. For the reasons given above we consider this distribution more favourable than the uniform distribution.

We note that the total number of cases having a particular class remains the same in the perturbed file when we use the *Random* technique. This is not the case with *Probabilistic* and *All Leaves Probabilistic* techniques. In the next section we present the experimental results that evaluate the similarity of trees obtained from perturbed data with the tree obtained from the original data. As expected we find that heterogeneous leaves which resided in the deep part of the decision tree appear to be the most sensitive to noise.

*3.2 Adding noise to other attributes.* In order to provide better privacy we perturb as many other attributes as possible, along with the confidential class attribute. We next introduce some terminology used in the remaining of this section. Let an *Innocent* attribute for a particular leaf be an attribute that is not tested in any of the nodes on the path between root and the leaf in a decision tree. Innocent attributes do not play a role in classifying cases belonging to the corresponding leaf. Some of these *Innocent* attributes may be totally absent from the tree in which case we refer to them as *Total Innocent Attributes (TIA)*. Other innocent attributes may appear in some other parts of the tree in which case they are called *Leaf Innocent Attributes (LIA)*.

We note that attributes other than the class are more often than not numerical attributes. For each leaf, we perturb LIAs by adding a random noise drawn from the normal distribution. We add noise to the TIAs in the same way.

Let us denote an attribute (LIA or TIA) by A. After the noise is added the value of the attribute will be $A^* = A + \varepsilon$, where $\varepsilon$ is a discrete noise with mean value of zero and variance of $\sigma$.

As in the previous subsection we evaluate the data quality by the similarity between decision trees built on perturbed data set and the decision tree built on the original data set. We present our experimental result in the next section.

To evaluate the security of our method, we consider the following scenario. Assume that an intruder is interested in learning the confidential class of case X from the data set. We may assume that the intruder has some previous knowledge about the case X as otherwise they would not be able to identify the case and learn the class.

We shall first assume that an intruder can uniquely identify record X by knowing the values in LIAs and TIAs. After the noise has been added to LIAs and TIAs the intruder is not able to uniquely identify the record any more. They can, however, estimate the probability $p(X \rightarrow Y)$ that a case Y in the perturbed data set corresponds to the original case X, assuming of course that the probability distribution of the added noise is known to them. We can express $p(X \rightarrow Y)$ as follows:

$$p(X \rightarrow Y) = \prod_{i=1}^{k} p_i(A_i^* - A_i)$$

where $p_i(A_i^* - A_i)$ is the probability that the noise added to the attribute $A_i$ is equal to $(A_i^* - A_i)$. If the data set is dense, that is, if there are many cases with similar values in TIAs and LIAs then there will be many cases $Y_i$ in the perturb file with a similar probability $p(X \rightarrow Y_i)$. In general, these cases will belong to different leaves with different leaf classes and intruder will have a great deal of uncertainty about the class of X.

We shall next assume that an intruder can uniquely identify record X by knowing the values of attributes different from LIAs and TIAs. The intruder will still be able to identify the case X as no noise has been added to attributes other than LIAs and TIAs, and thus to learn the class of X. We, however, argue that in this case the cases from the training set are not under greater privacy threat than any other case of that kind. Indeed, the intruder can run the classifier on any record for which he knows attributes other than LIAs and TIAs and obtain pretty good estimate of the class. One can argue that classifiers are typically more accurate when applied to the cases from the training set than when applied to other cases. Adding noise to the class would annulate this fact.

## 4 Experimental Results

We used two different data sets, namely *Boston Housing Price* datasets with 300 cases and *Wiscon-sin Breast Cancer (WBC)* data sets with 350 cases. Both data sets are much used by the data mining community and are available from the UCI Machine Learning Repository at http://www.ics.uci.edu/~mlearn/MLRepository.html. We first build decision trees on the original data for both data sets. We used Quinlan's famous decision tree builder *C5*. The decision tree built from the unperturbed 300 cases of Boston *Housing Price* dataset is shown in Figure 2.

We then perturbed the class by applying *Random* perturbation technique on both data sets. We repeated this process 5 times and each time we built a decision tree on the perturbed data set.

After careful analysis we found that trees built from the perturbed data set are very similar to the tree produced from the original data set. Out of four attributes represented in the original decision tree, three appear in all decision trees on perturbed data. The constants used for these three attributes are very similar in all trees. For example "percentage low income" earners ranges from 5.39 to 5.49. The attribute from the original tree, which is not represented in other trees, is only tested towards the bottom of the original decision tree and caters for only 11 cases out of 300. Ignoring logic rules including that attribute, 4 other rules from the original tree appear in the same or very similar form in all perturbed trees.

```
percentage low income earners >5.49:
:. av rooms per dwelling <= 7.041: bottom 80% (248/8)
:  av rooms per dwelling >7.041 : top 20% (10)
percentage low income earners <=5.49:
:..av rooms per dwelling <= 6.485: bottom 80% (5)
  av rooms per dwelling > 6.485 :
   :..pupil-teacher ration <=17.8 : top20% (26)
     pupil-teacher ratio > 17.8 :
     :..nitric oxides ppm <= .4161 : bottom 80 % (2)
       nitric oxides ppm > .4161:
       :..percentage low income earners <=4.45: top 20% (6)
         percentage low income earners > 4.45 : bottom 80%
(3/1)
```

FIGURE 2. Decision tree produced from the 300 cases of *Boston Housing Price* dataset.

Then we applied *Probabilistic* perturbation method on the original data set and we built a decision tree on the perturbed data set. We repeated the experiment 10 times. The analysis of the trees shows similar results as in the previous experiments. Out of 4 attributes from the original tree 2 appear in all other trees while the 3rd one appears in 8 out of 10 trees. The logic rules from the original tree (with the exception of rule involving *nitric oxides ppm*) appear in the same or similar form in most trees. However, 4 of the trees contain new rules that do not appear in the original tree and have significant number of cases belong to those new rules.

Finally, we applied our "All Leaves Probabili-stic" perturbation technique and generated 10 trees based on 10 perturbed data sets. We found that these trees are significantly different from the original tree. Still all of them contain the two most significant attributes from the original tree and 7 out of 10 trees contain the 3rd attribute. Some of these trees are much deeper than the original tree and contain quite a few new rules. On the other hand, some of the trees are very shallow and test only two attributes. We remark that *All Leaves Probabilistic* technique is somewhat unpredictable in respect to generated decision trees.

We then experimented with the attributes other than the class attribute. First of all, we perturbed the Total Innocent Attributes (TIA). We add noise that has mean of zero and standard deviation of 27.6% of the attribute value. The probability distribution of noise is given in Figure 3.

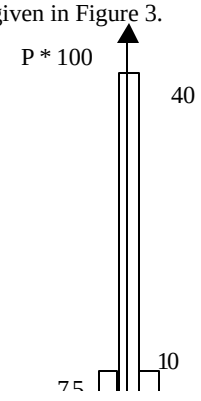

FIGURE 3. Probability distribution of the noise added to TIAs and LIAs. Horizontal axis represents noise as a percentage of the attribute value.

We experimented on 350 cases of WBC data set. We first added noise only to cases in heterogeneous leaves. Then we added noise to all cases of the data set. We ran in total 20 experiments and we found that the trees created from the perturbed dataset were similar to the tree created from original data set. All trees except for one had the same root as the original tree. Most logic rules remained the same as in the original tree. Some trees introduced new attributes, but used them only in deep nodes.

Then we focused on the Leaf Innocent Attributes. We used the same probability distribution described in Figure 3. In 8 out of 10 experiments we obtained trees very similar to the original one and to those from previous experiments. Only two trees are slightly different but they still retain the same root and some logic rules.

## 5 Conclusions

In this paper we introduced a new method for adding noise to data sets used for building decision trees, in order to protect the confidentiality of individual cases. We perturbed attributes other than the class and only those that appear irrelevant for determining the class for the case. Our method works best for dense data sets, which contain many records with similar values of these attributes. We experimentally confirm that this perturbation does not affect the resulting decision tree significantly. Additionally, we performed perturbation of the class suggested in [5] using 3 different techniques. Our experiments confirm their findings.

All the experiments presented in this paper used C5 software for building decision trees from both original and perturbed data sets. The experiments published in [5] used three different decision tree builders (C5, CN2 and EVOPROL) and obtained similar results with all of them. Thus we expect that our noise addition method will produce equally good results in the case where decision trees on perturbed data sets are built with software different from that used for building the original decision trees.

The data sets used in our experiments are among the best known and the most frequently used by the machine learning and data mining community. Hence, we are confident that similar results would have been obtained on other data sets that may be significantly larger in size (10, 000 cases or more).

In general, decision trees are considered sensitive to noise in the sense that small changes in a data set may result in significant differences in the corresponding decision tree and associated logic rules [8]. This is also confirmed by the results of our *"All Leaves Probabilistic"* method, where we obtained significantly different and somewhat unpredictable decision trees. We use this method to evaluate our other methods and we find that our other methods are significantly better, in preserving the decision tree and the associated rules, than this method. Our future work will include experiments with larger data sets and other decision tree builders.

## References


[1] Adam, N. and Wortmann, J.C., Security Control Methods for Statistical Databases: A Comparative Study, *ACM Computing Surveys*, **21**(4), 1999, 515-556.

[2] Agrawal, R. and Srikant, R., Privacy-preserving Data Mining, In *Proceedings of the 2000 ACM SIGMOD Conference on Management of Data*, Dallas, Tx, May 14-19, 2000.

[3] Agrawal, R., Srikant, R., Evfimievski, A. and Gehrke, J., *SIGKDD* 02, Edmonton,Alberta,Canada, 2002.

[4] Du, W. and Zhan, Z., *Workshop on Privacy, Security snd Data Mining, at the ICDM 02*, Conferences in Research and Practice in Information Technology, **14**, Clifton, C and Estivill-Castro, V, eds., 2002

[5] Estivil-Castro, V. and Brankovic, L., Data Swapping: Balan-cing Privacy Against Precision in Mining for Logic Rules, M. Mohania and A.M. Tjoa, eds., *Data Warehousing and Know-ledge Discovery DaWaK'99*, *LNCS 1676*, 1999, 389-398.

[6] IBM Multi-National Privacy Survey CONSUMER REPORT, (available from http://www1.ibm.com/services/files /privacy_survey_oct991.pdf), 1999.

[7] Kantarcioglu, M. and Clifton, C., *ACM SIGMOD Workshop on Research Issue in Data Mining of Knowledge Discovery*, 2002.

[8] Li, R.-H., *Instability of Decision Tree Classification Algorithms*, PhD Thesis, University of Illinois at Urbana-Champaign, 2001.

[9] Lindell, Yand Pinkas, B., Privacy Preserving Data Mining, M. Bellare (Ed.): *Proceedings of the Advances in Cryptology - CRYPTO 2000*, LNCS 1880, 2000.

[10] Muralidhar, K. and Sarathy, R., Security of Random Data Perturbation Methods, *ACM Transaction on Database Systems*, **24** (4), 1999, 487-493.

[11] Oliveira, S.R.M. and Zaane, O.R., Foundations for an Access Control Model for Privacy Preservation in Multi-Relational Association Rule Mining, *Workshop on Privacy, Security and Data Mining*, at the ICDM 02, Conferences in Research and Practice in Information Technology, **14**, Clifton, C. and Estivill-Castro, V., eds., 2002.

[12] Tendick, P., and Norman, N.S., Recent Result On The Noise Addition Method For Database Security, *In Proceedings of the 1987 Joint Meetings, American Statistical Association / Institute of Mathematical Statistics. ASA/IMA*, Washington, D.C., 1987.

[13] Tendick, P. and Matloff, N., A Modified Random Perturbation Method for Database Security, *ACM Transaction on Database Systems*, **19** (1), 1994, 47-63.