www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 5 Issue 5 May 2016, Page No. 16329-16332

Data Restoration and Privacy Preserving of Data Mining Using Random Decision Tree Over Vertically Partitioned Data

Ms.K.Muthukarupaee[1], Ms.Blessyselvam[2] ,Ms.V.Ranjani[3], Ms.N.Radha[4]

[1]M.E Student of Department of Computer Science,

University College of Engineering,BIT Campus,Trichy,India.

muthuballa@gmail.com

[2]Teaching Fellow, Department of Computer Science,

University College of Engineering,BIT Campus,Trichy.

blessyselvam@yahoo.co.in

[3]Assistant Professor, Department of Computer Science,

Saranathan College of Engineering,Trichy.

ranjani-cse@saranathan.ac.in

[4]Assistant Professor, Department of Computer Science,

Saranathan College of Engineering,Trichy.

radha-cse@saranathan.ac.in

Abstract: In recent years with the development of network data collection and storage technology, the usage and sharing of large amounts of data has become easy process. Once the data and information is accumulated, its will become the wealth of information. Data mining, otherwise known as knowledge discovery, can extracted meaningful information from the large amounts of data because it supports people's decision- making. However the traditional data mining techniques and algorithms directly operated on the original data set, which will cause the leakage of privacy data, these problems challenge the traditional data mining, so privacy-preserving data mining (PPDM) it has become one of the most newest trends in the privacy, security and data mining research. Existing cryptography is the based work for privacy-preserving data mining and is still too slow to be effective for the large scale. But proposed approach is based on exploit for fact that RDTs can naturally it fit into the parallel and fully distributed architecture.

1. Introduction

Data Mining is quick growing field of the distributed atmosphere and method of discovering the fascinating patterns and knowledge from giant of information. It's additionally known as KDD process that is information Discovery from knowledge. It permits knowledge of analysis and whereas conserving knowledge privacy. Data privacy conserving is forestalling the personal secret or non-public data from a unnecessarily distributed or in the public identified or not to be put upon by person or by the oppose. In privacy preserving data processing is the fascinating and helpful for data is distributed with privacy of guidance has been as preserved. There square measure 2 stages in privacy for conserving knowledge mining initial are knowledge assortment and for second is knowledge commercial enterprise. In data assortment, the knowledge holder stores of knowledge that is gathered by a data owner. In knowledge commercial Enterprise, knowledge may be free to knowledge recipient by knowledge holder and the knowledge recipient mine sprinted secured knowledge. Cryptographic techniques are square measure typically too slow to be sensible and can become a computationally expensive because it rise in size of the info set and it communications between numerous parties increase. Crypto graphical techniques cannot handle a huge data. To overcome this square measure victimization privacy conserving RDT (Random Decision Tree) is used for privacy conserving data processing which is developed by the Fan et al. Privacy conserving RDT Privacy conserving RDT is the combination of randomization and cryptography technique. This resolution provides Associate in nursing order of magnitude improvement inefficiency over existing solutions whereas providing a lot of knowledge privacy and knowledge utility. This can be an efficient resolution to privacy-preserving data processing for the massive knowledge challenge. Random Decision Tree provides higher potency and knowledge privacy than the crypto graphical technique. RDT provides the structural property, a lot of specifically, the very fact that solely specific nodes (the leaves) within the classification tree have to be compelled to the encrypted /decrypted, and secure token for passing prevents adversary from utilizing count techniques to decipher instance classifications, because the branch structure of the tree is hidden from all parties. A random structure provides security against investing priority information to get the whole classification model or instances. Existing cryptography-work based on privacy-preserving data mining and is still too slow to be effective for the large scale. It was implemented using a classification rule for data mining with the privacy preserving. The data was sorted by classification in a class with the similar

features of the other data.It was done by referring the original data or by following the model of data. Classification can be done in two steps the first step is supervised learning by where a classification model is constructed and the second step is a where the model is used to predict class labels for given data or it is used to classify the accuracy of the data. The identified drawbacks:

- Cryptographic techniques are often too slow to be practical and can become computationally expensive.
- It also rise in size of the data set and communications between various parties increase.
- Cryptographic techniques cannot handle big data.

2. Horizontally Partitioned Data

When data is horizontally partitioned and collect data for different entities, but have data for all of the attributes.We now need to figure out how the RDTs can be constructed and how classification is performed. Since for all the parties share the schema, a straight-forward solution is for all parties to independently create a few random trees. Together these will form the ensemble of random trees. However, each party can only independently create the structure of the tree. All parties must co-operatively and securely compute the parameters (i.e., values of each leaf node), over the global data set. Unlike the basic RDT approach, there is no need to keep the class distribution at each non-leaf node—this information is only required at the leaf nodes. Now, there are two possibilities: 1) The structure of the tree is known to each participant. 2) The structure of the tree is unknown to each participant. The global class distribution vector for each leaf node will be known to all parties. The global class distribution vector for each leaf node is known only to the party owning the tree. The global class distribution vector for each leaf node is known to none of the parties.

3. Vertically Partitioned Data

With vertically partitioned data, all parties collect data for the same set of entities. However, each party collects data for a different set of attributes. Now the parties cannot independently create even the structure of a random tree, unless they share the attribute information among each other. Thus, there are two possibilities: All parties share basic attribute information (i.e., metadata). Now they can independently create random trees (at least the structure). There is no sharing of information. Now, the parties need to collaborate to create the random trees. These trees could themselves exist in a distributed form. Unlike the horizontal partitioning case, the structure of the tree does reveal potentially sensitive information, since the parties do not know what are the attributes owned by the other parties. Therefore, we directly address the case of fully distributed trees.

4. Related Work

Random Decision Trees (RDTs) used for both horizontally and vertically partitioned the data sets. The important property of the RDT is that same code can be used for multiple data mining tasks such as classification, regression, ranking and multiple classification .RDT is an efficient implementation of Bayes optimal classifier BOC is effective non-parametric density estimation and can be explained via high order statistics such as moments. It provide better accuracy as well as reduce the computation time compare to RDT by using RSA cryptography algorithm. In privacy preserving data mining Random decision tree algorithm create multiple decision tree randomly. Firstly start with creating attribute lists from the training datasets. Now generate a tree by choosing an attributes randomly.

Here we apply Divide and conquer strategy.

1. We will select best attribute for splitting.
2. For each attribute create new child nodes.
3. For each child nodes
    a. If node is leaf node then stop
    b. Else keep splitting.
    c. End if.

4.1 Advantages

- RDTs can be used to generate the equivalent and data accurate
- It also provide sometimes better models with much smaller cost
- It can give a solid privacy with low computation.
- RDT performs better than the different models with respect to the computation speed, because of the characteristics of random partitioning utilized as a part of tree development.

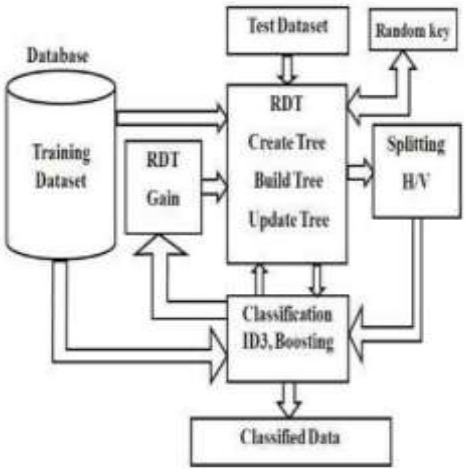

Figure 1: System Architecture

5. Modules Description

5.1 User Registration

The user register module is quite similar to our community builder registration module. It simply is a module that allows for user registration on any page access, it's an exact of the default page .It can allows for login describes the interface that must be implemented by authentication applications.

5.2 Admin Login

This displays the screen where user can enter the data's to the server and attributes to the data and the threshold value. Upon clicking the submit button in this module will add data to the server in any different datasets and admin can also view the user login details.

5.3 Data Owner Login

User can search and capture all the data for different data sets from the server. User can also select the type of search functions and view the specified results

5.4 Search

The Search module uses three different types for users

5.4.1 T-Correlation

T-Correlation module extracts only the relevant label and removes the redundant data from a dataset and displays the results. The irrelevant feature removal is a straightforward once the right relevance measure is defined or it selected, while the redundant feature elimination is a bit of sophisticated. In our proposed algorithm, it involves 1) the construction of the RDT from a weighted complete graph 2) the partitioning of the MST into a forest with each tree of representing a cluster; and 3) the selection of representative features from the clusters. In order to more precisely introduce the algorithm, and because of our proposed feature subset selection framework it involves irrelevant feature removal and redundant feature elimination present the traditional definitions of relevant and redundant features, then provide our definitions bases on variable correlation.

6. System Methodology

6.1 Data Clustering

Data Clustering identifies the clusters or groups for a set of objects whose classes are unknown. Clustering should be a done such that the similarities between objects of the same clusters are maximized and similarities between different clusters are minimized. Clustering is the unsupervised for classification of patterns (observations, data items, or feature vectors) into groups (clusters).

6.2 Multi Label Classification

A multi-label approach that allows the iterative learning algorithm is also a higher order label correlations. An auxiliary compositional label is the defined as a combination of primary labels with the interest of utilizing informative high order correlations. An efficient of data mining method called association rule mining is adopted. An informative correlation can be found from the one association rule and formativeness is measured by the support and confidence of the rule.

7. Random Decision Tree

Random decision tree algorithm constructs multiple is a depth decision trees randomly. RDT is based on two stages, training and classification and structure of a random tree is constructed completely independent of the training data. When constructing each tree the first start with a list of attributes from a data set. Generate a tree by randomly choosing a one of the attributes without using any training data.The tree stops growing once when the height limit is reached. Then use the training data to update the statistics of each node in the tree. In this only the leaf nodes need to record the number of values in different classes that are classified through the nodes in the tree. The training data is scanned exactly once to update the statistics in a multiple random trees. The training phase consists of creating the trees (BuildTreeStructure) and populating the nodes with the training instance of data (UpdateStatistics). It is assumed the number of attributes is known to all parties based on the training data set. The depth of each tree is decided based on a heuristic Fan et al. When the depth of the tree is equal to half of the total number of attributes present in the data, the most diversity is achieved for preserving the advantage of random modeling.

7.1 Build Tree Structure

Build_Tree(tree_level,tree_depth)
1: if ltree_level $\leq$ tree_depth then
2: Randomly choose i from [1 ...j] (uniform random distribution)
3: At B_i: Randomly choose rt from [1 ... n_i] (uniform random distribution)
4: At B_i: Create a Interior Node Nid with attribute Nid.A <— A_{rt} (the rth attribute)
5: if A_{rt} is numeric then
6: At B_i: Randomly choose a split point π from within the range of A_{rt}
7: At B_i: Constraints.set(Nid.. A_{rt}," < π") {Update local constraints tuple}
8: node_ld <— Build_Tree(tree_level + 1, tree_depth)
9: Nid.leftsubtree < node_ld {Add appropriate branch to interior node}
10: At B_i: Constraints.set(Nid. A_{rt}, ">= π") {Update local constraints tuple}
11: node_ld <— Build_Tree(tree_level + 1,tree_ depth)
12: Nid.rightsubtree <— node_ld {Add appropriate branch to interior node}
13: else
14: for each attribute value c_i $\in$ Nid.A do
15: Constraints.set(Nid.A, c_i) {Update local constraints tuple}
16: node_ld <— Build_Tree(tree_level + 1,tree_ depth)
17: Nid. c_i +- node_ld {Add appropriate branch to interior node}
18: end for
19: end if
20: Constraints.set(A_{rt},'?') {Returning to parent: should no longer filter transactions with Ar }
21: Store Nid locally keyed by Node _ID
22: return Node _ID of interior node Nid {Execution continues at site owning parent node}
23: else
24: for i = 1...n do
25: Randomly choose r from the range of random numbers for the cryptographic system
26: $level_i$ <- Encrypt(0, rt) {Random encryption of 0}
27: end for
28: Create leaf node Nd with p values 1v1,..., iVp
29: return Node ID of leaf node Nd
30: end if

7.2 Update Statistics

UpdateStatistics()
Require: Transaction set T partitioned vertically between sites Pv_1 ,…. , Pv_k
Require: Pv_u, holds k_i, attributes
Require: pv class values, e_1,……,e_p, with Pv_k holding the class attribute
Require: s, the number of random trees built (let $node\_id_i$ represent the root node of the jth tree)
1: for each instance n_x do
2: {At Pv_k, build the class vector}
3: for i = 1 ...p do

4: Randomly choose r from the set of positive integers
5: if n_x has class e_i then
6: lv_i <- Encrypt(1,r) {Random encryption of 1}
7: else
8: lv_i <- Encrypt(0,r) {Random encryption of 0}
9: end if
10: end for
11: for j = 1 ... s do
12: incrementStats(n_x,node_Id_j, lv_i)
13: end for
14: end for

8. Computational Cost

Since we are particularly interested in comparing against cryptographic algorithms, we only count the number of cryptographic operations (encryption, decryption, exponentiation). In general, the non-cryptographic operations do not incur much computation overhead as compared to the cryptographic operations, so their overhead can be ignored. For horizontally partitioned data, only the leaf nodes are encrypted, by each party. Since the depth of a tree is n=2 (assuming binary splits), there are 2 n=2_1 With p classes, m trees, and k parties, there are a total of Oðpmk2 n=2_1 Þ encryptions. Classifying a new instance requires homomorphic multiplications and p threshold decryptions. For vertically partitioned data, all leaf nodes are originally encrypted and then updated. Classifying a new instance requires p threshold decryptions.

9. Conclusion and Future Work

RDTs can be used to generate the equivalent, accurate and sometimes better models with much smaller cost. we using distributed privacy- preserving RDTs. Our approach leverages the fact that randomness in structure can provide strong privacy with less computation. In the future, we plan to develop general solutions that can work for arbitrarily partitioned data and overlapping transaction

Privacy preserving RDTs framework provides better accuracy with security and efficiency than boolean with privacy preserving framework. It can be used to generate accurate and sometimes better models with much smaller cost. It enhanced the performance of RDT with privacy preserving framework. It can efficiently handle distributed data than RDT with privacy preserving framework. It can reduce computational time than RDT with privacy preserving framework. Privacy preserving ARDT produces a highly accurate classifier and learning is fast. Privacy preserving ARDT is growing tree so it grows as instances are created.

In the future work, plan to generate general solutions system that can work for arbitrarily partitioned data and overlapping transaction.

References

[1] Jaideep Vaidya, Senior Member, IEEE, Basit Shafiq, Member, IEEE, Wei Fan, Member, IEEE, Danish Mehmood, And David Lorenzi "A Random Decision Tree Framework Or Privacy-Preserving Data Mining" Proc. IEEE Transactions On Dependable And Secure Computing, Vol. 11, No. 5, September/October 2014

[2] J. Vaidya, C. Clifton, and M. Zhu, Privacy-Preserving Data Mining.ser. Advances in Information Security first ed., vol. 19, Springer-Verlag, 2005.

[3] W. Fan, H. Wang, P.S. Yu, and S. Ma, "Is Random Model Better? On Its Accuracy and Efficiency," Proc. Third IEEE Int'l Conf. Data Mining (ICDM '03), pp.51-58, 2003.

[4] W. Fan, J. McCloskey, and P. S. Yu, "A General Framework for Accurate and Fast Regression by Data Summarization in Random Decision Trees," Proc. 12th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '06), pp. 136-146, 2006.

[5] X. Zhang, Q. Yuan, S. Zhao, W. Fan, W. Zheng, and Z.Wang,"Multi-Label Classification without the Multi- Label Cost," Proc. SIAM Int'l Conf. Data Mining (SDM'10), pp. 778-789, 2010.

[6] A. Dhurandhar and A. Dobra, "Probabilistic Characterization of Random Decision Trees," J. Machine Learning Research, vol. 9, pp. 2321-2348,2008.

[7] G. Jagannathan, K. Pillaipakkamnatt, and R.N. Wright, "A Practical Differentially Private Random Decision Tree Classifier," Proc. IEEE Int'l Conf. Data Mining Workshops (ICDMW '09), pp. 114-121,2009