BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **17**, No 2
Sofia • 2017 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.1515/cait-2017-0015

Partition Based Perturbation for Privacy Preserving Distributed Data Mining

M. Antony Sheela, K. Vijayalakshmi

Computer Science & Engineering, Arunachala College of Engineering, Nagercoil, India
Computer Science & Engineering, Ramco Institute of Technology, Rajapalayam, India
E-mails: sheelagpsathish@yahoo.co.in vijaya@mepcoeng.ac.in

Abstract: *Data mining on vertically or horizontally partitioned dataset has the overhead of protecting the private data. Perturbation is a technique that protects the revealing of data. This paper proposes a perturbation and anonymization technique that is performed on the vertically partitioned data. A third-party coordinator is used to partition the data recursively in various parties. The parties perturb the data by finding the mean, when the specified threshold level is reached. The perturbation maintains the statistical relationship among attributes.*

Keywords: *Vertically partitioned data, perturbation, recursive partitioning, data mining, anonymization.*

1. Introduction

Data mining technology extracts identifying patterns and trends from huge quantities of data. Recent advances in digital and networking technologies enabled the collection, management and sharing of large amounts of distributed data. Example of such large distributed data repository is found in organizations, industries, government, hospitals, military, etc. In most of the situations the data is distributed horizontally, vertically or both among different parties. The basic data mining algorithms on centralized data needs gathering all data into a central site and then executing it on the data. However, due to legal restrictions among parties, they do not want to reveal their data to other parties during the data mining process. The task of executing data mining algorithms over distributed data sources without revealing any private information is often referred to as privacy preserving data mining [1, 2].

This paper addresses the problem of publishing data distributed vertically over from multiple parties by perturbing individual party's data and then gathering in a central site to publish. Microaggregation is a special clustering problem where it aims to gather elements into groups of at least a threshold value k in such a way that groups are as homogeneous as possible. When considering microaggregation for information systems, elements are database instances. Microaggregation implements privacy in

statistical data sources by gathering a set of d-dimensional elements into groups of elements which are almost homogeneous that can be masked. The aim of this paper is to perturb the vertically partitioned dataset in each party by perturbing the grouped homogeneous data and then publishing it as the single perturbed dataset to perform data mining classifications.

Perturbation on vertically partitioned data – An overview. The proposed method makes use of a third-party coordinator and the participants. The coordinator initiates and controls the perturbation of data in individual parties. It finds the maximum variance max(v_1, …, v_n), where v_i is the variance of i-th party among all the parties. Based on the ids send by the coordinator the parties' partitions their local dataset and then perturbs their corresponding data. The parties' then send their perturbed data for publishing. The framework of the proposed algorithm is given in Fig. 1.

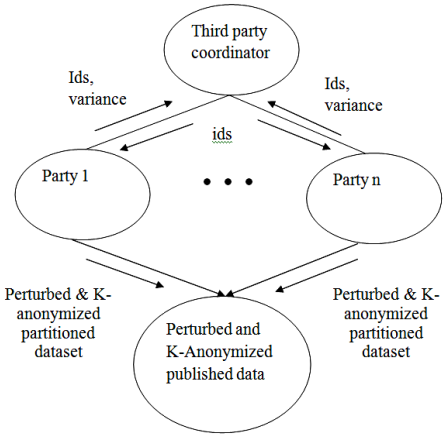

Fig. 1. Framework of perturbation

The remainder of this paper is organized as follows. Section 2 describes the related work in privacy preserving data mining mainly using perturbation technique. Section 3 discusses the perturbation based privacy preserving data mining over vertically partitioned dataset. Section 4 discusses the disclosure measure of the centralized database and that of the distributed database. Section 5 finally concludes the paper.

2. Related works

Preserving the sensitive or private data is progressively becoming a more important issue in many data mining applications. The ultimate aim of a privacy preserving data mining system is to extract information from the data source without revealing the sensitive data. Privacy preservation can represent protecting individual values, protecting sources, protecting record linkage, etc. Many research works are carried out to preserve privacy. Some of the important techniques are data perturbation, cryptography and anonymization.

2.1. Data publishing

Privacy-Preserving Data Publishing (PPDP) is a task to develop methods and tools for publishing data in an antagonistic environment, so that the published data remains practically useful while individual privacy is preserved [3]. For example, a hospital collects data from patients and publishes the patient records to an external medical centre. For example, a hospital (data publisher) gives the patients (owner) records to a medical centre (data recipient) [4]. The medical centre could then perform data mining operations on the data.

2.2. Data perturbation

A popular disclosure protection method is data perturbation [5, 6] which alters individual data in a way such that the summary statistics remain approximately the same. Many works are carried out in the area of distorting data, without compromising the statistical information like sum, average, maximum, count, etc as otherwise they may not yield the desirable mining result [7, 8]. proposed a value distortion technique to protect the privacy by adding random noise from a Gaussian distribution. The decision tree models produced results with better accuracy. As an enhancement of this work Expectation-Maximization-based (EM) algorithm was developed for a better reconstruction of the distribution [9]. In addition of noise the noise added is independent of the scale of X. Multiplicative noise method was proposed to overcome this drawback [10]. Two basic forms of multiplicative noise are multiplying each element of the original data by noise with mean one and small variance and the other is to compute the covariance of logarithmic data and generate random noise with a multivariate Gaussian distribution with mean zero and variance equalling a constant times the covariance computed [11]. Traditional additive perturbation and multiplicative perturbation perturb each data element independently, and the relation between attributes or instances may not be preserved. As the enhancement, multiplicative perturbation noise is applied to random projections instead of applying to each data element that also preserves certain statistical properties of the data, e.g., the inner product, the angles, the correlation, and the Euclidean distance [7].

In Statistical Databases (SDCs), when summary statistics are derived using data on very few individuals, releasing summary statistics leads to the disclosure of private data if they are derived on very few individuals [12]. MicroAggregation (MA) is another perturbation technique that groups the sensitive data into small clusters and then perturbs these clusters. Microaggregation is done based on single attribute (univariate) or multidimensional (multivariate) attribute. Univariate microaggregation causes bias in the variance of the confidential attribute, as well as in the relationships between attributes. Multivariate microaggregation maintains better the relationships between attributes but needs higher computational time complexity which could be inefficient for large data sets. An improved microaggregation that maintains the mean of attributes was developed [12]. It partitions the data recursively and make into small groups and computes the average of each group. The mean of the attribute remains the same and the relationships

between attributes are expected to be reasonably preserved. We have developed a perturbation over vertically partitioned data.

2.3. Secure multiparty computation and anonymization

The Secure Multiparty Computation (SMC) approach evaluates a function on the private inputs of two or more parties' [13, 14]. The computation is carried out in such a way that only the results of the mining activity are revealed and nothing else. A two-phase security is provided for stored data in cloud using encryption by new key distribution method [15]. Various research works are carried out in this field. The circuit evaluation protocol, oblivious transfer and oblivious polynomial evaluation, privacy preserving distributed association rule mining over vertically partitioned data [16], privacy preserving association rule mining over horizontally partitioned data [2, 17] proposed an approach to conduct association rule mining based on randomized response techniques. Circuit, homomorphic encryption, commutative encryption operations needs more computation and communication cost which makes impractical in real world situations. An improved method for privacy preserving decision tree learning over horizontal partitioned data with secret sharing was proposed [2]. A bit matrix is suggested to provide privacy in check-in services [18]. *k*-Anonymity is another technique to preserve privacy by retrieving at least $k-1$ other records that satisfy the query. This technique can be classified into two categories based on their attack principles [4].

The first category considers that a privacy threat when an attacker is able to link the published record or data table or sensitive attribute. The second category aims at achieving the uninformative principle with little additional information beyond the background knowledge. A multidimensional suppression to perform k- anonymity is proposed in the centralized dataset [19]. Their work has two phases. Phase 1 constructs a C4.5 classifier. Phase 2 uses this classifier to construct anonymous datasets. Suppression is carried out mainly for categorical data then option is given for numeric data. The algorithm scales well with large datasets and has no significance on classification accuracy. The drawbacks are pruning of not complying internal nodes leads to over anonymity, unnecessary loss of instances, performs random selection of instances which can be improved by other greedy algorithms. A two-party framework is proposed that generates *k*-anonymous data from two vertically partitioned sources without disclosing data from one site to the other [20]. The vertically partitioned data is anonymized locally and then the identifiers are joined globally. The secure join is performed using homomorphic encryption between two parties. The data of both parties encrypt their own local identifiers and perform secure set intersection. Then every set of one is compared with every set of the other party. If all the set intersection sizes are greater than the threshold for anonymization, then the parties anonymize the data. Otherwise the data are generalized one step further. Datafly is an algorithm used to perform anonymization of dataset in a centralized environment. The authors make the datafly algorithm to be performed on vertically partitioned dataset. The framework suggested by [20] is a two-party anonymization which is difficult to extend for multiple parties. The framework does secure intersection and comparison. Even though the framework is

a general solution, there are issues to address with respect to the construction of *k*-anonymous datasets from distributed sources.

3. Perturbation and anonymization over vertically partitioned data

This section presents a perturbation method using microaggregation over a vertically partitioned dataset that maintains the mean of the attribute and considerably maintains the relation among attributes.

3.1. Problem Definition

Let the number of parties be N, where $N \geq 2$. Each party p has vertically partitioned sensitive dataset. The aim is to recursive partition the data till a threshold level is reached, in collaboration with participating parties that have the vertically partitioned data. The parties finally perturb the data when the number of data entities equals the threshold. The actual data of the individual parties is not disclosed. Finally, the perturbed data is being published.

3.2. Method

The proposed algorithm follows the tree based perturbation for privacy preserving data mining [12] A coordinator function is described that uses the variance and ids collected that satisfies the constraint from different participants. We then describe the initial setup that is made by the various parties before the perturbation takes place. The party function is described which performs the variance computation, sorting the corresponding vertically partitioned dataset, finding ids and perturbation. At last the final setup made by various parties.

3.3. Initial setup by the parties

Let D_{in_i} be the vertically partitioned dataset of the i-th party p_i and n_i be the number of attributes of i-th party. Let a_{in_i} be the attributes of the i-th party. The number of instances in all the parties is assumed to be same and let it be m which is multiple of k a threshold value. The sites include random instances to make m a multiple of k. All parties globally agree and shuffle their rows and then generate pseudo IDs in collaboration with other parties. Then each party shuffle their attributes locally. The distribution of the variance of the perturbed attribute is same as the distribution of the original attributes. So, local shuffling of attributes will not have much effect. Local shuffling of attributes and Global shuffling of rows makes no information gained by the coordinator through the intermediate results.

3.4. Initial setup by the coordinator

The third-party coordinator requests all the N participating parties to send the pseudo ids and the number of attribute corresponding to the party. The pseudo ids are of the form id = 1, 2, …, m. Initialize a flag array with all zeros. The flag array is a sequence of bits and each n_i bit correspond to the number of attributes of the parties 1, …, N, respectively.

The main idea of the proposed algorithm is perturbing the data at their local sites with the help of the third-party coordinator without revealing the party's private data. The proposed algorithm for perturbing data over vertically partitioned dataset is given in Fig. 1.

Algorithm 1

Phase 0: Initial setup by coordinator and parties

Initialization by the coordinator

Step 1. The coordinator gets the number of attributes n_i, the threshold, the number of instances m and the pseudo ids id = 1, 2, …, *m*, from each party.

Step 2. flag = [0, …, 0], where size of the flag = $\sum_{i=1}^{N} n_i$.

Initialization by the parties

Step 1. Each party agree with a value $k \geq 3$, which is a threshold value to perturb the data.

Step 2. Make the number of instances as *m*, where *m* is a multiple of *k* and close to the original number of instances by including pseudo instances.

Generating the pseudo ids

Step 3. Party one generates a random sequence of *m* numbers in the range [1 – *m*] and sends it to other parties.

Step 4. All the parties map this number with their instances and then sort the rows with references to the random sequence. This sorted random number is the pseudo ids.

Phase 1: Recursive partitioning by the coordinator

Step 1. If the corresponding flag bits are not all ones then send the pseudo ids and the portion of the flag corresponding to *i*-th party, where *i* = 1, …, *N*, to compute the maximum variance.

Step 2. Let *temp* is assigned with the number of pseudo ids.

Step 3. Receives the variance v_i and the attribute number from each party.

Step 4. Computes $\max(v_i)$ and finds the *i*-th party.

Step 5. Set the corresponding bit in the flag array to 1.

Step 6. Sends request to *i*-th party to find the two subset ids.

Step 7. Receives the two id subsets from *i*-th party and sends it to all the other parties.

Step 8. Check whether temp equals *k* if so send ids to each party for perturbing. Stop the procedure.

Step 9. Repeat step 1 with the first id subset and the second id subset.

Phase 2: Computation by each party

Variance computation

Step 1. Receives the flag corresponding to the party from the third-party coordinator.

Step 2. If the bit corresponding to the attribute is zero in the flag array then find the variance of that attribute otherwise the variance of that attribute is 0.

Step 3. Find the (variance (a_{in_i})), where n_i is the number of attributes.

Step 4. Send the variance and the attribute number to the coordinator.

Subset computation

Step 1. Sort the rows of the party's dataset with respect to the attribute with maximum variance.

Step 2. Find the midrange value of the attribute and split the ids into two subsets having less than or equal to midrange and greater than midrange.

Step 3. Move the last of first few data instances between subsets to make both as multiples of k.

Step 4. Send the two subsets to the coordinator.

Perturbation computation

Step 1. Locate the ids received from the coordinator. Compute the mean $\overline{x_i} = \sum_{j=1}^{k} x_{ij}$ for each confidential attribute.

Theorem 1. Algorithm 1 privately computes the perturbation of the confidential attributes by each party revealing at most:

1. The pseudo ids of the dataset,
2. The variance of the attributes.

$Proof$: In Phase 0 the coordinator receives only the pseudo ids and not the original ids. The parties collaborate among themselves to generate the pseudo ids. The mapping between pseudo ids and original ids are known to all the parties. In Phase 1 the coordinator receives the variance and the the position of the attribute with maximum variance. But this attribute order is shuffled after compete perturbation by each party. So, no intermediate information is gained by the coordinator. In Phase 2 the parties know the pseudo ids of the two partitions. But the data is perturbed by finding the mean of k confidential attributes. So, the original information is not revealed. Even though the intermediate ids are revealed by the parties the original data in the published dataset is not revealed because the perturbed value is the mean of k values.

Definition 1. Let x be the set of attributes that may be used to identify the specific individual, D be the original dataset and D_k be the instances k-anonymous data generated by the query. $D_k(x)$ satisfies k-anonymity if and only if each instance in it appears at least k times.

The perturbation computation of Phase 2 performs the mean of confidential attributes of k instances. The instances are then replaced by the mean of each attribute. Phase 0, Phase 1 and Phase 2 splits the vertically partitioned dataset recursively into groups of similar data. The original data is replaced by the mean of k homogeneous data. The data can be used by the data mining algorithms.

4. Security against collusion

Each site has its own vertically partitioned data. Collusion of the participants may disclose information about the colluding parties and not the other parties. The statistical information like the midrange, minimum or maximum value of other parties may not be revealed by any participant. The intermediate results disclosed are variance, party with maximum variance and subset of ids. These intermediate results reveal no information about the actual values in the attributes of other parties. Collusion has almost no effect on data values of other parties.

5. Computation and communication

In Phase 0, the setup phase, the coordinator sends request to N parties and receives the information from all the parties and initializes the flag array. The communication cost is $O(2N)$. Each party makes its dataset as multiples of k by including at most $k-1$ rows. Party one sends a random sequence of number to other parties. All parties sort the rows with reference to the pseudo id.

Sorting needs $O(m^2)$ computation in each party. This computation is performed in each party independent of other parties.

In Phase 1, the coordinator makes recursive call with two pseudo id subsets till the threshold k is reached. This requires $O(\log_2 m)$ computation. During each recursion, the coordinator sends request and receives information three times. First to get the maximum variance of each party, second to get two pseudo id subsets from the party with maximum variance among all the parties and third send request to parties to find the mean of the rows for the records with the given pseudo ids. The total communication cost is $O(3\times 2N)$.

The computation cost in each party is for locating the pseudo ids and finding the variance $O(m)$, the party with attribute having maximum variance sorts its rows with reference to the attribute $O(m^2)$, and splits the pseudo ids into two subsets based on the midrange values. When the threshold is reached the parties locate the pseudo ids and perturb their data $O(m)$. For each recursive call by the coordinator the maximum computation time consumed by the parties is $O(m^2)+2\ O(m)$. The total computation cost is approximately $O(\log_2 m)\times O(m^2)$.

6. Experimental analysis

Five datasets from UCI machine repository were used to test the performance of the algorithm. The datasets are namely iris, diabetes, ecoli, biodeg and ionosphere. All the attributes were considered as confidential during the experiment. We have assumed two parties which can easily be extended to arbitrary number of parties.

The threshold value was assumed to be three. The vertically perturbed datasets were perturbed in their local sites with the collaboration of the coordinator. The algorithm implements two different privacy preserving data mining techniques namely perturbation and k-anonymization. The perturbation is done when the

partition has exactly threshold *k* instances. The mean is found for the threshold number of instances which leads to *k*-anonymization and perturbation. As the number or instances to be perturbed is taken as exactly equal to the threshold value, naturally the information loss will be more than the case that the perturbation is done on partition with less than or equal to the threshold value.

The quality of the perturbed algorithm is compared with respect to the disclosure measure, information loss and accuracy when used for decision tree classification. The disclosure risk is estimated using the square Root of Average Standard Deviation (RASD) and information loss is measured using Bias In Standard deviation (BIS) and Bias In Mean (BIM) [12]. RASD is calculated by

$$\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}((x_{ij} - m_{\mathrm{o}j})/\sigma_{\mathrm{o}j} - (y_{ij} - m_{\mathrm{p}j})/\sigma_{\mathrm{p}j})^2},$$

where *m* is the number of instances of the vertically partitioned dataset, *n* is the total number of attributes of all the parties, $m_{\mathrm{o}j}, \sigma_{\mathrm{o}j}$ are the mean and the standard deviation of the particular attribute in original dataset, and $m_{\mathrm{p}j}, \sigma_{\mathrm{p}j}$ are the mean and the standard deviation of the particular attribute in perturbed dataset. The bias in mean and standard deviation are given by $\sum_{i=1}^{m}(m_{\mathrm{p}i} - m_{\mathrm{o}i})/\, m_{oi}$ and by $\sum_{i=1}^{m}(\sigma_{\mathrm{p}i} - \sigma_{\mathrm{o}i})/\, \sigma_{oi}$, respectively. The increase in RASD indicates large deviation which decreases the disclosure risk. Low values of bias decrease the information loss which increases the accuracy of the data mining outputs. The perturbation algorithm should have less disclosure risk and less information risk to give desirable outputs by the data mining algorithms. The minimum threshold used in the algorithm is three which produces a *k*-anonymity output of three records for each query. In general, for a threshold value of k, the queries produce exactly *k*-anonymous outputs. High values of *k* may decrease the utility of perturbed data.

The disclosure measure is higher and the information loss is less than the tree perturbation. The incorrectly classified data percentage is higher than the tree perturbed for diabetes, iris and ecoli. For biodeg dataset the incorrectly classified data percentage is less than the tree perturbs. For data mining result, more or less near to the value of data perturbation using tree based perturbation by Xiao et al. They have proved that the error percentage using their proposed algorithm is better than simple noise addition and multiplication. The one which we have used is a specialization of their algorithm that uses exactly *k* records for perturbation. The mean of the resultant perturbed dataset remains unchanged. The variance of the perturbed dataset is less than the original dataset, but the distribution of the variance is same for both the perturbed data and the original dataset.

Assume a set of *N* records with confidential original values x_i, where $i = 1, \ldots, N$, and perturbed values y_i, where $i = 1, \ldots, N$, respectively. The disclosure risk of the perturbation can be measured by the Root Average Squared Distance (RASD), expressed as

$$\mathrm{RASD} = \sqrt[2]{\frac{1}{N}\sum_{i=1}^{N}(y_i - x_i)^2}.$$

Clearly, a larger RASD value indicates a smaller disclosure risk. The proposed kd-anonymous perturbation method gives a smaller disclosure risk than the kd-tree method. This is shown in Fig. 2. The CPU execution time of the proposed algorithm

is less than compared to the kd-trees algorithm. This is shown in Fig. 3. This is due to the fixing up of threshold value which causes the number of leaf nodes less than that of the existing one. as threshold value increases the number of leaf node decreases and the CPU execution time decreases. Smaller values of BISD mean that the information loss is less. So, smaller BISD values are desirable. More the value of threshold, the more severely the data are perturbed, which means a lower disclosure risk and a higher information loss. Smaller the value of k the information loss is less. Fig. 4 shows the BISD values between four datasets for kd-tree and kd-anonymous methods. The perturbed dataset is then tested in weka for decision tree classification. The incorrectly classified is slightly high when compared to the kd-tree algorithm. But this is acceptable since all the data instances are being perturbed where as in kd-tree algorithm the mean is found for one or two or till the instances less than or equal to threshold. The mean of one instance does not take part in perturbation. So, the incorrectly classified percentage is slightly lesser than the kd-anonymous perturbation. This is shown in Fig. 5.

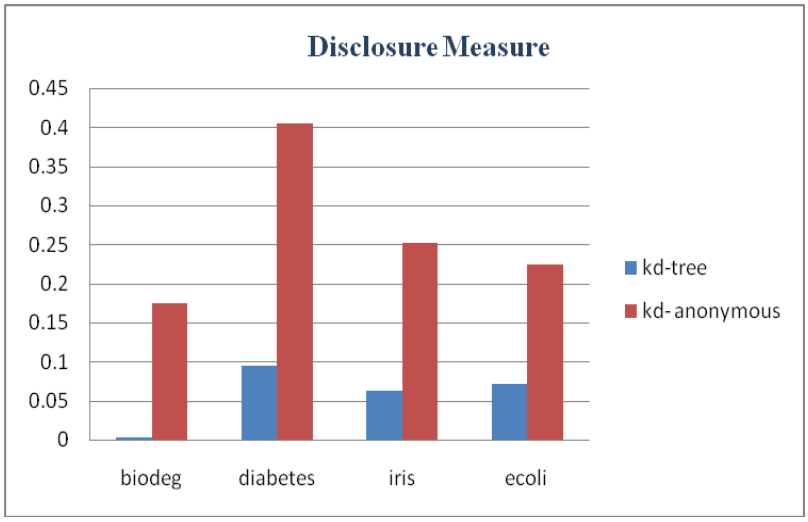

Fig. 2. Comparing disclosure measure between four datasets for kd-tree and kd-anonymous methods

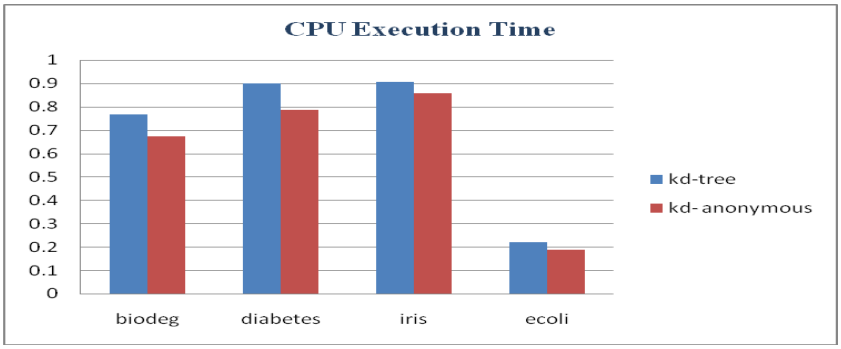

Fig. 3. Comparing execution times between four datasets for kd-tree and kd-anonymous methods

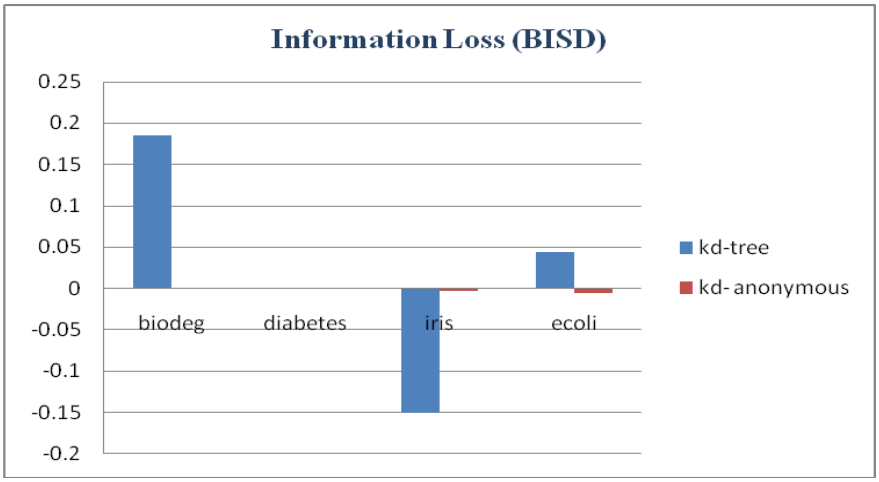

Fig. 4. Comparing information loss (BISD) between four datasets for kd-tree and kd-anonymous methods

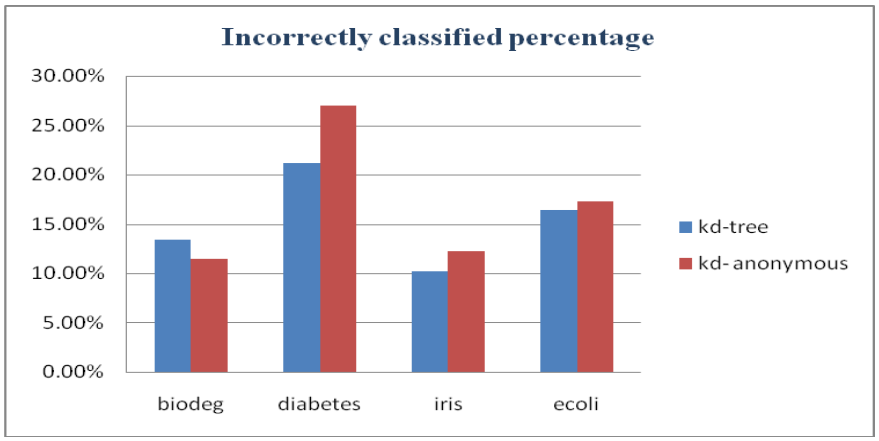

Fig. 5. Comparing percentage of incorrectly classified instances in decision tree between four datasets for kd-tree and kd-anonymous methods

7. Conclusion and further work

A way to perturb the individual data over a vertically partitioned dataset with a third-party coordinator is proposed. The third-party coordinator initiates the perturbation process between participants with vertically partitioned dataset and makes each individual to perturb their data when the threshold value is reached. This perturbed data can be published by each participant to perform data mining classifications as a centralized dataset. The future work that can be done is to find the threshold value of k for which the classification algorithms give accepted results. The data can be analysed to fix the value of k for the particular dataset to increase the correctly classified percentage.

References

1. Sharma, A., V. Ojha. Implementation of Cryptography for Privacy Preserving Data Mining. – International Journal of Database Management Systems, Vol. **2**, 2010, No 3, pp. 57-65.
2. Emekci, F., O. D. Sahin, D. Agrawal, A. El Abbadi. Privacy Preserving Decision Tree Learning over Multiple Parties. – Data & Knowledge Engineering, Vol. **63**, 2007, pp. 348-361.
3. Domingo-Ferrer, J., V. Torra. Ordinal, Continuous and Heterogeneous k-Anonymity Through Microaggregation. – Data Mining and Knowledge Discovery, Vol. **11**, 2005, pp. 195-212.
4. Fung, B. C. M., K. Wang, R. Chen, P. S. Yu. Privacy-Preserving Data Publishing: A Survey of Recent Developments. – ACM Computing Surveys, Vol. **42**, 2010, No 4, pp. 14-53.
5. Friedman, A., A. Schuster. Data Mining with Differential Privacy. – In: KDD'10, July 2010, Washington, DC, USA, pp. 25-28.
6. Patil, S. P., S. V. Badhe. Geometric Approach for Induction of Oblique Decision Tree. – International Journal of Computer Science and Information Technologies, Vol. **5**, 2015, No 1, pp. 197-201.
7. Kargupta, H., K. Liu, J. Ryan. Random Projection-Based Multiplicative Data Perturbation for Privacy Preserving Distributed Data Mining. – IEEE Transactions on Knowledge and Data Engineering, Vol. **18**, 2006, No 1, pp. 92-106.
8. Agrawal, R., R. Srikant. Privacy Preserving Data Mining. – In: Proc. of ACM SIGMOD, ACM, New York, 2000, pp. 439-450.
9. Charu, A., A. Dakshi. On the Design and Quantification of Privacy Preserving Data Mining Algorithms. – In: ACM PODS'01, Santa Barbara, California, USA, 2001.
10. Adam, N. R., J. C. Wortmann. Security-Control Methods for Statistical Databases: A Comparative Study. – ACM Computing Surveys, Vol. **21**, 1989, No 4, pp. 515-556.
11. Kim, J. J., W. E. Winkler. Multiplicative Noise for Masking Continuous Data. – In: Statistical Research Division, U.S. Bureau of the Census, Washington, DC, Tech. Rep. Statistics, 2003-01.
12. Li, X.-B., S. Sarkar. A Tree-Based Data Perturbation Approach for Privacy-Preserving Data Mining. – IEEE Transactions on Knowledge and Data Engineering, Vol. **18**, 2006, No 9, pp. 1278-1283.
13. Kiran, P., K. S. Sathish, N. P. Kavya. A Novel Framework Using Elliptic Curve Cryptography for Extremely Secure Transmission in Distributed Privacy Preserving Data Mining. – International Journal in Advanced Computing, Vol. **3**, 2012, No 2, pp. 85-92.
14. Li, L., M. Kantarcioglu, B. Thuraisingham. Privacy Preserving Decision Tree Mining from Perturbed Data. – In: Proc. of 42nd Hawaii International Conference on System Sciences, 2009, pp. 1-10.
15. Govinda, K., E. Sathiyamoorthy. Privacy Preservation of a Group and Secure Data Storage in Cloud Environment. – Cybernetics and Information Technologies, Vol. **15**, 2015, No 1, pp 46-54.
16. Vaidya, J., C. Clifton, M. Kantarcioglu, S. A. Patterson. Privacy-Preserving Decision Trees over Vertically Partitioned Data. – ACM Transactions on Knowledge Discovery from Data, Vol. **2**, 2008, No 3, pp. 14-27.
17. Kantarcioglu, M., C. Clifton. Privacy-Preserving Distributed Mining of Association Rules on Horizontally Partitioned Data. – IEEE Transactions on Knowledge and Data Engineering, Vol. **16**, 2004, No 9, pp. 1026-1037.
18. Chen, W. Privacy-Preserving of Check-in Services in MSNS Based on a Bit Matrix. – Cybernetics and Information Technologies, Vol. **15**, 2015, No 2, pp. 111-118.
19. Kisilevich, S., L. Rokach, Y. Elovici, B. Shapira. Efficient Multidimensional Suppression for k-Anonymity. – IEEE Transactions on Knowledge and Data Engineering, Vol. **22**, 2010, No 3, pp. 334-347.
20. Wei, J., C. Clifton. A Secure Distributed Framework for Achieving k-Anonymity. – The VLDB Journal, Vol. **15**, 2006, pp. 316-333.