International Journal of Modern Engineering Research (IJMER)
www.ijmer.com Vol.3, Issue.2, March-April. 2013 pp-809-815 ISSN: 2249-6645

Privacy Preserving On Continuous and Discrete Data Sets- A Novel Approach

Sathya Rangasamy, [1] P.Suvithavani[2]
[1]*M.E Computer Science and Engineering, Sri Shakthi Institute of Engineering and Technology, Coimbatore India*
[2]*Assistant Professor, Department of Computer Science and Engineering, Sri Shakthi Institute of Engineering and Technology, Coimbatore India*

Abstract: *Privacy preservation is important for machine learning and data mining, but measures designed to protect private information often result in a trade-off: reduced utility of the training samples. This introduces a privacy preserving approach that can be applied to decision tree learning, without concomitant loss of accuracy. It describes an approach to the preservation of the privacy of collected data samples in cases where information from the sample database has been partially lost. This approach converts the original sample data sets into a group of unreal data sets, from which the original samples cannot be reconstructed without the entire group of unreal data sets. Meanwhile, an accurate decision tree can be built directly from those unreal data sets. This novel approach can be applied directly to the data storage as soon as the first sample is collected. The approach is compatible with other privacy preserving approaches, such as cryptography, for extra protection.*

Keywords: *Classification, data mining, machine learning, security and privacy protection.*

I. Introduction

Data mining is widely used by researchers for science and business purposes. Data collected (referred to as "sample data sets" or "samples") from individuals (referred as "information providers") are important for decision making or pattern recognition. Therefore, privacy-preserving processes have been developed to sanitize private information from the samples while keeping their utility.

A large body of research has been devoted to the protection of sensitive information when samples are given to third parties for processing or computing [1], [2], [3], [4], [5]. It is in the interest of research to disseminate samples to a wide audience of researchers, without making strong assumptions about their trustworthiness.

Even if information collectors ensure that data are released only to third parties with non-malicious intent (or if a privacy preserving approach can be applied before the data are released, there is always the possibility that the information collectors may inadvertently disclose samples to malicious parties or that the samples are actively stolen from the collectors. Samples may be leaked or stolen anytime during the storing process [6], [7] or while residing in storage [8], [9]. This focuses on preventing such attacks on third parties for the whole lifetime of the samples.

Contemporary research in privacy preserving data mining mainly falls into one of two categories: 1) perturbation and randomization-based approaches, and 2) secure multiparty computation (SMC)-based approaches [10]. SMC approaches employ cryptographic tools for collaborative data mining computation by multiple parties. Samples are distributed among different parties and they take part in the information computation and communication process. SMC research focuses on protocol development [11] for protecting privacy among the involved parties [12] or computation efficiency [13]; however, centralized processing of samples and storage privacy is out of the scope of SMC.

We introduce a new perturbation and randomization-based approach that protects centralized sample data sets utilized for decision tree data mining. Privacy preservation is applied to sanitize the samples prior to their release to third parties in order to mitigate the threat of their inadvertent disclosure or theft. In contrast to other sanitization methods, our approach does not affect the accuracy of data mining results. The decision tree can be built directly from the sanitized data sets, such that the originals do not need to be reconstructed. Moreover, this approach can be applied at any time during the data collection process so that privacy protection can be in effect even while samples are still being collected.

The following assumptions are made for the scope of this paper: first, as is the norm in data collection processes, a sufficiently large number of sample data sets have been collected to achieve significant data mining results covering the whole research target. Second, the number of data sets leaked to potential attackers constitutes a small portion of the entire sample database. Third, identity attributes (e.g., social insurance number) are not considered for the data mining process because such attributes are not meaningful for decision making. Fourth, all data collected are dis-cretized; continuous values can be represented via ranged-value attributes for decision tree data mining.

II. Related Work

In Privacy Preserving Data Mining: Models and Algorithms [14], Aggarwal and Yu classify privacy preserving data mining techniques, including data modification and crypto-graphic, statistical, query auditing and perturbation-based strategies. Statistical, query auditing and most crypto-graphic techniques are subjects 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 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. Dataset Complementation Approach

In Dataset Complementation approach, Unrealized training set algorithm is used. Traditionally, a training set, T_S, is constructed by inserting sample data sets into a data table. However, a data set complementation approach, requires an extra data table, T^P. T^P is a perturbing set that generates unreal data sets which are used for converting the sample data into an unrealized training set, $T^{'}$. The algorithm for unrealizing the training set, T_S, is shown as follows:

Algorithm Unrealize-Training-Set (T_S, T^U, T', T^P)
Input: T_S, a set of input sample data sets
T^U, a universal set
T', a set of output training data sets
T^P , a perturbing set
Output: <T', T^P >

1. if T_S is empty then return <T',T^P >
2. t ←a dataset in T_S
3. if T is not an element of T^P or T^P={t} then
4. $T^P \leftarrow T^P + T^U$
5. $T^P \leftarrow T^P - \{t\}$
6. t' ← the most frequent dataset in T^P
7. return Unrealize-Training-Set (T_S-{t}, T^U , T'+{t'} , T^P-{t'})

To unrealize the samples, T_S, we initialize both T' and T^P as empty sets, i.e., we invoke the above algorithm with Unrealize-Training-Set ($T_{S,\ TU}$,{},{}). The resulting unrealized training set contains some dummy data sets excepting the ones in T_S. The elements in the resulting data sets are unreal individually, but meaningful when they are used together to calculate the information required by a modified ID3 algorithm.

IV. Decision Tree Generation

The well-known ID3 algorithm [18] shown above builds 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

$$H_{a_i}(T_S) = -\sum_{e \in K_i} \left(\frac{\left| T_{S_{(a_i=e)}} \right|}{|T_S|} \right) \log_2 \left(\frac{\left| T_{S_{(a_i=e)}} \right|}{|T_S|} \right)$$

and

$$H_{a_i}(T_S/a_j) = \sum_{f \in K_j} \left(\frac{\left|T_{S_{(a_j=f)}}\right|}{|T_S|} \right) H_{a_i}\left(T_{S_{(a_j=f)}}\right)$$

Where K_i and K_j are the sets of possible values for the decision attribute, a_i, and test attribute, a_j, in T_S, respectively, and the algorithm Majority-Value retrieves the most frequent value of the decision attribute of T_S.

Algorithm Generate-Tree*(T_S, attribs, default)*
Input: T_S, the set of training data sets
attribs, set of attributes
default, default value for the goal predicate
Output: tree, a decision tree
1. **if** T_S is empty **then return** default
2. default ←Majority- Value(T_S)
3. **if** $H_{a_i}(T_S) = 0$ **then return** default
4. **else if** *attribs* is empty **then return** default
5. **else**
6. best← Choose-Attribute(attribs; T_S)
7. tree← a new decision tree with root attribute best
8. **for** each value v_i of best **do**
9. T_{S_i} ←{datasets inT_S as best =K_i}
10. subtree ←Generate-Tree(T_{S_i}; attribs-best,default)
11. connect *tree* and *subtree* with a branch labelled K_i
12. return tree

Already, we discussed an algorithm that generates an unrealized training set, T', and a perturbing set, T^P, from the samples in T_S. In this section, we use data tables T' and T^P as a means to calculate the information content and information gain of TS, such that a decision tree of the original data sets can be generated based on T' and T^P.

4.1 Information Entropy Determination

From the algorithm Unrealize-Training-Set, it is obvious that the size of T_S is the same as the size of T'. Furthermore, all data sets in (T' + T^P) are based on the data sets in T^U, excepting the ones in T_S, i.e., T_S is the q-absolute complement of (T' + T^P) for some positive integer q. The size of qT^U can be computed from the sizes of T' and T^P , with qT^U = 2* |T'|+|T^P|. 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.

4.2 Modified Decision Tree Generation Algorithm

As entropies of the original data sets, T_S, can be determined by the retrievable 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. Generate-Tree' (size, T', T^P, attribs, default)
Input: *size*, size of qT^U
T', the set of unreal training data sets
T^P , the set of perturbing data sets
attribs, 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. *default* ←Minority_Value(T'+T^P)
3. if $H_{a_i}(q[T' + T^P]^c) = 0$ then return default
4. **else if** attribs is empty **then return** default
5. **else**
6.*best*←Choose-attribute'(attribs,Size,(T',T^P))
7.*tree*←a new decision tree with root attribute *best*
8. *size*← *size*=number of possible values k_i in best
9. **for** each value v_i of best **do**
10. T_i'= {data sets in T' as best= k_i}
11. T_i^P={data sets in T^P as best=k_i}
12. *subtree*← Generate-Tree(*size*, T_i', T_i^P, *attribs-best*, default)

13. connect *tree* and *subtree* with a branch labelled k_i
14. **return** *tree*

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 $| qT^U |$ (which equals $2* |T'|+|T^P|$), 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 (*$2* |T'|+|T^P|$, T_S, T^U, A- a_d, k_d). The resulting decision tree of our new ID3 algorithm with unrealized sample inputs is the same as the tree generated by the traditional ID3 algorithm with the original samples

4.3 Data Set Reconstruction

Section B 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, followed by an application of the conventional ID3 algorithm for generating the decision tree from T. The reconstruction process is dependent upon the full information of T' and T^P (whereas $q =2* |T'|+|T^P|/|T^U|$); reconstruction of parts of T_S based on parts T' and T^P is not possible.

4.4 Enhanced Protection with Dummy Values

Dummy values can be added for any attribute such that the domain of the perturbed sample data sets will be expanded while the addition of dummy values will have no impact on T_S. Dummy represents a dummy attribute value that plays no role in the data collection process. In this way we can keep the same resulting decision tree (because the entropy of T_S does not change) while arbitrarily expanding the size of T^U. Meanwhile, all data sets in T' and T^P , including the ones with a dummy attribute value, are needed for determining the entropies of $(q[T' + T^P]^c)$ during the decision tree generation process.

4.5 C5.0 algorithm

In the proposed algorithm, consider C5.0 Algorithm for data mining. The enhancement and the optimization of the C4.5 emerge as algorithm C5.0, which exhibits the better performance as compared to the other existing mining algorithms. C5.0 algorithm to build either a decision tree or a rule set. A C5.0 model works by splitting the sample based on the field that provides the maximum information gain. Each sub sample defined by the first split is then split again, usually based on a different field, and the process repeats until the sub samples cannot be split any further. Finally, the lowest-level splits are re-examined, and those that do not contribute significantly to the value of the model are removed or pruned. C5.0 can produce two kinds of models. A decision tree is a straightforward description of the splits found by the algorithm. Each terminal (or "leaf") node describes a particular subset of the training data, and each case in the training data belongs to exactly one terminal node in the tree.

In contrast, a rule set is a set of rules that tries to make predictions for individual records. Rule sets are derived from decision trees and, in a way, represent a simplified or distilled version of the information found in the decision tree. Rule sets can often retain most of the important information from a full decision tree but with a less complex model. Because of the way rule sets work, they do not have the same properties as decision trees. The most important difference is that with a rule set, more than one rule may apply for any particular record, or no rules at all may apply. If multiple rules apply, each rule gets a weighted "vote" based on the confidence associated with that rule, and the final prediction is decided by combining the weighted votes of all of the rules that apply to the record in question. If no rule applies, a default prediction is assigned to the record. It was introduced an alternative formalism consisting of a list of rules of the form "if A and B and C and ... then class X", where rules for each class are grouped together. A case is classified by finding the first rule whose conditions are satisfied by the case; if no rule is satisfied, the case is assigned to a default class. Each case belongs to one of a small number of mutually exclusive classes. Properties of every case that may be relevant to its class are provided, although some cases may have unknown or non-applicable values for some attributes. C5.0 can deal with any number of attributes. Rule sets are generally easier to understand than trees since each rule describes a specific context associated with a class. Furthermore, a rule set generated from a tree usually has fewer rules than the tree has leaves, another plus for comprehensibility. Another advantage of rule set classifiers is that they are often more accurate predictors than decision trees.

C5.0 decision tree is constructed using *GainRatio*. *GainRatio* is a measure incorporating entropy. Entropy (*E(S)*,) measures how unordered the data set is. It is denoted by the following equation when there are classes $C_1 \ldots C_N$ in data set *S* where $P(S_c)$ is the probability of class *C* occurring in the data set *S*:

$$E(S) = -\sum_{c=1}^{N} P(S_c) * \log_2 P(S_c)$$

Information Gain is a measure of the improvement in the amount of order.

$$Gain(S,V) = E(S) - \sum_{Values(V)} (S_v|S) * E(S_v)$$

Gain has a bias towards variables with many values that partition the data set into smaller ordered sets. In order to reduce this bias, the entropy of each variable over its *m* variable values is calculated as *SplitInfo: GainRatio* is calculated by dividing *Gain* by *SplitInfo* so that the bias towards variables with large value sets is dampened.

$$Gain(S,V) = \frac{Gain(S,V)}{SplitInfo(S,V)}$$

C5.0 builds a decision tree greedily by splitting the data on the variable that maximizes gain ratio. A final decision tree is changed to a set of rules by converting the paths into conjunctive rules and pruning them to improve classification accuracy.

V. Theoretical Evaluation

This section provides a concise theoretical evaluation of our approach. For full details on our evaluation process, we refer to [19].

5.1 Privacy Issues

Private information could potentially be disclosed by the leaking of some sanitized data sets, T_L (a subset of the entire collected data table, T_D), to an unauthorized party if

1. The attacker is able to reconstruct an original sample, t_S, from T_L, or
2. If T_L (a data set in T_L) matches t_S (a data set in T_S) by chance

In the scope of this paper, t_S is non reconstructable because $|T_L|$ is much smaller than $|T'+T^P|$. Hence, we are focusing on the privacy loss via matching. Without privacy preservation the collected data sets are the original samples. Samples with more even distribution (low variance) have less privacy loss, while data sets with high frequencies are at risk. The data set complementation approach solves the privacy issues of those uneven samples. This approach converts the original samples into some unrealized data sets[$T' + T^P$], such that the range of privacy loss is decreased. Data Set complementation is in favour of those samples with high variance distribution, especially when some data sets have zero counts. However, it does not provide significant improvement for the even cases.

Adding dummy attribute values effectively improves the effectiveness of the data set complementation approach; however, this technique requires the storage size of $cR|T^U| - |T_S|$, where c is the counts of the most frequent data set in T_S. The worst case storage requirement equals $(R|T^U|-1)*|T_S|$.

VI. Experiments

This section shows the experimental samples of data's from the data set complementation approach,

1. normally distributed samples and evenly distributed samples
2. extremely unevenly distributed samples
3. Six sets of randomly picked samples, where (i) was generated without creating any dummy attribute values and (ii) was generated by applying the dummy attribute technique to double the size of the sample domain.

For the artificial samples (Tests 1-3), we will study the output accuracy (the similarity between the decision tree generated by the regular method and by the new approach), the storage complexity (the space required to store the unrealized samples based on the size of the original samples) and the privacy risk (the maximum, minimum, and average privacy loss if one unrealized data set is leaked).

6.1 Output Accuracy

In all cases, the decision tree(s) generated from the unrealized samples (by algorithm Generate-Tree' described in Section 4.2) is the same as the decision tree(s), Tree T_S, generated from the original sample by the regular method. This result agrees with the theoretical discussion mentioned in Data Set Complementation approach

6.2 Storage Complexity

From the experiment, the storage requirement for the data set complementation approach increases from $|T_S|$ to $(2|T^U| -1)^*|T_S|$, while the required storage may be doubled if the dummy 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 originals. The worst case happens the samples are distributed extremely unevenly. Based on the randomly picked tests, the storage requirement for our approach is less than five times (without dummy values) and eight times (with dummy values, doubling the sample domain) that of the original samples.

6.3 Privacy Risk

Without the dummy attribute values technique, the average privacy loss per leaked unrealized data set is small, except for the even distribution case (in which the unrealized samples are the same as the originals). By doubling the sample 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 dummy values are used.

VII. Conclusion

We introduced a new privacy preserving approach via data set complementation which confirms the utility of training data sets for decision tree learning. This approach converts the sample data sets, T_S, into some unreal data sets ($T'+ T^P$) such that any original data set is not reconstructable if an unauthorized party were to steal some portion of ($T'+ T^P$) .Meanwhile, there remains only a low probability of random matching of any original data set to the stolen data sets, T_L. The data set complementation approach ensures that the privacy loss via matching is ranged from 0 to $|T_L|*(|T_S|/|T^U|)$, where T^U is the set of possible sample data sets. By creating dummy attribute values and expanding the size of sample domain and the privacy loss via matching will be decreased.

Privacy preservation via data set complementation fails if all training data sets are leaked because the data set reconstruction algorithm is generic. Therefore, further research is required to overcome this limitation. As it is very straightforward to apply a cryptographic privacy preserving approach, such as the (anti)monotone framework, along with data set complementation, this direction for future research could correct the above limitation. This covers the application of this new privacy preserving approach with the ID3 decision tree learning algorithm and discrete-valued attributes only. In proposed approach, we can develop the application with the help of algorithm, C5.0, and data mining methods with mixed discretely—and continuously valued attributes. The storage size of the unrealized samples, the processing time when generating a decision tree from those samples and privacy can be increased using C5.0 algorithm for both continuous and discrete data sets. When compared with the existing Modified ID3 algorithm, proposed method provide the better results.

REFERENCES

[1] S. Ajmani, R. Morris, and B. Liskov, "A Trusted Third-Party Computation Service," Technical Report MIT-LCS-TR-847, MIT, 2001.
[2] S.L. Wang and A. Jafari, "Hiding Sensitive Predictive Association Rules," Proc. IEEE Int'l Conf. Systems, Man and Cybernetics, pp. 164-169, 2005.
[3] R. Agrawal and R. Srikant, "Privacy Preserving Data Mining,"Proc. ACM SIGMOD Conf. Management of Data (SIGMOD '00),pp. 439-450, May 2000.
[4] Q. Ma and P. Deng, "Secure Multi-Party Protocols for Privacy Preserving Data Mining," Proc. Third Int'l Conf. Wireless Algorithms, Systems, and Applications (WASA '08), pp. 526-537, 2008.
[5] J. Gitanjali, J. Indumathi, N.C. Iyengar, and N. Sriman, "A Pristine Clean Cabalistic Fortuity Strategize Based Approach for Incremental Data Stream Privacy Preserving Data Mining," Proc. IEEE Second Int'l Advance Computing Conf. (IACC), pp. 410-415, 2010.
[6] N. Lomas, "Data on 84,000 United Kingdom Prisoners is Lost," Retrieved Sept. 12, 2008, http://news.cnet.com/8301-1009_3-10024550-83.html, Aug. 2008.
[7] BBC News Brown Apologises for Records Loss. Retrieved Sept.12, 2008, http://news.bbc.co.uk/2/hi/uk_news/politics/7104945.stm, Nov. 2007.
[8] D. Kaplan, Hackers Steal 22,000 Social Security Numbers from Univ.of Missouri Database, Retrieved Sept. 2008, http://www.scmagazineus.com/Hackers-steal-22000-Social-Security-numbers-from-Univ.-of-Missouri-database/article/34964/, May 2007.
[9] D. Goodin, "Hackers Infiltrate TD Ameritrade client Database," Retrieved Sept. 2008, http://www.channelregister.co.uk/2007/09/15/ameritrade_database_burgled/, Sept. 2007.
[10] L. Liu, M. Kantarcioglu, and B. Thuraisingham, "Privacy Preserving Decision Tree Mining from Perturbed Data," Proc. 42nd Hawaii Int'l Conf. System Sciences (HICSS '09), 2009.
[11] Y. Zhu, L. Huang, W. Yang, D. Li, Y. Luo, and F. Dong, "Three New Approaches to Privacy-Preserving Add to Multiply Protocol and Its Application," Proc. Second Int'l Workshop Knowledge Discovery and Data Mining, (WKDD '09), pp. 554-558, 2009.
[12] J. Vaidya and C. Clifton, "Privacy Preserving Association Rule Mining in Vertically Partitioned Data," Proc Eighth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '02), pp. 23-26, July 2002.
[13] M. Shaneck and Y. Kim, "Efficient Cryptographic Primitives for Private Data Mining," Proc. 43rd Hawaii Int'l Conf. System Sciences (HICSS), pp. 1-9, 2010. [14] C. Aggarwal and P. Yu, Privacy-Preserving Data Mining:, Models and Algorithms. Springer, 2008.
[15] L. Sweeney, "k-Anonymity: A Model for Protecting Privacy," Int'l J. Uncertainty, Fuzziness and Knowledge-based Systems, vol. 10, pp. 557-570, May 2002.
[16] J. Dowd, S. Xu, and W. Zhang, "Privacy-Preserving Decision Tree Mining Based on Random Substitutions," Proc. Int'l Conf. Emerging Trends in Information and Comm. Security (ETRICS '06), pp. 145-159, 2006.
[17] S. Bu, L. Lakshmanan, R. Ng, and G. Ramesh, "Preservation of Patterns and Input-Output Privacy," Proc. IEEE 23rd Int'l Conf.Data Eng., pp. 696-705, Apr. 2007.

[18] S. Russell and N. Peter, Artificial Intelligence. A Modern Approach 2/E. Prentice-Hall, 2002.
[19] P.K. Fong, "Privacy Preservation for Training Data Sets in Database: Application to Decision Tree Learning," master's thesis, Dept. of Computer Science, Univ. of Victoria, 2008.

AUTHOR BIOGRAPHY

Ms.Sathya Rangasamy received B.E degree in CSE from Avinashilingam University, Coimbatore and currently pursuing M.E degree in Computer Science and Engineering in Sri Shakthi Institute of Engineering and Technology, under Anna University, Chennai. Her research interest includes Computer Networks and Data Mining.

Mrs.Suvithavani.P received M.Sc (Integrated-5Yrs) degree in IT from Government College of Technology, Coimbatore and received her M.E degree from Computer Science and Engineering in Sun College of Engineering and Technology, Nagercoil. She is currently working as Assistant Professor in Department of CSE in Sri Shakthi Institute of Engineering and Technology, Coimbatore. Her main research interest is Data Mining.