A Robust Data-obfuscation Approach for Privacy Preservation of Clustered Data

Rupa Parameswaran and Douglas M. Blough
School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, GA
[rupa,dblough]@ece.gatech.edu

Abstract

Privacy is defined as the freedom from unauthorized intrusion. The availability of personal information through online databases such as government records, medical records, and voters' lists poses a threat to personal privacy. Intelligent search engines and data mining techniques further exacerbate the problem of privacy by simplifying access and retrieval of personal records. **Data Obfuscation (DO)** *techniques distort data in order to hide information. One application area for DO is privacy preservation. Many data obfuscation techniques have been suggested and implemented for privacy preserving data mining applications. However, existing approaches are either not robust to privacy attacks or they do not preserve data clusters, thereby making it difficult to apply data mining techniques. The absence of a standard for measuring the privacy provided by the various data obfuscation techniques makes it hard to compare the robustness of the techniques. The main contributions of this paper are (*1*) to propose a data obfuscation technique called* **Nearest Neighbor Data Substitution (NeNDS)***, that has strong privacy-preserving properties and maintains data clusters, (*2*) to define a property called* **Reversibility** *for the categorization and comparison of data obfuscation techniques, in terms of their resilience to reverse engineering, and (*3*) to formally prove that cluster preserving geometric transformations, by themselves are extremely easy to reverse engineer.*

1. Introduction

The concern over privacy of personal and sensitive information has led to the implementation of several techniques for hiding, obfuscating and encrypting sensitive information in databases. The need for privacy has led to the development of several (data obfuscation) DO techniques that provide privacy preservation at the cost of information loss. Most of the techniques cater to specific domains and perform well for a limited set of applications. In the absence of a standard for classifying DO techniques, comparison and performance analysis of the different techniques is not straight-forward. The domain of interest in this research is data mining. Many data mining applications involve learning through cluster analysis. The term *Usability* is used to refer to the usefulness of the transformed data. In this paper, the usability is measured in terms of the preservation of the inherent clustering of the original data. The need for an obfuscation technique that preserves privacy as well as usability of the transformed data has motivated the design, development, and preliminary performance analysis of a robust cluster retaining DO technique in this research. The paper proposes the use of the *Reversibility Property* as a measure of privacy preservation. The privacy provided by the proposed data obfuscation technique *NeNDS* is evaluated and compared with other obfuscation techniques with respect to its reversibility and usability.

The main contribution of this paper is the design, development, and analysis of the proposed DO technique *NeNDS* as well as a hybrid *Geometrically Transformed* version called *GT-NeNDS*. The motivation for the choice of the DO technique as well as the description of the proposed technique is provided in Section 3. The definition of the *Reversibility Property*, the classification of different transformation techniques based on reversibility, and the evaluation of existing DO techniques is provided in Section 4. An experimental analysis of *NeNDS* is carried out in Section 6 to study its cluster preserving characteristics.

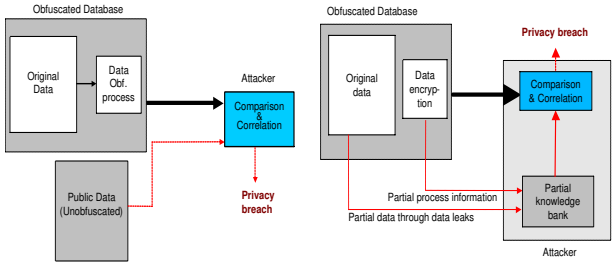
2. Motivation and Related Work

The abundance of information available online has resulted in the loss of individual privacy [5]. Several methods have been proposed and implemented towards privacy preservation of sensitive data sets. DO techniques [4] range from encryption based techniques [1, 16] to geometrical transformation schemes [12, 13]. In the case of encryption based DO techniques, the data is unusable in the encrypted form, and the decryption key for obtaining the original data is provided only to a limited set of users. For sev-

eral applications, it is necessary to provide different levels of precision of data, based on the type of user type of user requesting access. Data encryption does not provide this capability as the data is either usable in its original form or completely unusable. Hence, for trend analysis and statistical and inference-based computations from data sets, encryption-based security schemes add complexity without much benefit in terms of privacy. Geometric transformation schemes, on the other hand, are extremely vulnerable to privacy breaches and provide very little privacy.

Other existing techniques include Data Randomization [2], Data Anonymization [17] [9] and Data Swapping [15]. Data Randomization and Data Anonymization perform obfuscation by "modification" of the original data and do not address cluster preservation. They are also vulnerable to the notion of privacy breaches proposed in [6], which describes a privacy breach as the revelation of any property of the original data in the obfuscated data. One of the techniques that proposes to preserve usability while preserving privacy is Geometric Transformation [12] [13]. While this technique does involve "modifying" of data, the inter-relation of the data elements within the data sets and across the fields are maintained even after the obfuscation. Geometric transformation based DO is very weak in terms of privacy preservation and unsuitable for use in sensitive databases. The concept of data-swapping was first proposed in [15]. This technique intelligently swaps entries within a single field in a set of records so that the individual record entries are unmatched. The reflective nature of data swapping, however, makes it vulnerable to reversal. The requirement of preserving privacy as well as the usability of sensitive data has led to the proposal and development of a robust DO technique called *Nearest Neighbor Data Substitution (NeNDS)*. A hybrid version of *NeNDS* is also proposed here, called *GT-NeNDS*, which provides stronger privacy by combining geometric transformations with *NeNDS*.

The attack model for data obfuscation is different from the attack model for encryption-based security techniques, but no common standard has been implemented as yet for DO. Each of the proposed obfuscation techniques uses a different form of comparison of the effectiveness of the approach. Existing work on the privacy analysis of DO techniques has primarily considered a model where the attacker correlates obfuscated responses with data from other publicly-accessible databases in order to reveal the sensitive information of interest. In this work, we consider a model where the attacker uses side-channels to obtain some partial information about the process used to obfuscate the data and/or some of the original data items themselves. The attacker can then use this partial information to attempt to reverse engineer the entire data set. To motivate this new attack model, we give two concrete examples where partial information can be revealed. In the first example, the database is temporarily left *open*, i.e. without an obfuscation mechanism in place. Before the situation is detected, an attacker can access some unobfuscated data records. Clearly, if the database is extremely large and the problem is discovered quickly, only a small percentage of the database will be revealed in its original unobfuscated form. Such situations can occur due to programming errors, soft failures, or configuration (human) errors. The second example is a large distributed database, e.g. that of an international corporation with many data sites throughout the world. In this situation, an inside attacker will be able to access the unobfuscated information from one data site and might use this to try to reveal information from the remaining sites.



(a) Existing model (b) Proposed Model

Figure 1. Attack Models for Analysis

One useful byproduct of this model is a measure of the robustness of a data obfuscation technique, namely the percentage of the unobfuscated data set that an attacker must know in order to be able to learn the entire set. Using this new measure, we are able to demonstrate that many well-known data obfuscation techniques are highly vulnerable to reverse engineering through unintentional release of only a small percentage of the unobfuscated data set. We also propose to use the amount of information required for reverse engineering as a measure of privacy preservation for this attack model.

3. Proposed Data Obfuscation Technique

This section provides a detailed description of the proposed DO technique called *Nearest Neighbor Data Substitution (NeNDS)*. Applications of the proposed technique lie in sensitive databases that require a data protection technique without loss of information content. Examples of such applications are medical records as well as micro-databases released by the Census Bureau, where the privacy of individuals is important as the correctness of the data provided to the end user [10]. The data substitution

technique proposed here preserves privacy by permuting elements among groups of data items that are close to each other. Data substitution is performed individually for each field (dataset) in the database, and each field is permuted independently of the rest of the fields. *NeNDS* can be used for transformation of any data set that has some notion of distance among the elements. In other words, any dataset that forms a metric space can be transformed using *NeNDS*.

3.1. Nearest Neighbor Data-Substitution - NeNDS

NeNDS is a lossless DO technique that preserves privacy of individual data elements by substituting them with one of their neighbors in the metric space. A set of neighboring data elements are grouped together to form a neighborhood. The minimum number of neighbors that comprise a neighborhood is specified by the parameter c, where $1 < c < N - 1$, and N is the size of the data set. The minimum size of a neighborhood is given as $c + 1$, so that each data element in a neighborhood has at least c neighbors. Hence, the number of neighborhoods in a data set is given by $\lfloor \frac{N}{c+1} \rfloor$. In the case where $c = 1$, each neighborhood would contain at least two neighbors, reducing the substitution technique to data swapping in some cases. The reflective nature of data swapping makes it vulnerable to privacy breaches in case of prior knowledge of some of the elements of the original data set thereof. In order to strengthen the privacy preserving capability of *NeNDS*, c is set to be greater than 1.

The *NeNDS* process involved is explained here with an example database. Each field in the database is treated individually and *NeNDS* transformed independently of other fields in the database. Let Σ represent the entire database that consists of m attributes and n records. The obfuscation technique performs substitutions on data items that lie close to each other within a single attribute field, so that the correlation of the data across the different attributes is not destroyed.

Age	Salary	Location
35	75,000	LA
37	80,000	NY
40	78,000	SJC
42	95,000	SFO

Table 1. Orig. DB

Age′	Salary′	Location
40	80,000	LA
35	95,000	NY
42	75,000	SJC
37	78,000	SFO

Table 2. Trans. DB

In the example in Tables 1 and 2, a simple substitution technique is used, where data substitution is performed with the entire dataset as a whole. A more usable transformed database can be obtained by creating smaller subsets of data that fall within pre-specified intervals. The data substitution algorithm *NeNDS* proposed in this paper uses a two step process for DO: In the first step, the dataset is divided into a finite number of subsets, which is then followed by a data substitution algorithm that is performed on the subsets. The first step, called the neighborhood selection step, proceeds by selecting sets of c non-identical elements that are close to each other in the metric space. Once the neighborhoods are selected, data substitution is performed in parallel on each of the neighborhoods.

The substitution process is performed by determining the optimal permutation set subject to the following conditions: (1) No two elements in the neighborhood undergo swapping, (2) The elements are displaced from their original position, and (3) Substitution is not performed between identical elements. These three conditions ensure that each element in the transformed dataset is different from the original dataset. The restriction on swapping of data makes the transformed data robust to partial reversibility, which is a shortcoming of data swapping techniques. The cluster preservation of the data set depends strongly on the value of c that is selected. Reducing the size of the neighborhoods, however, would lead to fewer elements in each neighborhood, which would fail to provide the necessary protection of privacy. Selecting a small value for c results in a highly cluster preserving database, but limited privacy protection, while a large value of c might render the database less clustered as a result of substitution among neighbors that are further away. The selection of c is specific to the nature of the database and the amount of protection that is required. The effect of c on the computation time and the cluster preservation property is evaluated in Section 6.

The neighborhoods created are likely to be of different sizes depending on the number of identical elements in each neighborhood. The algorithm uses a tree-traversal approach to obtain an optimum substitution pattern. The nodes of the tree correspond to the elements of a single data set with the first element as the root of the tree. The children are ordered from left to right based on their nearness to the parent node. The distance between the parent and child are given along the edge connecting them. A Depth First Search (DFS) approach is used here to traverse the tree. Identical values of grandparent and grandchild nodes imply swapping and thus the node is aborted, followed by back-tracking to the next unexplored node in the subtree. Child nodes with identical values as the parent are aborted as well. Since the tree is finite, DFS will complete and will yield a solution. A maximum edge C_{ME} cost counter is maintained for each path being probed. An optimum substitution pattern is one that has the least cost C_{ME}. The substitution corresponding to the path chosen is the permutation used to replace the original data set. The data substitution process for neighborhoods that contain multiple identical elements is the same as described above, with the additional restriction on identical element substitution - a parent node cannot have an identical child. The existence of such a path with no identi-

cal substitutions is ensured by including a sufficient number of non-identical elements in each neighborhood. In situations where the number of identical elements exceeds the number of non-identical elements in the neighborhood and the neighborhood size can no longer be increased, a preprocessing step is performed, in which a randomized offset is added to the identical data elements, after which the dataset is transformed using *NeNDS*.

3.2. Algorithm for NeNDS

1. For each $i \in [1, m]$ perform steps (a) to (e)
   (a) Divide Σ^i into neighborhoods $1 \ldots r$
   (b) For each NH_j perform steps (i) to (v)
      i. Create a c-ary tree with first element of Σ^i as root
      ii. Order the child of each parent from left to right in increasing order of distance from the parent
      iii. All the children of a parent have a non-zero distance from the parent node
      iv. Mark the distance between parent(p) and child(c) node to the edge connecting them $d_{p,c}$.
      v. Mark child node x as leaf node if node x has appeared earlier in the path
   (c) Only paths that have a depth equal to one less than the number of elements in the neighborhood are selected as candidates for substitution
   (d) The path with minimum value for C_{ME} is selected as the substitution pattern to produce the new data subset.
   (e) If $\Sigma^i \neq \Sigma^m$, apply substitution pattern of Σ^1 to the next Σ^i to be obfuscated

In the algorithm, a database of m attributes and n records is denoted by Σ. Each individual field (also called dataset, attribute) is denoted by Σ^i, where $i \in [1, m]$. c is the threshold for the minimum number of non-identical elements in each neighborhood. *NeNDS* can be performed on any data set in which the elements are related by some notion of distance, and can be expressed as a metric space.The algorithm is run for each field in the database forms a metric space. Each neighborhood is denoted by $NH_j, j \in [1, r]$,and the number of neighborhoods r is dependent on the value of c. The algorithm produces the most optimum substitution transformations subject to the conditions listed previously in this section. A brute-force analysis of the DFS based algorithm for finding the substitution pattern indicates that the algorithm has an exponential order of complexity. However, the heuristic nature of the branch and bound implemented reduces the exponential order of complexity to a much smaller value, which is indicated by the successful completion of *NeNDS* even for large data sets.

NeNDS ensures a completely robust framework for data mining applications by preserving all the information content for cluster preservation and providing a secure and privacy preserving framework for drawing inferences on the data. As *NeNDS* preserves the original values of the data even after transformation, it is still vulnerable to privacy breaches as mentioned in [6]. This type of privacy breach may be unacceptable in highly sensitive databases. Section 3.4 provides a hybrid version of *NeNDS* that preserves all the favorable characteristics of *NeNDS* and also overcomes this shortcoming of *NeNDS*.

3.3. Geometric Transformation Technique

An overview of the geometric transformation based DO proposed in [12] [13] is given here. This approach is of interest in data mining applications due to its inherent cluster preservation property. Hence, this technique will be used as a benchmark to evaluate the cluster retention capability of *NeNDS*. Transformations such as rotation, scaling and translation are used for distorting the data [7].

With geometric transformations, any pair of numerical fields in the database is interpreted as a two-dimensional space and the co-ordinates of the data items are distorted by geometric transformation. The approaches can also be scaled to three or more dimensions without loss of generality. The database is denoted by $D_{d,n}$, where d is the number of attributes and n represents the number of records or entries in the database. The transformations translation, scaling and rotation can be implemented using matrix multiplication. Each of the three transformations can be represented in terms of the equation $[X'\,Y']^T = A[X\,Y]^T + B$. In all of the transformations, A, B are the transformation matrices, (X, Y) are the original data, and (X', Y') are the results of the transformations on the original data. From the description of the transformations, it can be observed that each data set is distorted by the same amount relative to the placement of the individual elements in the set. In this way the clusters are maintained during obfuscation.

3.4. A Hybrid Data Substitution Approach

In this section, we propose a hybrid version of *NeNDS*. In this approach, termed as *GT-NeNDS*, the data sets are first geometrically transformed, and then operated upon by *NeNDS*. *NeNDS* provides a privacy preserving wrapper on the geometrically transformed data. The transformation functions like rotation and translation are isometric in nature, thereby preserving cluster information of the data sets and retaining the nearest neighbor information for the substitution step. In this way, the data can be transformed to a form suitable for use by a third party analyst. The two step transformation results in transformed data that preserve

clustering information, but bear no resemblance to the original database. As a result, *GT-NeNDS* is also robust to the notion of privacy breaches as proposed by [6], making it a suitable candidate for privacy preserving data mining.

4. Reversibility - A Standard for Classification

The proposed DO technique, *NeNDS*, was described in Section 3.1. The absence of a standard for measuring and comparing the privacy provided by different DO techniques, makes it difficult to evaluate the performance of the techniques. The term *Reversibility* is used to denote the property of the DO technique, that dictates the ease or difficulty of the process of reverse engineering obfuscated data. This property, was first proposed in [4], and specifies how robust a given obfuscation technique is in terms of hiding sensitive data. The reversibility property of an obfuscation technique is an indicator of the robustness of its privacy preservation. Cryptanalysis is used for analyzing the security provided by encryption-based techniques [18]. Since encryption is a deterministic and reversible process, cryptanalysis assumes the transformation to be deterministic as well as reversible. However, DO techniques have no such restriction and therefore require a new standard for analysis.

An obfuscation technique that can be reversed with the knowledge of the process, is known as a process reversible transformation function. Process reversible DO techniques are analyzed with respect their vulnerability to complete reversal with little or complete a priori knowledge of the process used for DO. Process reversibility is sub-classified into the following categories.

1. Partial knowledge reversibility: Partial knowledge reversibility implies that a transformation function exhibiting this property can be reverse engineered with the knowledge of either some of the original data entries, or a combination of some original entries of data and some information regarding the process used. The level of difficulty of the reversal process is dependent on the DO technique. Obfuscation techniques that involve a *one-to-one* mapping between the original and the transformed data, are vulnerable to partial knowledge reversibility. The reversibility analysis for linear and non-linear one-to-one transformations is provided in Section 5.2.

2. Random number reversibility: This property indicates that the original data set can be reverse engineered with knowledge of the process, the *Pseudo-Random-Number Generator (PRNG)* and the seed. Most obfuscation techniques invoke PRNGs to generate random sequences. The robustness of DO techniques exhibiting this property relies in protecting the PRNG sequence. As long as the random seed and the sequence are unknown to the attacker, the obfuscated data is robust to reversal. Once this information is revealed, and the obfuscation process is known, the entire data is compromised. Transformations that fall under this category cannot be analyzed using cryptanalysis due to their non-deterministic nature.

Obfuscation techniques that result in a non-invertible transformation exhibit Irreversibility. A *Maximum likelihood reversibility* estimate can be made in the case of some of the techniques, which provides an estimate of the confidence with which a guess can be made on the original data. Cryptanalysis fails to account for such transformations as well. With irreversible techniques, there is an inherent loss of information. Lossy compression techniques and data generalization techniques, which make it impossible to exactly recover the original data, fall under this category.

5. Reversibility Analysis

Section 4 provides a classification of all transformation functions based on their reversibility property. Random data perturbation techniques are hard to reverse because they exhibit random number reversibility. Geometric transformations, being linear *one-to-one* transformations can be reversed with the knowledge of a finite number of original records. *NeNDS* involves a non-linear *one-to-one* transformation, and hence can also be reversed with the knowledge of sufficient number of original records. In this section, we derive the value for the minimum number of original records that are required to reverse engineer data that is obfuscated using Geometric Transformations and *NeNDS*.

5.1. Analysis of Geometric Transformations

Geometric transformations fall under the category of linear transformation functions. These functions are the most vulnerable DO techniques that are subject to partial reversibility. A cryptanalysis of linear geometric transformations renders it weak to cipher-text only attacks. The knowledge of the type of obfuscation technique used results in an immediate reversal of the data. The linearity property of this data obfuscation technique preserves the clustered nature of the data, but also results in weak privacy protection. The assumption made here is that the attacker is aware that the DO process is a linear transformation. In this case, we prove that for a database with $d * n$ entries, where d is the number of attributes and n is the number of records, the knowledge of only $d + 1$ affinely independent [11] records in the original matrix, is sufficient to uniquely determine the linear transformation. Once the transformation matrix is obtained, all the original data entries for which the obfuscated values are available, are compromised. Therefore the

Geometric Transformations of [12] [13], being instances of linear transformation functions, are compromised with the knowledge of $d+1$ affinely independent records in the original data, which is a proved fact.

5.2. NeNDS versus Data Swapping

Data Swapping as well as *NeNDS* fall under the category of Non-linear bijective transformations. In this type of transformation, reversibility is dependent on the minimum number of records r that are sufficient for complete reverse engineering. In the case of data swapping, the minimum value for r is half the number of elements in the data set. For each element in the data set that is known a priori, the corresponding element involved in the swap is revealed. In the case of *NeNDS*, complete reversal of the entire data set would require the knowledge of at least $r = c$ distinct data elements for each neighborhood, where $c + 1$ is the minimum size of a neighborhood. Even partial reversal of a neighborhood would require the knowledge of c of its elements. The fraction $\frac{c_i}{c_i+1}$ determines the ease of reversal of a specific neighborhood i having exactly c_i elements. The robustness of the obfuscation technique proposed increases with larger values of c. For the case where $c = 1$, data substitution is reduced to data swapping. In this case, the complexity of reversal is reduced to $1/2$. For all values of $c > 1$, *NeNDS* provides a more robust DO, since complete reversal would require the knowledge of at least $\frac{\sum_{i=1}^{NH}\frac{c}{c+1}}{NH} \geq \frac{c}{c+1}$, where NH is the number of neighborhoods. This shows that the reversibility provided by *NeNDS* is stronger than data swapping making it a favorable candidate for use in public databases as well as Census records, where "unmodified" techniques are favored. *GT-NeNDS* is a combination of a geometric transformation and *NeNDS*, hence the fraction of the original data that is required for complete reversal is greater than or equal to that required for *NeNDS*, along with the added robustness to the notion of privacy breaches [6].

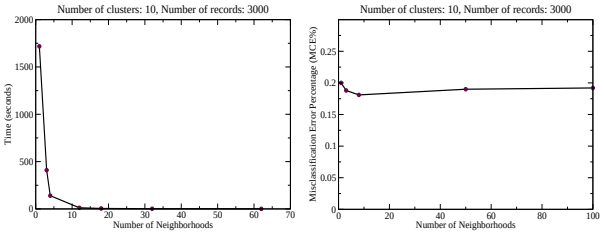
6. Experimental Results

This section provides an experimental analysis of the cluster preserving performance of *NeNDS*. Geometric transformations are inherently cluster preserving and are therefore used as a benchmark for evaluating the performance of *NeNDS* with respect to cluster preservation. The datasets used for performance analysis are obtained from the *UCI Knowledge Discovery Archive* database [20] as well as an open source synthetic data generator [8]. The experiments are performed using the clustering toolbox in Matlab, and cluster analysis is performed by *k-means*, which is a partition-based clustering technique [3]. K-means takes as input the number of clusters k, and selects k centroids in the data space representing the k cluster centers. A quantitative analysis of the cluster preservation is evaluated using the *Misclassification Error (MCE)*, which is a measure of the percentage of legitimate data points that are not well-grouped in the transformed data set. The expression for MCE M_E [19] is given as $M_E = \frac{1}{N} * \sum_{i=1}^{k}(|Cluster_i(X)| - |Cluster_i(X\prime)|)$, where N is the total number of records in the data set $X : X \in D_{k,n}$, k is the number of clusters into which the data are grouped, and $|Cluster_i(Y)|$ is the number of points of Y in the cluster i.

The selection of the number of neighborhoods is an important factor in *NeNDS*. The number of neighborhoods NH is expressed as as $\lfloor\frac{N}{c+1}\rfloor$, where $c + 1$ is the neighborhood size and N is the size of the data set. The effect of the change in number of neighborhoods on the computation time of *NeNDS* as well as the misclassification error after clustering are shown in Figure 6. The data set size is 3000, and the X-axis represents the number of neighborhoods $[1, 32]$. In Figure 6(a) it is observed that the computation time reduces exponentially as the number of neighborhoods increases. This decrease is due to the fact that an increase in number of neighborhoods leads to a smaller size of each neighborhood, which results in an exponential decrease in the time taken for the tree-based search. The graph in Figure 6(b) shows the variation of MCE$\%$ with the number of neighborhoods. The figure shows that the misclassification error is maximum for a single neighborhood and has a minimum value when the number of neighborhoods corresponds to the inherent clustering degree of the data, which is 10 for this data set. The misclassification error increases slightly when the number of neighborhoods is increased beyond the inherent clustering degree. The actual values of the MCE$\%$ are very small, being 0.02% between the extreme values in the figure shown. Hence, the effect of the number of neighborhoods on MCE$\%$ is almost negligible.

The computation time for *NeNDS* is dependent on the number of neighborhoods and yields better performance for smaller neighborhoods. The branch-and-bound search technique used by *NeNDS* is able to reduce the computation time significantly compared to an exponential-time brute-force search. This is evidenced by the fact that, even for a very large neighborhood size (3000 records), *NeNDS* yielded a solution within $1,700$ seconds. Furthermore, Figure 6 shows that the selection of the number of neighborhoods does not significantly affect the accuracy of the result. Therefore, the neighborhood size can be suitably chosen based on the level of privacy required for the database. In order to evaluate the worst case performance of *NeNDS*, the experimental evaluation for the rest of the section is carried out for a single neighborhood of size N.

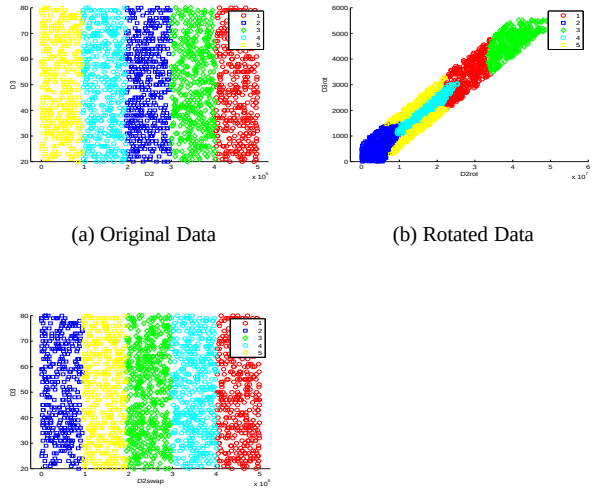
Figure 6 shows the performance of the *NeNDS* transformed data with respect to rotational transformation. The



(a) Computation Time (b) MCE Percentage

Figure 2. Effect of Varying Number of Neighborhoods

data used for these graphs is generated using the synthetic data generator. Here, *D1*, *D2*, *D3* represent the *Salary*, *Commission*, *Age* fields of the synthetic database. The DO results are displayed for grouping parameter values of 2 (default), 5, and 15. The output clustering parameter C_{qu} in this case is the same as the inherent clustering of the original data. The angle of rotation between the attributes $D1, D2$ is 89.9 and between $D1, D3$ is 35.4 degrees. The database is inherently grouped into 5 clusters.



(a) Original Data (b) Rotated Data

(c) NeNDS transformed Data

Figure 3. Comparison of Cluster Preservation

In this figure, a comparison of the clustering nature with respect to the attributes $D2, D3$ are shown. As the number of clusters are small, the neighborhoods remain intact, and the error in clustering is very small. It can be observed that the rotational transformation has changed the shaped of the clusters slightly, but cluster strengths remain the same. The misclassification error would be minimal across the entire database only if all the attributes on which clustering is performed are rotated by the same angle. This, however, would weaken the privacy preservation capability of the transformation. *NeNDS*, on the other hand, is noted to perform consistently in all cases. A detailed experimental evaluation of the performance of *NeNDS* is provided in [14].

Table 3 shows a summary of the misclassification error for the different DO techniques. Two sets of experiments are performed for Random Data Perturbation (RDP), denoted by RDP_low and RDP_high. The values for the noise vector $(mean, var)$ for RDP_low are $(0.0, 1)$ and for RDP_high are $(0.0, 100)$. The angle of rotation for rotation based Geometric Transformation is 89.4 degrees. The value of c for *NeNDS* is kept as $N - 1$ in order to compare the worst case performance of the algorithm. The size of the database used for comparison is $N = 10,000$ and the inherent clustering factor $C_{in} = 10$. The error percentages resulting from k-means is used in the table. The table provides a comparison of MCE as a percentage.

Obf. — Clu.	RDP low	RDP high	Rot. const	Rot. var	NeNDS c=N-1	GT-NeNDS c=N-1
2	0.0	10.1	0.0	0.0	0.0	0.0
3	0.03	25.02	0.05	0.10	0.08	0.11
5	0.10	36.1	0.08	0.17	0.11	0.13
10	0.21	40.5	0.18	0.24	0.20	0.22
20	0.25	40.5	0.40	0.45	1.60	2.18

Table 3. Comparison of MCE %

It is observed that RDP_low yields a very low value of MCE for all cases. This is because the amount of noise added is extremely small. RDP_high performs poorly for all cluster sizes, whereas the other obfuscation techniques are comparable. Although *Rotation* provides a smaller MCE percentage, its vulnerability to reverse engineering makes it unusable for DO of sensitive data. The two columns for *Rotation* techniques show the performance of the algorithm for a constant rotational angle over the entire database, and for different angles selected for each transformation. The data obtained for rotational transformation assumes a $2-D$ rotation. The performance of *NeNDS*, *GT-NeNDS* are observed to be almost as good as the rotational transformation, and their robust privacy preservation capability makes them more suitable candidates for data protection. The performance of the obfuscation techniques degrade if the number of clusters required is chosen as a number much larger than the inherent clustering of the data as can be noted in the case

where the number of clusters is 20. This is twice the value of C. The loss of information in this case is a necessary condition for privacy preservation in order to prevent individual records from being exposed. The results indicate that *NeNDS* and *GT-NeNDS* yield cluster-preserving obfuscated data that are difficult to reverse engineer.

7. Conclusion

Technique	Disp.	Reversibility	Clustering
RDP_low	Very Low	Difficult	Good
RDP_high	High	Difficult	Poor
Data Swapping	Low	Partial $\frac{1}{2}$	Good
NeNDS	Low	Partial $\frac{c}{c+1}$	Good
Geometric	High	Easy	Good
GT-NeNDS	High	Difficult	Good

Table 4. Performance of DO Techniques

The main contributions of this paper are: (1) the proposal of a robust DO technique for clustered data, (2) the definition of a standard for the classification of DO techniques, and (3) the demonstration of the weak privacy provided by existing obfuscation techniques such as linear transformations and data swapping. Table 4 provides a comparison of *NeNDS*, *GT-NeNDS* with existing DO techniques with respect to three parameters: displacement, reversibility, and cluster preservation. The first two parameters indicate the strength of privacy provided by the DO technique, while the third parameter is an indicator of the usability of the DO. Displacement is the average value of MCE (MCE_{Avg}). A robust DO technique is one with *High* displacement, that is *Difficult* to reverse engineer, and that has *Good* cluster preservation. Random Data perturbation is difficult to reverse engineer, but the other two parameters are dependent on the amount of noise added. A large offset provides more displacement, and better privacy, but results in poor cluster capability. On the other hand, a small offset preserves clustering, but results in data with very small displacement, thereby making them vulnerable. Data swapping and *NeNDS* provide small displacements dependent on the nature of the dataset, but are cluster preserving. *NeNDS* is more difficult to reverse than Data Swapping, which makes it a more robust technique. Geometric transformations displace the data substantially and also preserve clustering, but are extremely easy to reverse, which makes them unsuitable for sensitive databases. *GT-NeNDS* provides cluster preservation, high displacement, and is also difficult to reverse, thereby proving to be a robust DO approach for the privacy preservation of clustered data.

References

[1] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. "Order Preserving Encryption for Numeric Data". In *Proc. of Special Interest Group on Management of Data*, pages 563–574, Paris, France, June 2004. ACM Press.

[2] R. Agrawal and S. Ramakrishnan. "Privacy-Preserving Data Mining". In *ACM Special Interest Group on Management of Data*, pages 439–450, 2000.

[3] http://www-2.cs.cmu.edu/awm/tutorials/kmeans.html.

[4] D. Bakken, R. Parameswaran, and D. Blough. "Data Obfuscation: Anonymity and Desensitization of Usable Data Sets". *IEEE Security and Privacy*, 2(6):34–41, Nov-Dec 2004.

[5] D. Denning and M. Schwartz. "The Tracker A Threat to Statistical Database Security". In *ACM Transactions on Database Systems*, volume 4, pages 76–96, 1979.

[6] A. Evfimievski, J. Gehrke, and R. Srikant. "Limiting Privacy Breaches in Privacy Preserving Data Mining". In *Principles of Database Systems*, San Diego, CA, June 2003.

[7] R. Gonzalez and R. Woods. *"Digital Image Processing"*. Addison-Wesley Publishing Company, 1992.

[8] http://www.almaden.ibm.com/software/quest/resources/datasets/syndata.html.

[9] W. Klosgen. "Anonimization Techniquesfor Knowledge Discovery in Databases". In *Proc. of the First International Conference on Knowledge and Discovery in Data Mining*, pages 186–191, Montreal, Canada, Aug 1995.

[10] R. Moore. "Controlled Data-swapping Techniques for Masking Public Use Microdata Sets". In *SRD Report RR 96-04, U.S. Bureau of the Census*, 1996.

[11] http://mathworld.wolfram.com/Affine.html.

[12] S. Oliveira and O. Zaane. "Privacy Preserving Clustering by Data Transformation". In *Proc. of the 18th Brazilian Symposium on Databases*, pages 304–318, Manaus, Brazil, Oct 2003.

[13] S. Oliveira and O. Zaane. "Achieving Privacy Preservation When Sharing Data for Clustering". In *Workshop on Secure Data Management in conjunction with VLDB2004*, Toronto, Canada, Aug 2004. Springer Verlag LNCS 3178.

[14] R. Parameswaran and D. Blough. "An Investigation of the Cluster Preservation Property of Nends". Technical report, Georgia Institute of Technology, 2005.

[15] S. P. Reiss. "Practical Data-swapping The First Steps". In *ACM Transactions on Database Systems*, volume 9, pages 20–37, Mar 1984.

[16] R. Rivest, L. Adleman, and M. Dertouzas. "On Data Banks and Privacy Homomorphisms". In R. A. D. et al, editor, *Foundations of Secure Computations*, pages 169–179. Academic Press, 1978.

[17] P. Samarati. "Protecting Respondent's Privacy in Microdata Release". *IEEE Transactions on Knowledge and Databases*, 13(6), 2001.

[18] W. Stallings. *"Network Security Essentials"*. Prentice Hall, 2000.

[19] G. Toussaint. "Bibliography on Estimation of Misclassification". *IEEE Transactions on Information Theory*, 20(4):472–479, July 1974.

[20] http://kdd.ics.uci.edu/.