International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 4 (2015) pp. 11085-11102

http://www.ripublication.com

Distributed Differential Privacy Preserving Mechanism on Real Time Datasets

Y.A.Siva Prasad and Dr.G.Rama Krishna

Research Scholar, Dept of CSE KLUniversity, Andhra Pradesh
sivaprasady@gmail.com
Professor, Dept of CSE KLUniversity , Andhra Pradesh
Gavirneni.ramakrishna@gmail.com

Abstract:

Data mining is the technique of discovering patterns among data to analyze patterns or decision making predictions. Protecting private data is an important concern for various data mining tasks like association rule mining, classification, clustering etc. Privacy preserving data mining can be used to disclose sensitive information from the distributed datasets. Various traditional algorithms have been proposed to hide sensitive data using statistical and cryptographic mechanisms. Most of the existing models ensure database privacy in which data miner is allowed to pose queries to the database.

In this proposed work, we extended the privacy model in distributed databases using multi-objective distributed decision tree algorithm. Proposed algorithm uses distributed entropy measure for selecting relevant attributes from the databases. Multi-Objective mechanism provides sensitiveness within the attributes as well as on the decision classes. Multi-Objective process introduces lower and upper bound mechanism for each node in the decision tree construction to preserve the data values in the decision rules. Experimental result performs well against different distributed datasets in terms of time and accuracy.

Keywords – Outlier, Data Mining, Patterns.

I. INTRODUCTION

Data mining is the extracting knowledge or pattern from large volumes of distributed data. It's usually employed by researchers for science and business plan. Data collected from information providers are significant for pattern recognition and selection process. However attacks are attempted to steal these sample datasets and personal information might be leaked by reviewing those stolen datasets. Therefore

privacy preserving data mining are developed to resize sensitive datasets into sanitized version through which private or sensitive info is hidden from unauthorized retrievers.

Privacy preservation via dataset complementation is basically a data perturbed approach that substitutes each original dataset through use of an entire unreal dataset. Privacy preservation is matched to sanitize the samples in advance of their release to 3rd parties so that you can mitigate the specter of their inadvertent disclosure or theft. In comparison to other sanitization methods, our approach fails to prohibit the accuracy hard drive data mining results. The alternative tree can possibly be built direct from the sanitized data sets, in a way that the originals are not required to get reconstructed. Moreover, this procedure can easily be applied anytime through the data collection process ensuring that privacy protection is able to be in effect no matter if samples remain to be being collected.

Several options for managing missing values have been proposed. One general technique to handling missing values will be to create data mining algorithms that "internally" handle missing values but still produce good results. For instance, the CART decision-tree learning algorithm internally handles missing values essentially using an implicit kind of imputation in accordance to regression. However, in this particular paper, we follow the most relevant privacy approach, where pre-processing is performed first as well as having the resulting data is well suited for use utilizing a number of data mining algorithms. This happens to be particularly needed among the setting of privacy-preserving data mining because, as of yet, the existing privacy-preserving data mining algorithms never render any special internal handling of missing data. A stand-alone strategy to privacy-preserving imputation can therefore be operated in conjunction with the entire privacy-preserving data mining algorithm for a similar distributed setting.

Privacy-preserving data mining make it possible for traditional data mining algorithms to preserve data protection in the mining process. Much works have been done to explore privacy preserving on vertically or horizontally partitioned data involving multiple parties to make sure that not a single one party keeps the overall data. Figures 1 describes vertically and horizontally partitioned data. For vertically partitioned data, two parties if not more keep the different multitude of attributes for similar multitude of objects. For horizontally partitioned data, two parties if not more hold different objects for a similar range of attributes. It indicates each object among the virtual database is fully managed by one party. In arbitrarily partitioned data, different disjoint portions are held by different parties.

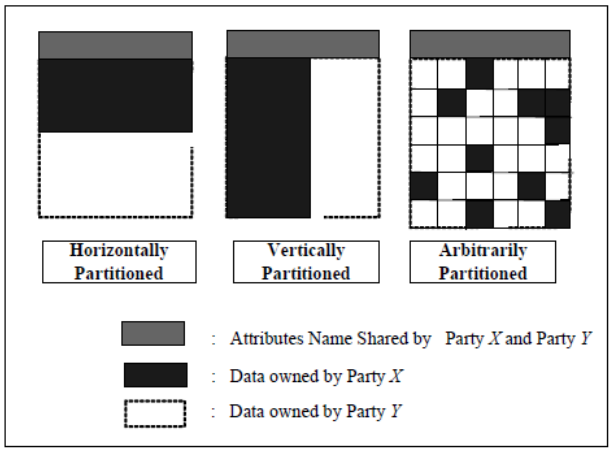
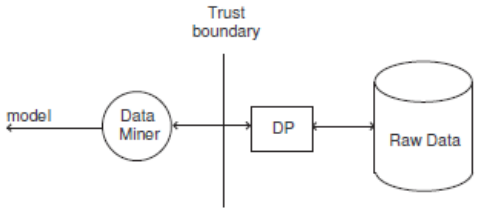


Fig 1: Data partition based Privacy-preserving data mining

Need of Privacy Preserving:

Data mining presents many opportunities for enhanced services and products in diverse areas such as healthcare, banking, business planning, online search, and so on. However, its promise is hindered by concerns regarding the privacy of the individuals whose data are being mined. Therefore, there is great value in data mining solutions that provide reliable privacy guarantees without significantly compromising accuracy.



With the emergence of such unpredictable amount of information stored at different physical location, distributed data mining has become one of the key enablers of large scale knowledge extraction. However, information is almost always collected under certain privacy considerations. Organizations might not want to share with each other the contents of their data, sometimes even the statistic either due to legal or competition constraints.

II. LITERATURE SURVEY

Related Work:

Fang et al proposed algorithms a Privacy-preserving distributed decision-tree mining algorithm, which happens to be in accordance to concept of Privacy-preserving decision tree and passing control from site to site [2]. The disadvantage of the strategy is that each party has got the class attribute. Missing attribute values do not happen to be handled by these methods.

Vaidya et al proposed algorithms on building decision tree, however, the tree on every party doesnot contain any information that remain in other party, resulting class can easily be altered by the malicious party[3].

Lindell *et al* has proposed a privacy protection technique to generate a decision tree using ID3 among two parties over horizontally partitioned data using SMC[4].

Weber-Jahnke [5-6] introduce a brand new perturbation and randomization based approach that serves centralized sample data sets utilized for decision tree data mining. They introduced a fresh privacy preserving approach via data set complementation. This procedure converts the unique sample data sets into your team of unreal data sets. The alternative tree can easily be built straight from the sanitized data sets, there isnt any need be reconstructing the main dataset. This system requires extra storage for storing perturbed and complement of sample data set. So optimizing the storage size of the unrealized samples ought to be explored.

Using ID3 algorithm over multiple parties- a scalable secured distributed ID3 for building a tree recursively. The contributing parties cannot here are the secret value of another party even if they should exchange their shares other people. This system assumes that each one participant does computation honestly in distribution and intermediate phase. Could possibly be computationally intensive [7].

Proposed Architecture:

III. LITERATURE SURVEY

In order to overcome the deficiencies of distance-based methods, Breunig et al. [1] proposed that each data point of the given data set should really be assigned a degree of outlier. With their view, for example other recent studies, a data point's measure of anomaly should be measured relative to its neighbors; hence they refer to it just like the "local outlier factor\" of the data point. Tang et al. [2-4] argued that any outlier doesn't always have to remain of lower density and lower density is't a necessary condition to remain an outlier. They modified LOF to search for the "connectivity-based outlier factor\" (COF) which they argued is so much more effective each time a cluster and a neighboring outlier have similar neighborhood densities. Local density is widely measured in terms of k nearest neighbors; LOF and COF both exploit properties associated with k nearest neighbors of causing given object in the data set. However, it is possible that any outlier lies in a location between objects given by a sparse as well as a denser cluster. To account for such possibilities, Jin et al. [5-6] proposed another modification, called INFLO, that's in accordance to a symmetric neighborhood relationship. That is, their proposed modification considers neighbors and 'reverse neighbors' associated with a data point when

estimating its density distribution. Tao and Pi [2] have proposed a density-based clustering and outlier detection (DBCOD) algorithm, which also is a member of the density-based algorithms. Density-based algorithms assume that all neighborhoods associated with a data point have similar density. If some neighbors of one's point can be found a single cluster, plus the other neighbors near each other another cluster and to discover the two clusters have different densities, then

comparing the density of a given data point with all of that neighbors may lead to a wrong conclusion and the recognition of real outliers may fail.

Anomalous series detection and contextual abnormal subsequence detection are both viewed as applicable for time series data set. In the current research, only real-valued time series are actually, categorical-valued time series are out of the coverage of this investigation. Anomalous series detection only places focus on identifying anomalous series whereas the contextual abnormal subsequence problem requires that we all detect abnormal subsequence in the context of a single series, and this requires the comparison between subsequences and the majority of this game's series. The main gap between these two techniques is: the first one works to find out which series is anomalous as the latter one wants to know when abnormal behaviors occur.

Some of the problems in Time series data are: Historical information associated with a series has to be examined, but how to summarize the useful historical details are a difficult problem. The behavior of outliers is different for different applications, and it makes detecting abnormal behavior a hard activity. Within a single application domain, the outlier is likewise changing with time, so it requires any effective algorithms or techniques to be very adaptive and malleable to contend with dynamic detection. The algorithm ought to be aware of the dynamically changing outliers[7].

Knorr et al. [3] proposed the DB(pt, dt) Outlier detection scheme, wherein an object obj is said to be to get an outlier if at the very least fraction pt of the total objects have greater than dt distance to obj. They defined several techniques to find such objects. For instance the index based approach computes distance range using spatial index structure and excludes an object if its dt-neighbourhood contains greater than 1−pt fraction of total objects. They proposed nested loop algorithm to avoid the cost of building an index. They additionally proposed growing a grid so that any two objects beginning with the same grid cell have a distance of the most dt to one another. In this way objects ought to be in relation to those from neighboring cells to examine if they're outliers.

In statistics, regression analysis is made use of to approximate the relationship between attributes.

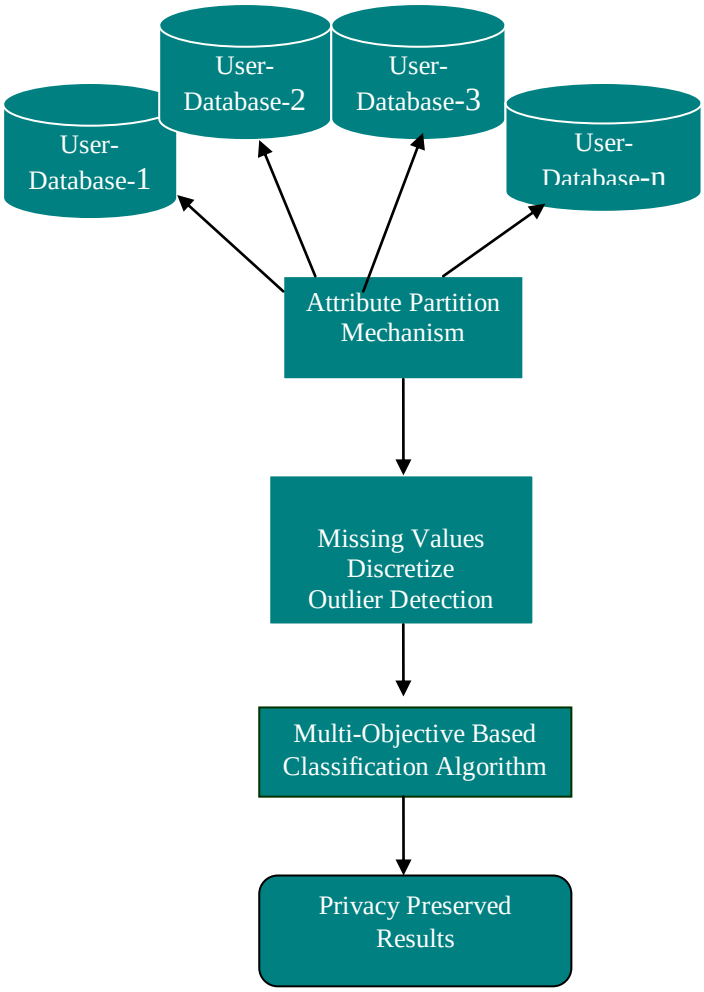
Linear regression and logistic regression are two common models. An outlier in regression analysis is undoubtedly an observation whose value is removed from the prediction. To detect such outliers, the residuals of the observations are computed dependent on a trained model.

Traditional clustering techniques effort to segment data by grouping related attributes in uniquely defined clusters. Each data point within the sample space is granted to just 1 cluster. K-means algorithm and also its different variations will be the most well-known and commonly used partitioning methods. The value ‘k’ stands for the number of cluster seeds initially provided for the algorithm. This algorithm takes the input parameter ‘k’ and partitions a set of m objects into k clusters [4]. The procedure work by computing the gap between an information point and to discover the cluster center to enhance one item into one of this very clusters ensuring that intra-cluster similarity is high but inter-cluster similarity is low. A common method to

obtain the distance will be to calculate to sum of the squared difference as shown below and it is known as the Euclidian distance.

$$d_k = \sum_n \left\| X_j^k - Ci_j \right\|^2$$

where, *dk* : is the distance of the kth data point from Cj



Databases:

In this part, different datasets are taken as input to preserve privacy against decision rules.

Let $D_1, D_2, D_3 .... D_n$ be the distributed user data. Each dataset has different combinations of numerical, nominal, interval or ratio scaled attributes. Each database has one trained data along with or without test data. Each dataset is represented as D= $\{\tau_1 \tau_2 \tau_3 ... \tau_n\}$ with attributes A=$\{at_1 at_2 .... at_n\}$.

Attribute Partition and Transformation Algorithm:

Step 1: Read dataset D.

Step 2: // Numerical and Non-Numerical Attributes Partition

- For each attribute a(i) in D
- Do
- if a(i) is numeric
- then
- index=i
- end if
- done.
- Construct two datasets one is numeric and other nonnumerical attributes $D^{'}$ $D^{''}$.

Step 3: Missing_Values($D^{'}$)

MaxProb:=0;

Class cls=null;

//Checking each attribute missing values.

- For each attribute A(i) where i=1 to n
- Do
- For each attribute value V(j) where j=1 to n
- Do
- If(V(j)==null)
- Then
- addList(i,V(j),class(V(j)); // missing attribute index, missing value and class value of the missing value.
- end if
- done
- done

// finding max prob of each attribute to each class

- For each attribute A(i) where i=1 to n
- Do
- for each class C(k) where k=1..m classes
- do
- prob=Find max probability of each attribute A(i) corresponding to each class.
- MaxProb(i,k)=prob;
- Done
- Done

//

- For each attribute A(i) where i=1 to n
- Do
- For j=0 to addList length

- Do
- Cls=Get missing class value addList(i,V(j));
- for each class C(k) where k=1..m classes
- do
- If(Cls==C(k))
- then
- V(j)=MaxProb(i,k);
- End if
- Done
- Done
- Done

Step 4: Missing_Values(D'')
Replace the missing values using Naïve Bayesian Probability of the most probable value to each attribute.

Step 5:
//Applying numerical dataset for discretization D'

- For each instance I in dataset D'
- Do
- Tokens[]=Tokenize(I,",");
- For each token value tokval in Tokens[]
- Do
- Check tokval is numeric or not
- If tokval is numeric
- Then
- List(tokval,attributeindex)
- End if
- Done

//Discretize the numerical attributes using global and min value range.

Step 6: Discretize(List)
Step 7: Outlier_removal(List)
Done

Discretize(List)

- Sort A(i) in ascending order
- For each attribute A(i) where i=1 to n
- Do
- For each attribute value V(j) where j=1 to n
- Do

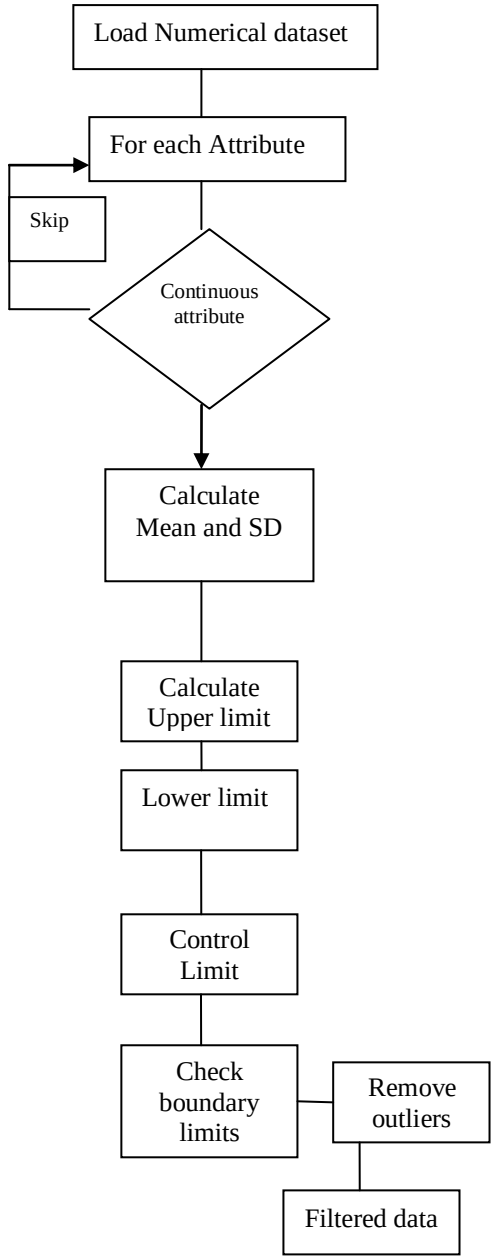
- Attribute values V(j) is partitioned corresponding to m classes as $P_1^i, P_2^i, ...P_m^i$
- Done
- Done
- For each Class C(k) where k=1 to m
- Do
- For each partition P_k
- Do
- Find the conditional probability P(C(k)/P) on each cut point and select the cut point with maximum probability value
- Done
- Done

Outlier_removal(List)

- For each attribute in List
- Do
- Statistical Control limits to eliminate outliers
- Lower Control limit: $\mu_X - \lambda\sigma_X$
- Upper Control Limit: $\mu_X + \lambda\sigma_X$
  - Control Limit: μ_X
- Done

Procedure:

Input: Continuous dataset
Output: Dataset without anomalies.
Procedure:
Step 1:Load dataset with continuous attributes.
Step 2: Check each attribute in the dataset as real attribute or not.
Step 3: Calculate mean and standard deviation of each attribute.
Step 4: Calculate upper control limit of each attribute (2)
Step 5: Calculate control limit of each attribute (3).
Step 6: Calculate lower control limit of each attribute (1)
Step 7: Check whether each object in the dataset falls within three categories i.e lower, upper or control limits.
Step 8: If the object is out of bound then it is removed from the dataset instances.
Step 9: This process is repeated until all data points are completed.
Step 10:Finally dataset without outliers are stored in file.



Multi-Objective Based Classification:

Let $F_1 : \square^n -> \square$ and $F_2 : \square^n -> \square$ represent the distributed tree node data privacy to node cost function respectively, where $\varepsilon \in \square^n$ represents the n tuple input vector with n attributes. F(ε) denotes the loss of decision rules due to privacy preserving. Therefore, optimization function of the distributed privacy preserving model to minimize the loss of data due to privacy can be represented as:

Min $F = w' F(\varepsilon) = w_1' F_1(\varepsilon) + w_2' F_2(\varepsilon) + \sigma_\varepsilon$

Subject to $\varepsilon_i^{(p)} \le \varepsilon_i \le \varepsilon_i^{(q)}, \forall i = 1...n$

$w_1^{'} + w_2^{'} = 1$

$w_1^{'} > 0$

$w_2^{'} > 0$

$\sigma_\varepsilon > 0$

Where $w_1^{'}$ and $w_2^{'}$ are the reliability factors of the privacy data and the cost function F_2. Optimality of the data privacy depends on the reliability factors. For each classified node in the decision tree, optimal Laplace transformation functions $F_1^{'}$ and $F_2^{'}$ are selected based on the reliability factors of the node.

Distributed Privacy Preserved C4.5 is an improved version of decision trees over C4.5 from the training data, using the concept of information entropy.The training data is a set d1, d2,d3 .... dn data objects in the dataset D. Each di = xl,x2,... is a sample values where xl,x2,... represents features or attributes of the sample. The training data associated with a vector C = cl,c2,... where cl,c2,...cn represents the class to which each sample belongs to dataset. Every node of the decision tree, chooses one attribute of the data the most efficiently splits its range of samples into subsets in a single class. The attribute with the highest calculated information gain is selected to get the decision attribute. In this proposed approach the most relevant split attribute is given input to sensitivity to preserve privacy. Each most relevant split attribute is checked against multi-objective model to preserve the privacy.

Most relevant attribute is calculated by using following distributed entropy measure:

DEM(DistributedEntropyMeasure) (D)= $-D_i \sum_{i=1}^{m} l \operatorname{og} \sqrt[\alpha]{D_i}$,m different classes

DEM(DistributedEntropyMeasure) (D)= $-D_i \sum_{i=1}^{m} l \log \sqrt[\alpha]{D_i}$

$= -D_1 \log \sqrt[\alpha]{D_1} + D_2 \log \sqrt[\alpha]{D_2} .....D_n \log \sqrt[\alpha]{D_n}$

Where D_1 indicates set of samples which belongs to target class C_1, D_2 indicates set of samples which belongs to target class C_2 and so on. α is the sensitive factor of the attributes.

Distributed Entropy to each attribute is calculated using

$$DistributedInfo_A(D) = \sum_{i=1}^{v} |D_i| / |D| \times DEM(D_i)$$

The term Di is the ith partition data. DEM(D) is the expected information required to classify a tuple from D based on the partitioning by A.

Let $F_1(\varepsilon)$ be the data sensitive function defined as $e^{-\alpha\varepsilon}$ and $F_2(\varepsilon)$ be the sensitive cost function defined as $(1/\sigma\sqrt{2\pi})\,\mathrm{e}^{-(\varepsilon-\mu)^2/2\sigma}$ which follows Normal

Distribution with mean μ and variance σ^2.

i.e $F_1(\varepsilon) = e^{-\alpha\varepsilon}$ and $F_2(\varepsilon) = (1/\sigma\sqrt{2\pi})\,\mathrm{e}^{-(\varepsilon-\mu)^2/2\sigma}$

$\mu = 0$

if $x = \varepsilon$

then $F_2(\varepsilon) = (1/\sigma\sqrt{2\pi})\,\mathrm{e}^{-\varepsilon^2/2\sigma}$

Now consider,

Min $F = w'F(\varepsilon) = w_1'F_1(\varepsilon) + w_2'F_2(\varepsilon) + \sigma_\varepsilon$

$\approx w_1'\, e^{-\alpha\varepsilon} + w_2'\,(1/\sigma\sqrt{2\pi})\,\mathrm{e}^{-\varepsilon^2/2\sigma}$

Subject to $\varepsilon_i^{(p)} \le \varepsilon_i \le \varepsilon_i^{(q)}, \forall i = 1...n$

$$w_1' + w_2' = 1 \quad \text{----(1)}$$

$$w_1' > 0$$

$$w_2' > 0$$

$$\sigma_\varepsilon > 0$$

By solving the above multi-objective equation we get,

$$\frac{dF}{d\varepsilon} = 0$$

$$\frac{d}{d\varepsilon} w_1'\, e^{-\alpha\varepsilon} + \frac{d}{d\varepsilon} w_2'\,(1/\sigma\sqrt{2\pi})\,\mathrm{e}^{-\varepsilon^2/2\sigma} = 0$$

$$-w_1'\,\alpha.e^{-\alpha\varepsilon} + w_2'\,(1/\sigma^2\sqrt{2\pi})\varepsilon\,\mathrm{e}^{-\varepsilon^2/2\sigma} = 0$$

$$w_2'\,(1/\sigma^2\sqrt{2\pi})\varepsilon\,\mathrm{e}^{-\varepsilon^2/2\sigma} = w_1'\,\alpha.e^{-\alpha\varepsilon}$$

$$w_2'/w_1' = \alpha.e^{-\alpha\varepsilon}/(1/\sigma^2\sqrt{2\pi})\varepsilon\,\mathrm{e}^{-\varepsilon^2/2\sigma} \quad \text{--(2)}$$

Let $F_3' = \alpha.e^{-\alpha\varepsilon}/(1/\sigma^2\sqrt{2\pi})\varepsilon\,\mathrm{e}^{-\varepsilon^2/2\sigma}$ be the function mapping from $\square \to \square$ ----(3)

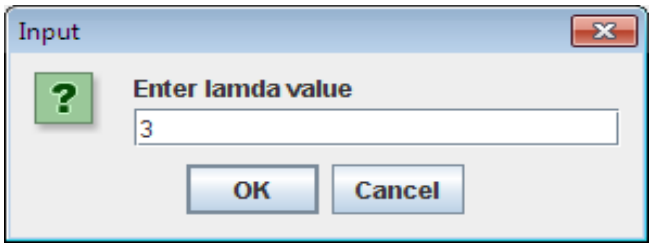
By solving $F_3' = \varepsilon^{(p)}$ and equation (1) then we will get lower bound limits $\delta^l{}_1, \delta^u{}_1$ of $\varepsilon^{(p)}$.

Similarly by solving $F_3' = \varepsilon^{(q)}$ and equation (1) we will get upper bound limits $\delta^l{}_2, \delta^u{}_2$ of $\varepsilon^{(q)}$.

Different nodes in the distributed decision tree can choose any values within the range of lower and upper bounds for privacy preserving.

Experimental Results:

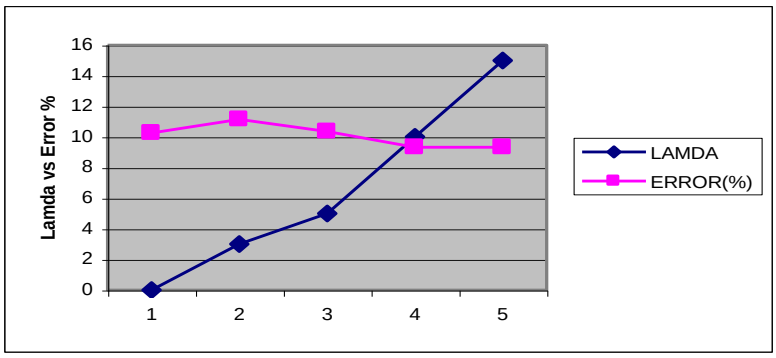
Outliers Results:

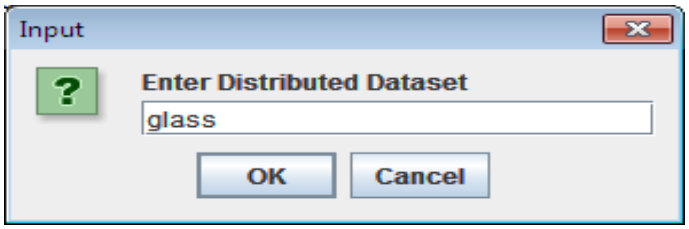


Min outlier and max outlier 1.2146790466222046 1.8220517944992913
Min outlier and max outlier -68.25250510420858 95.06820603878802
Min outlier and max outlier -141.55625177676382 146.92531719732455
Min outlier and max outlier -48.48205801799239 51.371871102104535
Min outlier and max outlier -4.803644897071962 150.10551405595052
Min outlier and max outlier -64.72212848113165 65.71624063066434

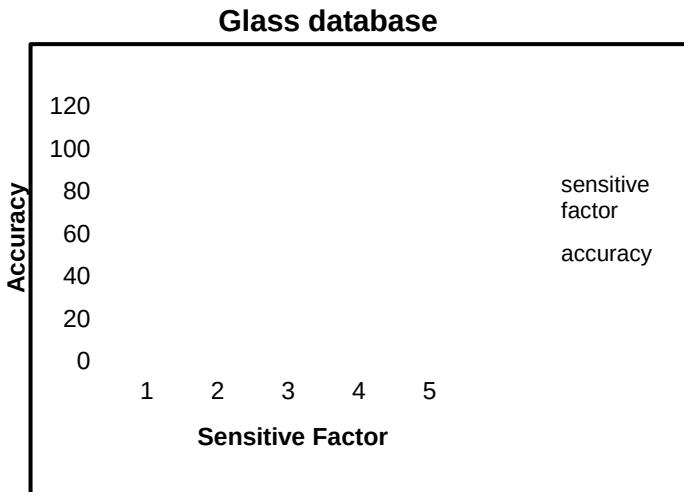
Glass dataset:

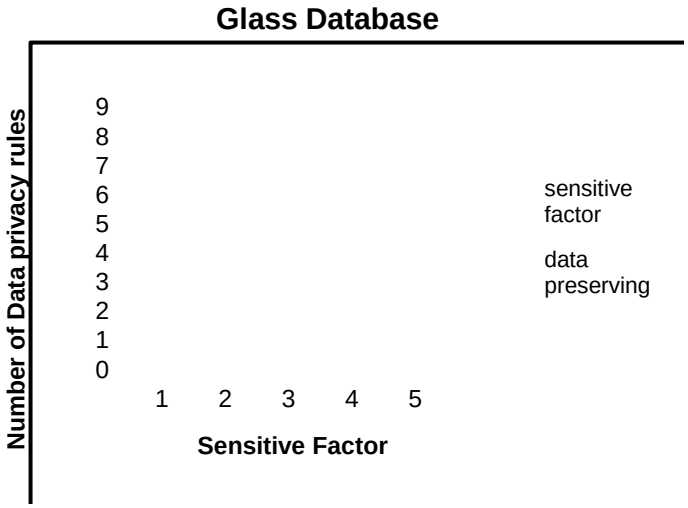
Lamda	0	3	5	10	15
Error	10.26	11.15	10.35	9.33	9.33
Outliers	0	36	49	51	51
Accuracy	66.8224	64.0449	65.4545	69.9387	69.9387

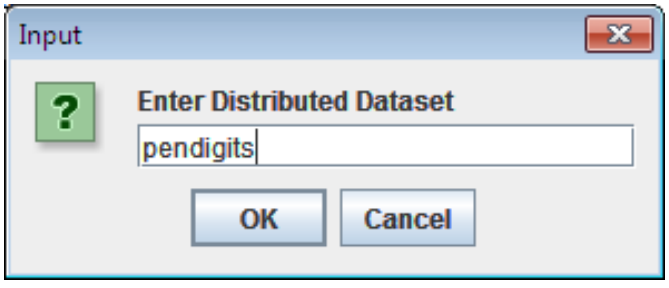




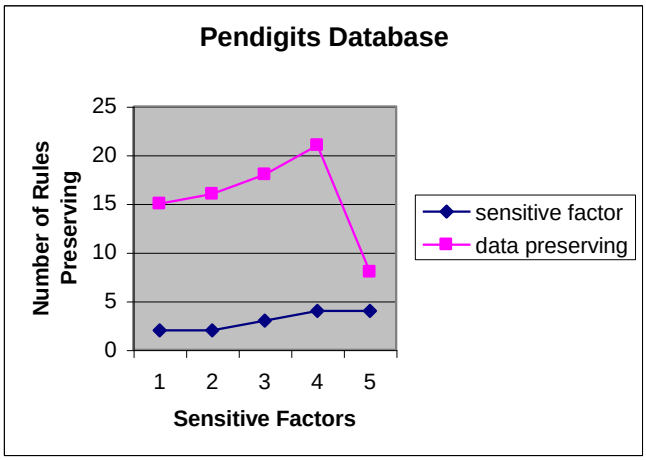

```
|   |   |   |   |   Fe <= 0.12
|   |   |   |   |   |   Mg <= 3.54: vehic wind float (5.0)
|   |   |   |   |   |   Mg > 3.54
|   |   |   |   |   |   |   RI <= 1.51667: *****
|   |   |   |   |   |   |   RI > 1.51667: *****
|   |   |   |   |   Fe > 0.12: build wind non-float (2.0)
|   |   |   RI > 1.51707
|   |   |   |   K*******************
|   |   |   |   |   Mg <= 3.34: build wind non-float (2.0)
|   |   |   |   |   ********** > 3.34
|   |   |   |   |   |   Si <= 72.64
|   |   |   |   |   |   |   Na <= 14.01: build wind float (14.0)
|   |   |   |   |   |   |   Na > 14.01
|   |   |   |   |   |   |   |   RI <= 1.52211
|   |   |   |   |   |   |   |   |   Na <= 14.32: *****
|   |   |   |   |   |   |   |   |   Na > 14.32: *****
|   |   |   |   |   |   |   |   RI > 1.52211: build wind float (3.0)
|   |   |   |   |   |   Si > 72.64: vehic wind float (3.0)
|   |   |   |   K > 0.23
|   |   |   |   |   Mg <= 3.75
|   |   |   |   |   |   Fe <= 0.14
|   |   |   |   |   |   |   RI <= 1.52043: build wind float (36.0)
|   |   |   |   |   |   |   RI > 1.52043: build wind non-float (2.0/1.0)
|   |   |   |   |   |   Fe > 0.14
|   |   |   |   |   |   |   Al <= 1.17: build wind non-float (5.0)
|   |   |   |   |   |   |   Al > 1.17: build wind float (6.0/1.0)
|   |   |   |   |   Mg > 3.75: build wind non-float (10.0)
```

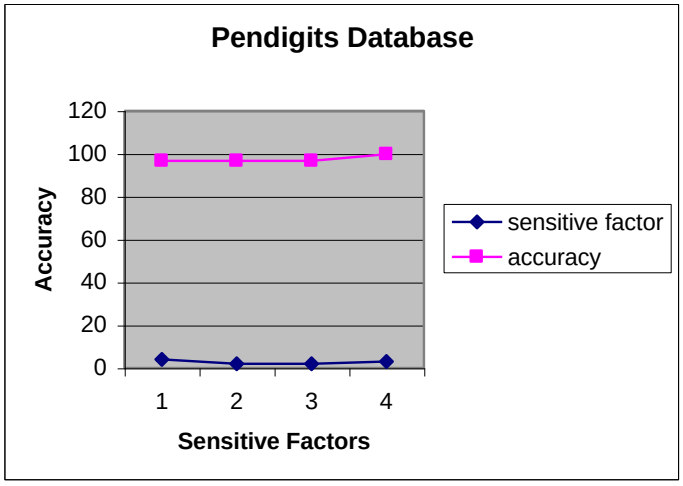

Glass database
Accuracy
120
100
80
60
40
20
0
1
2
3
4
5
Sensitive Factor
sensitive factor
accuracy


Glass Database
Number of Data privacy rules
9
8
7
6
5
4
3
2
1
0
1
2
3
4
5
Sensitive Factor
sensitive factor
data preserving

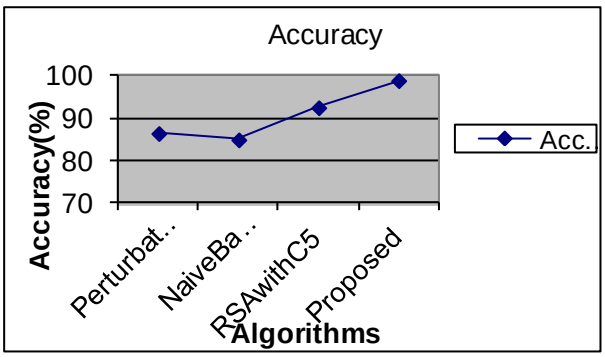

Input
Enter Distributed Dataset
pendigits
OK
Cancel

```
|   |   |   |   |   input9 > 34
|   |   |   |   |   |   input7 <= 64
|   |   |   |   |   |   |   input3 <= 29
|   |   |   |   |   |   |   |   input2 <= 65: *****
|   |   |   |   |   |   |   |   input2 > 65: *****
|   |   |   |   |   |   |   input3 > 29
|   |   |   |   |   |   |   |   input2 <= 87
|   |   |   |   |   |   |   |   |   input9 <= 37
|   |   |   |   |   |   |   |   |   |   input7 <= 60: 1 (15.0)
|   |   |   |   |   |   |   |   |   |   input7 > 60: 2 (3.0)
|   |   |   |   |   |   |   |   |   input9 > 37: 1 (98.0)
|   |   |   |   |   |   |   |   input2 > 87: 2 (4.0/1.0)
|   |   |   |   |   |   input7 > 64
|   |   |   |   |   |   |   input2 <= 93
|   |   |   |   |   |   |   |   input9********************
|   |   |   |   |   |   |   |   |   input3 <= 54: 2 (133.0)
|   |   |   |   |   |   |   |   |   input3 > 54
|   |   |   |   |   |   |   |   |   |   input5 <= 84: 1 (15.0/1.0)
|   |   |   |   |   |   |   |   |   |   input5 > 84
|   |   |   |   |   |   |   |   |   |   |   input9 <= 52: 2 (69.0/1.0)
|   |   |   |   |   |   |   |   |   |   |   input9 > 52: 1 (6.0/1.0)
|   |   |   |   |   |   |   |   input9 > 71
|   |   |   |   |   |   |   |   |   input12 <= 22: 1 (8.0)
|   |   |   |   |   |   |   |   |   input12 > 22: 3 (2.0)
|   |   |   |   |   |   ********** > 93
```





Accuracy Comparison with Traditional Algorithms:



REFERENCES

[1] Sweeney L. 2002 K-anonymity: A model for protecting Journal on Uncertainty, fuzziness and Knowledge based systems.

[2] Weiwei Fang, Bingru Yang, "Privacy Preserving Decision Tree Learning Over Vertically Partitioned Data," In Proceedings of the International Conference on Computer Science & Software Engineering, 2008.

[3] J. Shrikant Vaidya, "Privacy preserving data mining over vertically partitioned data," PH.D Thesis of Purdue University, Aug 2004, pp 28-34.

[4] Y. Lindell, B. Pinkas, "Privacy preserving data mining," In Journal of Cryptology vol. 15, no. 3, 2002, pp 177–206.

[5] Pui K. Fong and Jens H. Weber-Jahnke, "Privacy Preserving Decision Tree Learning Using Unrealized Data Sets" ." IEEE Transl. on knowledge and data engineering, vol. 24, no. 2, February2012.

[6] S. Russell and N. Peter, Artificial Intelligence. A Modern Approach 2/E. Prentice-Hall, 2002.

[7] F. Emekci* , O.D. Sahin, D. Agrawal, A. El Abbadi, "Privacy preserving decision tree learning over multiple parties

AUTHORS PROFILE

Mr.Y.A.Siva Prasad, Research Scholar in CSE Department,, KL University, Andhra Pradesh, and Life member of **CSI, IAENG**, Having 10 years of Teaching Experience .

Dr. G.Ramakrishna, Prof., CSE dept,KLU Vaddeswaram carried out research at Saha Institute, Calcutta for about five and half years from 1966-1971 in theoretical physics using computers at ISI, Calcutta, IIT Khargapur, IIT Kanpur, & IIT Madras extensively for solving problems in nuclear models and Obtained Ph.D. in 1975. Worked as a lecturer in NIT,Warangal in Physics Dept.From 1971-1975 teaching B.Tech and M.Sc.(tech) students .Taught computer programming for M.Phil (Computer Methods)students at University of Hyderabad ,Hyderabad from 1975 to 1980.Taught Computer programming and application analysis On leak detection in Oil and Gas Pipelines to the customers Of ONGC, IOC, OIL India, HPCL and BPCL at ECIL and Customer sites across India from 1975 to 2003