ISSN (Print)  : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative  Research in Computer and Communication Engineering
*Vol. 1, Issue 2, April 2013*

# Privacy Preservation Decision Tree Based On Data Set Complementation

**Madhusmita Sahu[1], Debasis Gountia[2], Neelamani Samal[3]**

M. Tech. Scholar, Dept. of Information Technology, College of Engineering and Technology, Bhubaneshwar, India[1]

Assistant Professor, Dept. of Computer Science & Application, College Of Engineering and Technology, Bhubaneshwar, India [2]

Assistant Professor, Dept. of Computer Science & Engineering, Gandhi Institute for Education and Technology, Bhubaneshwar, India[3]

**Abstract**: Privacy preservation in data mining has been a popular and an important research area for more than a decade due to its vast spectrum of applications. A new class of data mining method called privacy preserving data mining algorithm has been developed. The aim of this algorithm is to protect the sensitive information in data from the large amount of data set. The privacy preservation of data set can be expressed in the form of decision tree, cluster or association rule. This paper proposes a privacy preservation based on data set complement algorithms which store the information of the real dataset.  So that the private data can be safe from the unauthorized party, if some portion of the data can be lost, then we can reconstructed the original data set from the unrealized dataset and the perturbing data set.

**Keywords**: Data mining, classification, machine learning, privacy preservation.

## I. INTRODUCTION

Data mining is a recently emerging field, connecting the three worlds of databases, statistics and artificial intelligence. Data mining is the process of extracting knowledge or pattern from large amount of data. It is widely used by researchers for science and business process. Data collected from information providers are important for pattern reorganization and decision making. The data collection process takes time and efforts hence sample datasets are sometime stored for reuse. However attacks are attempted to steal these sample datasets and private information may be leaked from these stolen datasets. Therefore privacy preserving data mining are developed to convert sensitive datasets into sanitized version in which private or sensitive information is hidden from unauthorized retrievers.

Privacy preserving data mining refers to the area of data mining that seeks to safeguard sensitive information from unsanctioned or unsolicited disclosure. Privacy preservation data mining was introducing to preserve the privacy during mining process to enable conventional data mining technique. Many privacy preservation approaches were developed to protect private information of sample dataset.

## II. RELATED WORKS

In Privacy Preserving Data Mining: Models and Algorithms [14], Aggarwal and Yu classify privacy preserving data mining techniques, including data modification and cryptographic, statistical, query auditing and perturbation-based strategies. Statistical, query auditing and most cryptographic techniques are subjected beyond the focus of this paper. In this section, we explore the privacy preservation techniques for storage privacy attacks.

Data modification techniques maintain privacy by modifying attribute values of the sample data sets. Essentially, data sets are modified by eliminating or unifying uncommon elements among all data sets. These similar data sets act as masks for the others within the group because they cannot be distinguished from the others; every data set is loosely linked with a certain number of information providers. k-anonymity [15] is a data modification approach that aims to protect private information of the samples by generalizing attributes. K-anonymity trades privacy for utility. Further, this approach can be applied only after the entire data collection process has been completed.

Perturbation-based approaches attempt to achieve privacy protection by distorting information from the original data sets. The perturbed data sets still retain features of the originals so that they can be used to perform data mining directly or indirectly via data reconstruction. Random substitutions [16] is a perturbation approach that randomly substitutes the values of selected attributes to achieve privacy protection for those attributes, and then applies data reconstruction when these data sets are needed for data mining. Even though privacy of the selected attributes can be protected, the utility is not recoverable because the reconstructed data sets are random estimations of the originals.

Most cryptographic techniques are derived for secure multiparty computation, but only some of them are applicable to our scenario. To preserve private information, samples are encrypted by a function, f, (or a set of functions) with a key, k, (or a set of keys); meanwhile, original information can be reconstructed by applying a decryption function, f_1, (or a set of functions) with the key, k, which raises the security issues of the decryption function(s) and the key(s). Building meaningful decision trees needs encrypted data to either be decrypted or interpreted in its encrypted form. The (anti)monotone framework [17] is designed to preserve both the privacy and the utility of the sample data sets used for decision tree data mining. This method applies a series of encrypting functions to sanitize the samples and decrypts them correspondingly for building the decision tree. However, this approach raises the security concerns about the encrypting and decrypting functions. In addition to protecting the input data of the data mining process, this approach also protects the output data, i.e., the generated decision tree. Still, this output data can normally be considered sanitized because it constitutes an aggregated result and does not belong to any individual information provider. In addition, this approach does not work well for discrete-valued attributes.

## III. DECISION TREE CLASSIFIER

A decision tree[3][4][5] is defined as "a predictive modeling technique from the field of machine learning and statistics that builds a simple tree-like structure to model the underlying pattern of data". Decision tree is one of the popular methods is able to handle both categorical and numerical data and perform classification with less computation. Decision trees are often easier to interpret. Decision tree is a classifier which is a directed tree with a node having no incoming edges called root. All the nodes except root have exactly one incoming edge. Each non-leaf node called internal node or splitting node contains a decision and most appropriate target value assigned to one class is represented by leaf node. Decision tree classifier is able to break down a complex decision making process into collection of simpler decision. The complex decision is subdivided into simpler decision on the basis of splitting criteria. It divides whole training set into smaller subsets. Information gain, gain ratio, gini index are three basic splitting criteria to select attribute as a splitting point. Decision trees can be built from historical data they are often used for explanatory analysis as well as a form of supervision learning. The algorithm is designed in such a way that it works on all the data that is available and as perfect as possible. According to Breiman *et al.* [6] the tree complexity has a crucial effect on its accuracy performance. The tree complexity is explicitly controlled by the pruning method employed and the stopping criteria used. Usually, the *tree complexity* is measured by one of the following metrics:

- The total number of nodes;
- Total number of leaves;
- Tree depth;
- Number of attributes used.

Decision tree induction is closely related to rule induction. Each path from the root of a decision tree to one of its leaves can be transformed into a rule simply by conjoining the tests along the path to form the antecedent part, and taking the leaf's class prediction as the class value. The resulting rule set can then be simplified to improve its accuracy and comprehensibility to a human user [7].

Flowchart for tree based classification is shown in fig.1

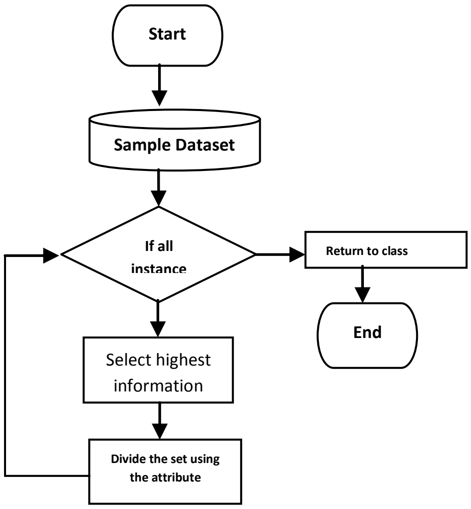

Start
Sample Dataset
If all instance
Return to class
End
Select highest information
Divide the set using the attribute

Fig.1. Flowchart for tree based classification

Hyafil and Rivest proved that getting the optimal tree is NP-complete [8]. Most algorithms employ the greedy search and the divide-and-conquer approach to grow a tree. In particular, the training data set continues to be split in small. The related algorithm ID3 and C4.5 [9] adopt a greedy approach in which decision trees are constructed in top down recursive divide and conquer manner. ID3 was one of the first Decision tree algorithms. It works on wide variety of problems in both academia and industry and has been modified improved and borrowed from many times over. ID3 picks splitting value and predicators on the basis of gain in information that the split or splits provide. Gain represents difference between the amount of information that is needed to correctly make a prediction both before and after the split has been made. Information gain is defined as the difference between the entropy of original segment and accumulated entropies of the resulting split segment. C4.5 is an extension of ID3, presented by the same author (Quinlan, 1993). It uses gain ratio as splitting criteria.

The splitting ceases when the number of instances to be split is below a certain threshold. C4.5 can handle numeric attributes. It performs error based pruning after growing phase. It can use corrected gain ratio induce from a training set that incorporates missing values.

## IV. DATA SET COMPLEMENTATION APPROACH

Privacy preservation via dataset complementation is a data perturbed approach that substitutes each original dataset with an entire unreal dataset. Unlike privacy protection strategies, this new approach preserves the original accuracy of the training datasets without linking the perturbed datasets to the information providers. In other words, dataset complementation can preserve the privacy of individual records and yield accurate data mining results. However, this approach is designed for discrete-value classification only, such that ranged values must be defined for continuous values.

### *A. Universal Set and Data Set Complement*

In set theory, a universal set $U$ is a set which contains all elements [20]. In this paper, a universal set $T^U$ , relating to a data table $T$ , is a set of datasets that contains a single instance of each valid dataset of $T$ . In other words, any combination of a possible value from each attribute in the dataset sequence of $T$ exists in $T^U$ . If $t$ is a dataset in $T$ associated with a tuple of attributes $< a1, a2, \dots am >$ and $a_i$ has $n_i$ possible values $K_i = \{k_1, k_2 \dots k_{ni}\}$, then $< l[a_1], l[a_2], \dots l[a_i] \dots, l[a_m] > \in T^U$ and $l[a_i] \in K_i$ .

We define: $T^U$ is a set containing a single instance of all possible datasets in data table $T$. The table associates with attributes $< Outlook, Humidity, Wind, Play >$ and possible attribute values are defined as: *Weather* = {*Sunny*, *Overcast*, *Rain*}, *Humidity* = {*High*, *Normal*}, *Wind* = {*Strong*, *Weak*} and *Play* = {*Yes*, *No*}; Since the datasets in a data table are not necessarily unique, we allow for multiple instances of an element existing in the same set (known as a multiset, or a bag[21]). If $T_D$ is a subset of *T* and *q* is a positive integer, then we define:

A *q*-multiple-of $T_D$ , denoted as $qT_D$ , is a set of datasets containing *q* instances of each dataset in $T_D$ .Therefore, $2T_D$ = { *Sunny*, *High*, *Strong*, *Yes}* , {*Sunny*, *High*, *Strong*, *No* }, {*Sunny*, *High*, *Weak*, *Yes* }, {*Sunny*, *High*, *Weak*, *No* }, {*Sunny*, *Normal*, *Strong*, *Yes* }, {*Sunny*, *Normal*, *Strong*, *No* }, {*Sunny*, *Normal*, *Weak*, *Yes* }, {, *Normal*, *Weak*, *No* }, {*Overcast*, *High*, *Strong*, *Yes* }, {*Overcast*, *High*, *Strong*, *No* }, {*Overcast*, *High*, *Weak*, *Yes* }, {*Overcast*, *High*, *Weak*, *No* }, {*Overcast*, *Normal*, *Strong*, *Yes* }, {*Overcast*, *Normal*, *Strong*, *No* }, {*Overcast*, *Normal*, *Weak*, *Yes* }, {*Overcast*, *Normal*, *Weak*, *No* }, {*Rain*, *High*, *Strong*, *Yes* }, {*Rain*, *High*, *Strong*, *No* }, {*Rain*, *High*, *Weak*, *Yes* }□ , {*Rain*, *High*, *Weak*, *No* },
{*Rain*, *Normal*, *Strong*, *Yes* }, {*Rain*, *Normal*, *Strong*, *No* }, {*Rain*, *Normal*, *Weak*, *Yes* }, {□*Rain*, *Normal*, *Weak*, *No* }, {*Sunny*, *High*, *Strong*, *Yes* }, {*Sunny*, *High*, *Strong*, *No* }, {*Sunny*, *High*, *Weak*, *Yes* }, {*Sunny*, *High*, *Weak*, *No* }, {*Sunny*, *Normal*, *Strong*, *Yes* }, {*Sunny*, *Normal*, *Strong*, *No* }, {*Sunny*, *Normal*, *Weak*, *Yes* }, {*Sunny*, *Normal*, *Weak*, *No* }, {*Overcast*, *High*, *Strong*, *Yes* }, {*Overcast*, *High*, *Strong*, *No* }, {*Overcast*, *High*, *Weak*, *Yes* }, {*Overcast*, *High*, *Weak*, *No* }, {*Overcast*, *Normal*, *Strong*, *Yes* }, {*Overcast*, *Normal*, *Strong*, *No* }, {*Overcast*, *Normal*, *Weak*, *Yes* },{*Overcast*, *Normal*, *Weak*, *No* }, {*Rain*, *High*, *Strong*, *Yes* }, {*Rain*, *High*, *Strong*, *No* }, {*Rain*, *High*, *Weak*, *Yes* }, {*Rain*, *High*, *Weak*, *No* }, {*Rain*, *Normal*, *Strong*, *Yes* }, {*Rain*, *Normal*, *Strong*, *No* }, {*Rain*, *Normal*, *Weak*, *Yes* }, {*Rain*, *Normal*, *Weak*, *No* }}

We introduce, with examples, the foundations of dataset complementation and its application in decision-tree learning. The data tables in these examples have an attribute "Sample #", which is used as a primary key reference but not as an option of a decision or test attributes.

### B. *Data Set Complement*

A relative complement of $X$ in $Y$, denoted as $Y \setminus X$ , refers to all elements in set $Y$, excepting those in set $X$ . $Y \setminus X$ can be determined by subtracting $X$ from $Y$. If $Y$ is a universal set $U$ , then $U \setminus X$ is called an absolute complement and denoted as $X^C$. In this paper, we apply the above concepts to the definitions of a complement of a set of datasets, relating to a data table $T$ .

A relative complement of a datasets $T_{D_2}$ in a set of datasets $T_{D_1}$ is denoted as $T_{D_1} \setminus T_{D_2}$ and equals to $T_{D_1}$ - $T_{D_2}$ .

An absolute complement of a set of datasets $T_{D_1}$ is denoted as $T_{D_1}{}^C$ and equal to $T^U / T_D$ .

A q-absolute-complement of a set of datasets $T_{D_1}$ is denoted as $qT_{D_1}{}^C$ and equal to $qT_{D_1} \setminus T_D$ .

Since $qT_{D_1}{}^C = qT_{D_1} - T_D \Rightarrow qT^U = qT_D{}^C + T_D, qT_D \subseteq qT^U and T_D \subseteq qT^U$ . Let's reconsider an example that is we have a set of datasets $T_{D1}$ = {<Rain, High, Weak, Yes>, <Sunny, High, Strong, No >} and $T_{D2}$ ={<Overcast, High, Weak, Yes>, <Overcast, High, Weak, No>, <Overcast, Normal, Weak, Yes>}, then $T_{D1}{}^C \setminus T_{D2} = T^U - T_{D1} - T_{D2}$ ={<Sunny, High, Strong, Yes,<Sunny, High, Weak, Yes>,<Sunny, High, Weak, No>,<Sunny, Normal, Strong, Yes>,<Sunny, Normal, Strong, No>, <Sunny, Normal, Weak, Yes>,<Sunny, Normal, Weak, No>,<Overcast, High, Strong, Yes>,<Overcast, High, Strong, No>, <Overcast, Normal, Strong, Yes>,<Overcast, Normal, Strong, No>, <overcast, Normal, Weak, No>,<Rain, High, Strong, Yes>,<Rain, High, Strong, No>,<Rain, High, Weak, No>,<Rain, Normal, Strong, Yes>, <Rain, Normal, Strong, No>, <Rain, Normal, Weak, Yes>, <Rain, Normal, Weak, No> }.

If $T_{D2}$ is a subset of $T_{D1}$, then all the elements existing in $T_{D2}$ also exist in $T_{D1}$. Thus, the information content gained by classifying q-absolute-complement of a set of datasets, say $T_D$, could be determined by using the size of $qT^U$ and the information of $T_D$.

| Outlook | Wind | Play |
|---|---|---|
| Sunny | Weak | No |
| Sunny | Strong | No |
| Overcast | Weak | Yes |
| Rain | Weak | Yes |
| Rain | Weak | Yes |
| Rain | Strong | No |
| Overcast | Strong | Yes |

(a) $T_S$

| outlook | WIND | Play |
|---|---|---|
| Sunny | Strong | Yes |
| Sunny | Strong | No |
| Sunny | Weak | Yes |
| Sunny | Weak | No |
| Overcast | Strong | No |
| Overcast | Weak | No |
| Overcast | Weak | No |
| Rain | Strong | Yes |
| Rain | Strong | No |
| Rain | Weak | No |

(b) $T^{'}$

| Outlook | Wind | Play |
|---|---|---|
| Sunny | strong | yes |
| Sunny | Weak | Yes |
| Overcast | strong | yes |
| Overcast | strong | No |
| Overcast | Weak | No |
| Rain | strong | yes |
| Rain | Weak | No |

(c) $T^P$

Fig 2. Unrealizing training samples in (a) by calling Unrealized-Training Set ($T_S, T^U$, {},{}) .The resulting tables $T^{'}$ and $T^P$ are given in (b) and (c)

## *C. Unrealized Training Set*

A training set $T_S$ is constructed by inserting sample data sets into a data table. However, a data set complementation approach, as presented in this paper requires an extra data table, $T^P$. $T^P$ is a perturbing set that generated unreal data sets which are used for converting the sample data into unrealized training set, $T^{'}$. The algorithm for unrealized the training set, $T_S$, as shown follows:

### Algorithm

```
UNREALIZED TRAINIG-SET ($T_S, T^U, T^{'}, T^P$)
inputs: $T_S$, a set of input sample datasets
        $T^U$, a universal set
        $T^{'}$, a set of output training datasets
        $T^P$, a set of unreal datasets
Output: $<T^{'}, T^P>$
1.  if $T_S$ is empty then return $<T^{'}, T^P>$
2.  $t_i \leftarrow$ a dataset in $T_S$
3.  if $t_i$ is an element of $T^P$ and $T^P \setminus \{t_i\} \neq \{\}$ then
4.      $T^P \leftarrow T^P - \{t_i\}$
5.      $t_i \leftarrow$ a dataset in $T^P$
6.  else        $T^P \leftarrow T^P + T^U - \{t_i\}\}$
7.  $t_i \leftarrow$ a dataset in $T^P$
```

```
8.  return UNREALIZED TRAINING-SET   $(\left(T_S - \{t_I\}, T^U, T^{'} + \{t_i\}, T^P - \{t_i\}\right))$
```

To unrealized the samples, $T_S$, we initialize both $T^{'}$ and $T^P$ as empty sets, i.e., we invoke the above algorithm with Unrealized-training-set($T_S$, $T^U$, {},{}). Figs. 2(b) and 2(c) show the tables that result from the unrealizing process of the samples in fig. 2(a) the resulting training set contains some dummy data sets excepting the ones in $T_S$. The elements in the resulting data sets are unreal individuals, but meaningful when they are used together to calculate the information required by a modified ID3 algorithm.

## V. DECISION TREE GENERATION

The ID3 algorithm [18] build a decision tree by calling algorithm Choose-Attribute recursively. This algorithm selects a test attribute (with the smallest entropy) according to the information content of the training set $T_S$. The information entropy functions are given as

$$H_d(T_s) = -\sum_{j=1}^{n} P(d = v_i)\log_2 p(d = v_i) = -\sum_{j=1}^{m} \frac{\left|T_{s\,d=v_i}\right|}{\left|T_s\right|}\log_2 \frac{\left|T_{s\,d=v_i}\right|}{\left|T_s\right|}$$

and $H_d(T_s \mid a)$ is the condition information content of $d$ with given $a$, equals:

$$H_d(T_s \mid a) = \sum_{i=1}^{n} P(a = K_i) H_d(T_{s\,a=k_i}) = \sum_{i=1}^{n} \frac{\left|T_{s\,a=k_j}\right|}{\left|T_s\right|} H_d\left(T_{s\,a=k_i}\right)$$

where a is the test attribute with possible values $k_i$ ($l$ is an integer and $(1 \le l \le n)$ and $d$ is the decision attribute with possible values $v_i$ ($l$ is an integer and ($1 \le j \le m$) and the Majority-value retrieves the most frequent value of the decision attribute of $T_S$

## VI. ALGORITHM

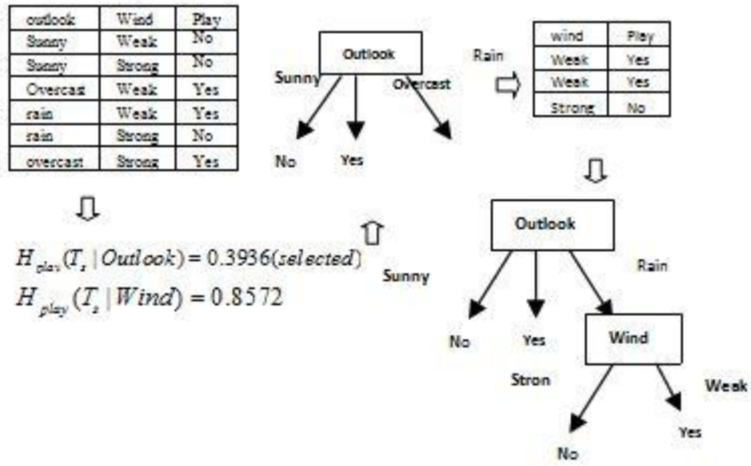
```
GENERATE-DECISION-TREE(examples, attributes, default)

inputs: examples, set of examples
        attributes, set of attributes
        default, default value for the goal predicate

output: tree, a decision tree

1. if examples is empty then return default
2. else if all examples have the same classification   then
       return the classification
3. else if attributes is empty then
       return MAJORITY-VALUE(examples)
4. else
5. best $\leftarrow$ CHOOSE-ATTRIBUTE(attributes, examples)
6. tree $\leftarrow$ a new decision tree with root test best
7.  for each value $v_i$ of best do
```

```
8.  examples ← {element $s_i$ of examples with best= $v_i$ }
9. m ← MAJORITY-VALUES(example $s_i$ )
10. subtree ← GENERATE-DECISION-TREE (example $s_i$ , attributes ← best, m)
11.add a branch to tree with lable $v_i$ subtree
12. return tree
```



| outlook | Wind | Play |
|---|---|---|
| Sunny | Weak | No |
| Sunny | Strong | No |
| Overcast | Weak | Yes |
| rain | Weak | Yes |
| rain | Strong | No |
| overcast | Strong | Yes |

| wind | Play |
|---|---|
| Weak | Yes |
| Weak | Yes |
| Strong | No |

$$H_{play}(T_s \mid Outlook) = 0.3936(selected)$$

$$H_{play}(T_s \mid Wind) = 0.8572$$

Fig 3. Illustration of Generate-Decision-Tree process by applying the conventional ID3 approach with the Original samples $T_s$

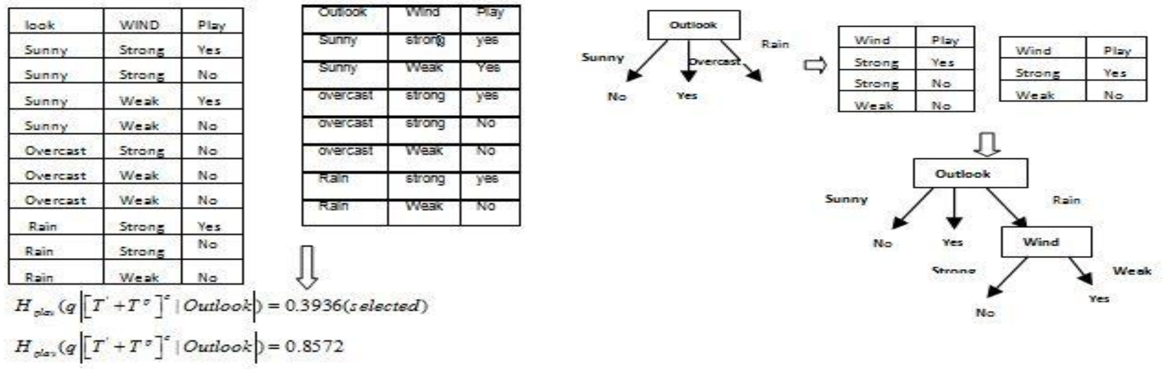
### *A. Information Entropy Determination*

Form the algorithm Unrealized-Training-Set, it is obvious that the size of $T_S$ is the same as the size of $T^{'}$. Furthermore, all datasets in ($T^{'}+T^{P}$) are based on the data set in $T^{U}$, excepting the ones in $T_S$, i.e. $T_S$ is the q-absolute-complement of ($T^{'}+T^{P}$) for store positive integer q according to q-absolute complement dataset, the size of q$T^{U}$ can be computed from the sizes of $T^{'}$ and $T^{P}$, with $qT^{u} = 2*\left|T^{'}\right|+\left|T^{p}\right|$. Therefore, entropies of the original data sets, $T_S$, with any decision attribute and any test attribute, can be determined by the unreal training set, $T^{'}$, and perturbing set $T^{P}$, as we will show with Theorem 1 below:

**Definition** $G(x) = x\log_2 x$

**Theorem 1.** if $T_s = q\left|T^{'}+T^{p}\right|$ and $\left|T^{'}\right| = \left|T^{s}\right|$ for some positive integer q, then.

$$\Pr oof .T_s = q\left|T^{'}+T^{p}\right|^{c}$$
$$\Rightarrow T_s = qT^{u}-\left(T^{'}+T^{p}\right)$$
$$\Rightarrow \left|T_s\right| = \left|qT^{u}-\left(T^{'}+T^{p}\right)\right|$$
$$\Rightarrow \left|T_s\right| = \left|qT^{u}\right|-\left(T^{'}+T^{p}\right)$$
$$\Rightarrow \left|T_s\right| = \left|qT^{u}\right|-\left|T^{'}\right|-\left|T^{p}\right|$$
$$\Rightarrow \left|T_s\right| = \left|qT^{u}\right|-\left|T^{'}\right|-\left|T^{p}\right|$$
$$\Rightarrow \left|T^{'}\right| = \left|qT^{u}\right|-\left|T^{'}\right|-\left|T^{p}\right|, \because \left|T^{'}\right| = \left|T_s\right|$$
$$\Rightarrow \left|qT^{u}\right| = 2*\left|T^{'}\right|+\left|T^{p}\right|$$



| look | WIND | Play |
|---|---|---|
| Sunny | Strong | Yes |
| Sunny | Strong | No |
| Sunny | Weak | Yes |
| Sunny | Weak | No |
| Overcast | Strong | No |
| Overcast | Weak | No |
| Overcast | Weak | No |
| Rain | Strong | Yes |
| Rain | Strong | No |
| Rain | Weak | No |

| Outlook | Wind | Play |
|---|---|---|
| Sunny | strong | yes |
| Sunny | Weak | Yes |
| overcast | strong | yes |
| overcast | strong | No |
| overcast | Weak | No |
| Rain | strong | yes |
| Rain | Weak | No |

$H_{play}(q\left|\left[T^{'}+T^{p}\right]^{c} \mid Outlook\right|) = 0.3936(selected)$

$H_{play}(q\left|\left[T^{'}+T^{p}\right]^{c} \mid Outlook\right|) = 0.8572$

| Wind | Play |
|---|---|
| Strong | Yes |
| Strong | No |
| Weak | No |

| Wind | Play |
|---|---|
| Strong | Yes |
| Weak | No |

Fig. 4 Illustration of Modified-Decision-Tree process by applying the modified ID3approach with the unrealized samples( $T^{'}+T^{P}$ ). For each step the entropy values and resultinh subtrees are exactly the same as the results of the traditional approach.

## B. *Modified Decision Tree Generation Algorithm*

As entropies of the original data sets, $T_S$, can be determined by the retrieval information – the contents of unrealized training set, $T^{'}$, and perturbing set, $T^{P}$, the decision tree of $T_S$, can be generated by the following algorithm.

**Algorithm** MODIFY-DECISION-TREE **(size,**, $T^{'}$, $T^{P}$, attribute, default**)**

```
inputs:  size, size of the q-multiple-of universal set
         $T^{'}$, a set of output training datasets
         $T^{P}$, a set of unreal datasets
         attributes, set of attributes
         default, default value for the goal predicate
output: tree, a decision tree
    1.  if $T^{'}+T^{P}$ is empty then return default
    2.  else if $H_{ai}(q[T^{'}+T^{P}]^{C}) = 0$ then
```

```
        return MINORITY-VALUE($T^{'} + T^{P}$)
else if attributes is empty then
        return MINORITY-VALUE($T^{'} + T^{P}$)
else
best $\leftarrow$ CHOOSE-ATTRIBUTE(attributes, size, ($T^{'} + T^{P}$))
tree $\leftarrow$ a new decision tree with root test best
            size $\leftarrow$ size/ number of possible values $v_i$ in best
for each value $v_i$ of best do
$T_i \leftarrow$ {datasets in $T^{'}$ with best $v_i$ }
$T^{P} \leftarrow$ {datasets in $T^{P}$ with best = $v_i$ }
m $\leftarrow$ MINORITY-VALUE($T^{'} + T^{P}$) sub tree $\leftarrow$ MODIFY-DECISION-TREE (size, $T^{'}$ , $T^{P}$ , attribute-best ,m) add a
branch to tree with label $v_i$ and subtree
```

Similar to the traditional ID3 approach, algorithm Choose-Attribute selects the test attribute using the ID3 criteria, based on the information entropies, i.e. selecting the attribute with the greatest information gain. Algorithm Minority-value retrieves the least frequent value of the decision attribute of ($T^{'} + T^{P}$), which performs the same function as algorithm Majority-Value of the tradition ID3 approach, that is receiving the most frequent value of the decision attribute of $T_S$ .To generate the decision tree with $T^{'}$, $T^{P}$ and $\left|qT^{u}\right|$( which equals $2*\left|T^{'}\right|+T^{p}$) , a possible value, $k_d$ , of the decision attribute, $a_d$ ,(which is an element of A, the set of attributes in T ) should be arbitrarily chosen, i.e., we call the algorithm Generate-Tree( $2*\left|T^{'}\right|+\left|T^{p}\right|, T_s, T^{u}, A-a_d, k_d$ ). Fig. 4 shows the resulting decision tree of our ne ID3 algorithm with unrealized sample inputs shown in figs. 2b and 2c. this decision tree is the same as the tree shown in Fig. 3 which was generated by the traditional ID3 algorithm with the original samples shown in fig. 2(a).

### *C. Data Set Reconstruction*

The previous section introduced a modified decision tree learning algorithm by using the unrealized training set, $T^{'}$ and the perturbing set, $T^{P}$. Alternatively, we could have reconstructed the original sample data sets, $T_S$ , from $T^{'}$ and $T^{P}$ (shown in fig. 4), followed by an application of the conventional ID3 algorithm for generating the decision tree from $T_S$. The reconstruction process is dependent upon the all information of $T^{'}$ and $T^{P}$ (where $q = \left(2*\left|T^{'}\right|+\left|T^{p}\right|\right)/T^{u}$; reconstruction of parts of $T_S$ based on parts $T^{'}$ and $T^{P}$ is not possible.

## VI. OUTPUT ACCURACY

The decision tree generated from the unrealized samples is same as the decision tree, Tree $T_S$ , generated from the original sample by the regular method.

### *A. Storage Complexity*

From the experiment, the storage requirement for the data set complementation approach increases from $|T_S|$ to $(2\left|T^{u}\right|-)*\left|T_s\right|$ , while the required storage may be doubled if the any attribute values technique is applied to double the sample domain. The best case happens when the samples are evenly distributed, as the storage requirement is the same as for the original.

### *B. Privacy Risk*

The average privacy loss per leaked unrealized data set is small, except the even distribution case (in which the unrealized samples are the same as the originals). By doubling the samples domain, the average privacy loss for a single leaked data set is zero, as the unrealized samples are not linked to any information provider. The randomly picked tests show that the data set complementation approach eliminates the privacy risk for most cases and always improves privacy security significantly when new values are used in the attribute.

## VII. CONCLUSION AND FUTURE WORKS

The privacy preserving process sometimes reduces the utility of training datasets, which causes inaccurate data mining results. Privacy preservation approaches focus on different areas of a data mining process, and data mining methods also vary. This paper focuses on privacy protection of the training samples applied for decision tree data mining.
In this paper we introduced a new privacy preservation technique via data set complementation by using the decision tree learning. Which convert the sample data set $T_S$ into unrealized data set $T^{'}$ and the perturbing datasets $T^{p}$. The original data set cannot be reconstructed if some portion of the unrealized data set is stolen by an unauthorized party. Therefore, there remains only a low probability of random matching of any original data set to the stolen data set $T_L$.
The data set complementation approach ensures that the privacy loss via matching is ranged from $0$ to $|T_L| * \left(|T_S| / |T_U|\right)$, where $T_U$ is the set of possible sample data sets. Therefore, this improved approach results in a matching rate that is always less than one-third of the best case of the unprotected samples. In all cases, the complexity of the sanitization process is O(|Ts |) . However, the worst case requires $\left(\left|2*\left|T_U\right|-1\right|\right)$ times the amount of storage needed for unprotected samples.

The data set complementation fails, if all training datasets were leaked, because the dataset reconstruction algorithm is generic. Therefore, further research is required to eliminate this limitation. This paper is implemented by using the ID3 decision tree algorithm and discrete values attributes. The further research should be develop by using the C4.5 decision tree algorithm with discrete value and continues value attributes. Future research should also explore means to reduce the storage requirement associated with the derived dataset complementation approach. This paper relies on theoretical proofs with limited practical tests, so testing with real samples should be the next step to gain solid ground on real-life application because the C4.5 decision tree algorithm is performed the numerical values.

## REFERENCES

[1] Agrawal, R., .and Srikant, R., Privacy-preserving data mining. In *ACM SIGMOD International Conference on Management of Data*, pages 439–450. ACM, 2000.
[2] Lindell, Y., and Pinkas. B., Privacy preserving data mining. In *Advances in Cryptology*, volume 1880 of *Lecture Notes in Computer Science*, pages 36–53. Springer-Verlag, 2000.
[3] Rokach L. and Maimon “Top-Down Induction of Decision Trees Classifiers – A Survey”, IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS: PART C, VOL. 1, NO. 11, NOVEMBER 2002.
[4] Han, J., Micheline Kamber, “Data Mining: Concepts and Techniques”, Morgan Kaufmann, 2001.
[5] Rokach, L. , Maimon O. ,”Data Mining and Knowledge Discovery Handbook “,Second edition pages 167-192, Springer Science + Business Media,2010
[6] Breiman, L. , Friedman, J. , Olshen, R., and Stone. C. ,” Classification and Regression Trees”, Wadsworth Int. Group, 1984.
[7] Quinlan, J.R., “ Simplifying decision trees”, International Journal of Man- Machine Studies, 27, 221-234, 1987.
[8] Hyafil, L. and Rivest, R. L. , "Constructing Optimal Binary Decision Trees is {NP}-Complete," *Inf. Process. Lett*, vol. 5, pp. 15-17, 1976.
[9] Quinlan, J. R. , *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
[10] Dowd, J., Xu, S. and Zhang, W., “Privacy-Preserving Decision Tree Mining Based on Random Substitions,” Proc. Int’l Conf. Emerging Trends in Information and Comm. Security (ETRICS ’06), pp. 145-159, 2006.
[11] Kargupta, H., Datta, S., Wang, Q., and Sivakumar, K., “On the privacy preserving properties of random data perturbation techniques”. In *IEEE International Conference on Data Mining*, 2003.

[12] Evfimievski, A. , Gehrke, J. , and Srikant, R.," Limiting privacy breaching in privacy preserving data mining. In *ACM Symposium on Principles of Database Systems"*, pages 211–222. ACM, 2003.

[13] Aggrawal, S. , and. Haritsa, J. R," A framework for high-accuracy privacypreserving mining". *In IEEE International Conference on Data Engineering*, 2005.

[14] Kadampur, M. A, Somayajulu D.V.L.N., "A Noise Addition Scheme in Decision Tree for Privacy Preserving Data Mining", Journal of Computing, Vol 2, Issue 1, January 2010, ISSN 2151-9617.

[15] Wang. L. L..,J.,, Zhang., J. ., ``Wavelet based data perturbation for simultaneous privacy preserving and statistics preserving.," *In Proceedings of IEEE International Conference on Data Mining workshop*., 2008.

[16] Aggrwal C. C..,Philip S Yu., " Privacy preserving data mining models and Algorithms.", *Springer Science+Business media.,LLC*..2008.

[17] Vaidya J. and Clifton, C., "Privacy-preserving decision trees over vertically partitioned data". In *Proceedings of the 19th Annual IFIP WG 11.3 Working Conference on Data and Applications Security*, Storrs,Connecticut, Springer , 2005.

[18] Liu, L., Kantarcioglu, M. , and Thuraisingham, B. , "Privacy Preserving Decision Tree Mining from Perturbed Data," Proc. 42nd Hawaii Int'l Conf. System Sciences (HICSS '09), 2009.

[19] Fong, P. K., and Jahnke, J. H. W. , "Privacy Preserving Decision Tree Learning Using Unrealized Data Sets" ." *IEEE Transl. on knowledge and data engineering*, vol. 24, no. 2, February2012.

## BIOGRAPHY

Madhusmita Sahu received the Master in Computer Application from Utkal University, Bhubaneswar, India. She is pursuing Master of Technology degree in Information Technology from the College Of Engineering & Technology, Bhubaneswar, India. Since February 2010, she is having around 05 years of teaching and research experience. Her research interests include Operating System, Data structures, Software Engineering, Data Mining and distributed systems.

Debasis Gountia received the Bachelor of Computer Science and Engineering degree from University College of Engineering, Burla, India. He received the Master of Technology degree in Computer Science and Engineering from the Indian Institute of Technology, Kharagpur, India. Since January 2006, he has been a Faculty with the College of Engineering & Technology, Bhubaneshwar, India. He is having around 10 years of teaching and research experience. His research interests include cryptography, data structures, formal language and automata theory, operating system, and distributed systems.

Neelamani Samal received the Bachelor of Computer Science and Engineering degree from Jagannath Institute for Technology & Management, Parlakhemundi , India. He received the Master of Technology degree in Information Technology from the College Of Engineering & Technology, Bhubaneswar, India. Since February 2010, he has been a Faculty with the Gandhi Institute For Education And Technology, Bhubaneshwar, India. He is having around 03 years of teaching and research experience. His research interests include Operating System, Data structures, Software Engineering and distributed systems.