

An Un-realized Algorithm for Effective Privacy Preservation Using Classification and Regression Trees

Satyanarayana Murthy Teki^{1*}, Balaji Banothu², Mohan Krishna Varma¹

¹ Madanapalle Institute of Technology & Science, Madanapalle 517325, India

² Vignan's Foundation for Science Technology & Research (Deemed to be university), Guntur 522213, India

Corresponding Author Email: murthyts@mits.ac.in

<https://doi.org/10.18280/ria.330408>

ABSTRACT

Received: 16 March 2019

Accepted: 22 July 2019

Keywords:

privacy, privacy preservation, decision tree, perturbation, un-realization, classification, regression

In this digital age, privacy preservation has attracted much attention, as a huge amount of data are generated from multiple sources and transmitted across the Internet. Several perturbation algorithms have emerged to keep sensitive data hidden behind additive noises. In this paper, a novel un-realization algorithm is developed based on a classification and regression tree (CART). First, the sample dataset was distorted, and the duplicate elements were removed, creating a perturbed dataset and an un-realized dataset. Then, a decision tree was set up by the modified CART algorithm and another by the traditional CART based on the un-realized dataset. Finally, the Gini values of the two trees were compared. If the results are the same, then the privacy of the data is preserved. The proposed algorithm was compared with several traditional un-realization algorithms through experiments. The results show that our algorithm achieved excellent results in Gini value, time complexity and output accuracy.

1. INTRODUCTION

Agrawal hide the sensitive information through an un-realization algorithm leading to a new research area called privacy preserving data mining (PPDM) [1-5]. The techniques used in it are broadly categorized into perturbation and randomization, data modification and secure multiparty computation (SMC) approaches. Rakesh proposed a perturbation and randomization approach to distort the original dataset and reconstruct it later from the distorted values at the cost of considerable computation time [6]. Sweeney proposed a k -Anonymity model to resolve identity disclosure using generalization and suppression techniques, but are prone to sensitive and background knowledge attacks [7]. A secure multi party computation approach using cryptography techniques that preserved sensitive information among the multiple parties, but at the cost of high computational complexity [8-9]. Lindell used ID3 algorithm for achieving SMC. Similarly, Kisilevich proposed a KATCUS algorithm based on ID3 decision tree approach [10]. Fong proposed an un-realization technique for sanitization of the dataset and an ID3 algorithm and its modifications are used for privacy preserving decision tree learning [11-13]. ID3 algorithmic approach took huge computational time in construction of decision tree. Satya proposed un-realization algorithms for association rule hiding [14-24]. In this manuscript, a modified un-realization algorithm and a modified classification and regression tree based algorithms are proposed for privacy preserving decision tree learning to reduce the computational time in decision tree construction process. Apart from algorithmic solution architecture has also been proposed to convert the sample dataset into secured and sanitized version, so that private and sensitive information is hidden from unauthorized users. Finally, faster decision tree obtained from the un-realized samples using Modified CART is found to be

better than the traditional algorithm.

The rest of the paper has been organized into 9 sections. The set theory pertaining to dataset complementation approach is discussed in Section 2. The architecture is depicted in Section 3. The un-realization algorithm basics are elaborately discussed in Section 4. The proposed work and introduction to CART algorithm discussed in Section 5, 6 and 7. The results are validated and verified in Section 8.

2. DATASET COMPLEMENTATION APPROACH

Following definitions are used in dataset complementation approach from Fong et al. [12].

Definition 1: State Space Z , a State Space of data table T , contains all the instances in a data table T . If a table T has attributes $X = \{A, B\}$ and $Y = \{C, D\}$ then State Space Z is simply their Cartesian product of two attributes given by $\{(A, C), (A, D), (B, C), (B, D)\}$. If a table has attributes a_1, a_2, \dots, a_i and if takes an a_j (where $j=1, \dots, i$) values v_1, v_2, \dots, v_n then

$$|Z| = v_1 * v_2 * \dots * v_m \quad (1)$$

Definition 2. Let p be an integer and if a subset of data table T is represented as A_D then p -multiple-of A_D is denoted as pA_D . also,

$$|pZ| = p * |Z|. \quad (2)$$

Definition 3. Attribute c contains the value of m and Attribute e contains the value of d in Z , where $Z_{(c=m)}$ specifies a subset of Z that contains all Datasets with attribute c equals m .

Theorem 1. If m_i is a possible value of attribute c_i in Z , then

$$|pZ_{(c_i=m_i)}| = (p \times y_1 \times y_2 \times \dots \times y_m) / y_i. \quad (3)$$

Proof. $|pZ_{(c_i=m_i)}| = q \times |Z_{(c_i=m_i)}|$
 $= p \times y_1 \times y_2 \times \dots \times y_{i-1} \times y_{i+1} \times \dots \times y_m$
 $= (p \times y_1 \times y_2 \times \dots \times y_m) / y_i.$
 $|pZ_{(c_i=m_i) \dots (c_j=m_j)}|$
 $= (p \times y_1 \times y_2 \times \dots \times y_m) / (y_i \times \dots \times y_j). \quad (4)$

Definition 4. If Z_V is a subset of Z , then the absolute complement of Z_V , denoted as Z_D^k , is equal to $Z - Z_V$, and a s-absolute-complement of Z_V , denoted as sZ_D^k , is equal to $sZ - Z_V$. If Z_{A1} and Z_{A2} are two sets of datasets that are subsets of Z .

$$|Z_{A1} - Z_{A2}| = |Z_{A1}| - |Z_{A2}| \quad (5)$$

Definition 5. If b_i and b_j are possible values of attributes c_i and c_j in Z , respectively, then

$$|[Z_{A1} - Z_{A2}]_{(c_i=b_i)}| = |Z_{A1(c_i=b_i)}| - |Z_{A2(c_i=b_i)}| \quad (6)$$

and

$$|[Z_{A1} - Z_{A2}]_{(c_i=b_i) \dots (c_j=b_j)}| = |Z_{A1(c_i=b_i) \dots (c_j=b_j)}| - |Z_{A2(c_i=b_i) \dots (c_j=b_j)}| \quad (7)$$

Definition 6. If $W_S = g[W^l + W^h]^C$ and $|W^l| = |W^S|$ then

$$|gW^F| = 2 * |W^l| + |W^h|. \quad (8)$$

Proof. $W_S = g[W^l + W^h]^C$
 $W_S = gW^F - W^l + W^h$
 $|W_S| = |gW^F - (W^l + W^h)|$
 $|W_S| = |gW^F| - |(W^l + W^h)|, \text{ (i.e. } (W^l + W^h) \subseteq gW^F)$
 $|W_S| = |gW^F| - |W^l| - |W^h|$
 $|W^l| = |gW^F| - |W^l| - |W^h|, \text{ (i.e. } |W^l| = |W_S|)$
 $|gW^F| = 2 * |W^l| + |W^h|.$

3. PROPOSED ARCHITECTURE

The architecture of a modified un-realization algorithm for privacy preserving decision tree learning using classification and regression tree is shown in Figure 1. This architecture is broadly classified into distortion, computational and analysis phases. The first phase deals with distortion phase where the sample dataset D_s subjected to distortion through modified un-realization algorithm. In this phase all the duplicate data elements are removed and as a result perturbed dataset D_p and an un-realized dataset D_u are generated. During the computational phase, decision tree obtained from the un-realized samples by a modified CART algorithm is proved to be same as the decision tree constructed from the traditional CART algorithm. In the last phase, the decision trees and their Gini values of both the algorithms are compared, and proved to be same. So that, a modified un-realization algorithm for privacy preserving decision tree learning using classification and regression trees has been achieved.

Proposed Architecture:

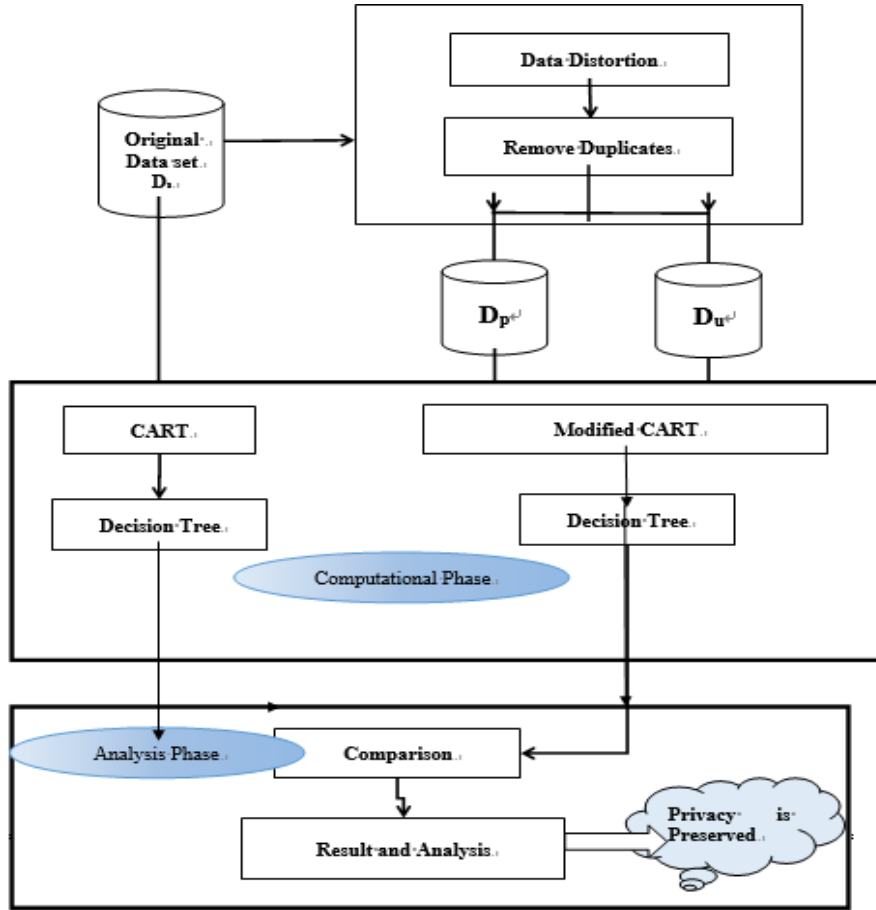


Figure 1. Proposed architecture

4. UN-REALIZATION ALGORITHM

Un-realization algorithm requires a sample dataset D_s and a State Space P_u are the inputs to the un-realization algorithm.

Definition 1 is used to generate a State Space P_u set. **Definition 2,3,4,5,7,8** consist of equations for a dataset complementation approach to produce perturbed dataset D_p and un-realized dataset D_u . The reconstruction of the original dataset is possible with the combination of $D_p + D_u$. In this algorithm T denotes the data element or a tuple in the Sample Dataset D_s . T^* denotes data element or a tuple other than T in the perturbing dataset D_p .

Un-realization Algorithm:

Un-realization Algorithm(D_s, P_u, D_u, D_p)

Input: { D_s _Sample Dataset, P_u _State Space, D_u _Unrealized set, D_p _perturbing set}

Output : { D_u _Unrealized set, D_p _perturbing set }

Unrealized Algorithm(D_s, P_u, D_u, D_p)

if(D_s _Sample_Dataset==null){

return{ D_u _Unrealized_set, D_p _perturbing_set }

}

$T1$ =a data element of D_s _Sample_Dataset;

if($T1$.isElementOf(D_p _perturbing_set){ }

else{

D_p _perturbing_set= D_p _perturbing_set+
 D_u _unrealized_set - $T1$;

$T1^1$ =a data element of D_p _perturbing_set

}

return Unrealized_Algorithm(D_s _Sample_Dataset- $T1$,
 P_u _State Space, D_u _Unrealized set+ $T1^1$, D_p _perturbing set-
 $T1^1$)

return{ D_u _Unrealized_set, D_p _perturbing_set }

Definition 7: If T_{d1} and T_{d2} are subsets then T_{d1}/T_{d2} represented as $T_{d2}-T_{d1}$. The relative complement of T_{d1}/T_{d2} given as $T_{d1}-T_{d2}$. The absolute complement of S_d is S_d^c . Equations for all the possible values of q are given below,

$$\begin{aligned} S_d^c &= P_u - S_d; \\ S_d &= P_u - S_d^c; \\ qS_d^c &= qP_u - S_d \\ qP_u &= qS_d^c + S_d. \end{aligned}$$

The Iterations of Un-realized algorithm are shown below.

Iteration-1: Initial values are $D_s=7; P_u=12; D_u=0; D_p=0$;

Unrealized_Algorithm(D_s, P_u, D_u, D_p) {

$T1$ ={sunny, weak, no}

$D_p + P_u - T1 = 0 + 12 - 1 = 11$;

$T1^1$ ={sunny, weak, yes}

return Unrealized_Algorithm(6, 12, 1, 10);

}

Iteration-2: Initial values are $D_s=6; P_u=12; D_u=1; D_p=10$;

Unrealized_Algorithm(D_s, P_u, D_u, D_p) {

$T1$ ={sunny, strong, no}

$D_p - T1 = 10 - 1 = 9$;

$T1^1$ ={sunny, strong, yes}

return Unrealized_Algorithm(5, 12, 2, 8);

}

Iteration-3: Initial values are $D_s=5; P_u=12; D_u=2; D_p=8$;

Unrealized_Algorithm(D_s, P_u, D_u, D_p) {

$T1$ ={overcast, weak, yes}

$D_p - T1 = 8 - 1 = 7$;

$T1^1$ ={sunny, weak, yes}

return Unrealized_Algorithm(4, 12, 3, 6);

}

Iteration-4: Initial values are $D_s=4; P_u=12; D_u=3; D_p=6$;

Unrealized_Algorithm(D_s, P_u, D_u, D_p) {

$T1$ ={rain, weak, yes}

$D_p - T1 = 6 - 1 = 5$;

$T1^1$ ={overcast, strong, yes}

return Unrealized_Algorithm(3, 12, 4, 4);

}

Iteration-5: Initial values are $D_s=3; P_u=12; D_u=4; D_p=4$;

Unrealized_Algorithm(D_s, P_u, D_u, D_p) {

$T1$ ={rain, weak, yes}

$D_p + P_u - T1 = 4 + 12 - 1 = 15$;

$T1^1$ ={overcast, strong, no}

return Unrealized_Algorithm(2, 12, 5, 14);

}

Iteration-6: Initial values are $D_s=2; P_u=12; D_u=5; D_p=14$;

Unrealized_Algorithm(D_s, P_u, D_u, D_p) {

$T1$ ={rain, strong, no}

$D_p - T1 = 12 - 1 = 11$;

$T1^1$ ={rain, weak, no}

return Unrealized_Algorithm(1, 12, 6, 12);

}

Iteration-7: Initial values are $D_s= 1; P_u=12; D_u=6; D_p=12$;

Unrealized_Algorithm(D_s, P_u, D_u, D_p) {

$T1$ ={overcast, strong, yes}

$D_p - T1 = 12 - 1 = 11$;

$T1^1$ ={rain, strong, yes}

return Unrealized_Algorithm(0, 12, 7, 10);

5. MODIFIED UN-REALIZATION ALGORITHM

As it is known in the existing algorithm, the size of perturbed dataset size is large and it also allows many duplicate records, due to high computational complexity involved in un-realizing the fact in Fong et al. [12]. A modified un-realization algorithm is proposed for removing duplicates from the perturbed dataset. The size of the perturbed dataset seems to be diminished when compared to the existing Un-realization algorithm.

Modified Algorithm:

Modified_Unrealized_Algorithm(D_s, P_u, D_u, D_p)

Input : { D_s _Sample Dataset, P_u _Power_set,

D_u _Unrealized_set, D_p _perturbing_set}

Output: { D_u _Unrealized set, D_p _perturbing set }

Modified_Unrealized Algorithm(D_s _Sample Dataset,

P_u _State Space, D_u _Unrealized set, D_p _perturbing set){

if(D_s _Sample_Dataset==null)

{

return{ D_u _Unrealized_set, D_p _perturbing_set }

}

$T1$ =a data element of D_s _Sample_Dataset

if($T1$.isElementOf(D_p _perturbing_set)

{

D_p _perturbing_set= D_p _perturbing_set- $T1$;

$T1^1$ =a data element of D_p _perturbing set;

if($T1^1$.Exists_Multiple)

{

$T1^1 = T1^1$;

}}

else{

```

    Dp_perturbing_set= Dp_perturbing_set+
    Du_unrealized_set-T1;
    T1l=a data element of Dp_perturbing set
    if(T1l.Exists_Multiple)
    {
        T11l=T1l;
    }
}
return Unrealized_Algorithm(Ds_Sample_Dataset- T1,
Pu_State_Space,Du_Unrealized_set+ T1l,Dp_perturbing_set-
T1l- T11l)
return {Du_Unrealized_set,Dp_perturbing_set}
}

```

In the flowchart, T denotes the Data element or a Tuple in the Sample Dataset D_s. T* denotes Data element or a Tuple other than T in the Perturbing Dataset D_p. T** denotes the Duplicate Data element in D_p.

6. CART ALGORITHM

To our knowledge, CART has become an alternative for privacy preserving decision tree learning because of its dual role i.e. it can be used as classification tree as well as regression tree based upon the data. It is favorable for many researchers because it can handle huge continuous variables; outlier's etc. It uses Gini Index so that mathematical complexity has been reduced. Breiman et al. [4] developed CART algorithm which uses a set of historical data with pre-assigned classes. CART identifies which variable to be used for best split, precise rule identification and assigning class labels to terminal nodes.

Algorithm: *Tree_cart (P_i, attribute, default)*

Input: P_i, A training data

attribute, set of attributes
default, goal predicate.

Output: A decision tree

Tree_cart (P_i, attribute, default)

```

if Pi is empty { return default
}
Majority – value( Pi ) -> default
if Gini (Pi) =0{
    return default
}
else if attribute is empty{ return default
}
else{
    Attribute_select (attribute, Pi) -> good
    A tree with good as root attribute ->tree
    for each value ai of good do
        datasets in Pi as good=bi -> Pi(i)
        Tree_cart (Pi(i), attribute_good, default) ->
        subtree
        connect tree & subtree with a branch bi
}
return tree

```

Initially sample dataset D_s and default value D as inputs. Let G be the Gini Index and max be the maximum that can be stored for each iteration of attribute A_i. Whenever the training set and Gini index will be zero then it returns default value. Calculate Gini index and construct decision tree for both

algorithms.

7. MCART ALGORITHM

Modified CART algorithm also used for privacy preserving decision tree learning based on Gini Index, so that mathematical complexity has been reduced compared with Entropy based tree construction. It uses D_p+D_u together to construct the original decision tree. Here *Choose-attribute (size,...)* method plays a vital role in tree construction. As per the definition 6 the reconstruction of decision tree based on size attributes. The performance of the Modified CART with the traditional CART algorithm is assessed in terms of Time Complexity and Output accuracy etc. Hence it is verified that the decision tree obtained from Modified CART is proved to be same as the original decision tree constructed from the traditional CART algorithm as shown in Figure 2.

Algorithm Generate (size,P^l,P^t, attri, default)

Input size, size equals 2* D_p+ D_u

D_u or P^t be the perturbing Dataset

D_p or P^l be the Un-realization Dataset

attri, set of attributes

default, the goal predicate

Output tree, a decision tree

If (P^l, P^t) is empty{

return default

}

default ← Minority - Value (P^l + P^t)

If d*ΔGini (P^l + P^t) =0{

return default

}

else if attri is empty{

return default

}

else{

Good ← choose-attribute (attri, size, (P^l, P^t))

tree ← a new decision tree with root attribute good

Size ← size/number of possible values a_i in good

for each value v_i of good **do**{

P^l_i ← { Datasets in P^l as good = a_i }

P^t_i ← { Datasets in (P^t as good = a_i) }

subtree ← Generate (size, P^l_i, P^t_i, attri-good, default)

Connect tree and subtree with branch label a_i

}

return tree

8. EXPERIMENTAL RESULTS

Experiments were conducted on Core i3 Processor with 8GB of RAM running a Windows operating system and using open source python program. The initial values of Un-realization algorithm <D_s, Pu, Du, D_p> are <7,12,0,0>. Initially D_s contain 7 records, Pu contain 12 records, and Du and D_p are null. The result produced by the Un-realization algorithm is <0,12,7,10> whereas Modified Un-realization algorithm produces <0,12,7,6> with less D_p values. Further, the modified algorithm eliminates the duplicates in data.

Algorithm1 represents the Un-realization algorithm (UA) and Algorithm2 represents the Modified Un-realization algorithm (MUA). The size of the sample Dataset, power set and Un-realization Dataset are remains same for UA and MUA.

The perturbed Dataset size decreased for MUA, so that MUA outperforms better than UA. Two performance parameters are used like perturbation size and Modified Un-realization size. Perturbation size defines the size of the perturbed Dataset whereas Modified perturbation size determines the size of the perturbed Dataset after modification. Experiments are conducted on PIMA Benchmark Datasets T100, T200... T600... etc. These are the datasets which consists of 100, 200...600 records. Each dataset has been perturbed and their perturbed record size of Un-realization algorithm. Similarly, these datasets are perturbed using modified un-realization algorithm outperforms better than un-realization Algorithm.

The Modified Un-realization algorithm removes the duplication of data, so that the processing time, space and memory consumption are reduced as compared with the Un-realization algorithm. The performance of this algorithm was measured in terms of speed of the CPU, space complexity, and time complexity and perturbation rate. The CPU time spent on the Un-realization algorithm has been reduced 50 % as compared to the Modified Un-realization algorithm. The removal of duplicates in data leads to the faster reconstruction of the original Dataset, so that the CPU time has been reduced. The Modified Un-realization algorithm record size are less as compared with the UA.

The Modified CART algorithm uses Gini index where as ID3 uses entropy in constructing the decision tree. The main advantage of CART can handle both numeric, non-numeric and missing attributes where as ID3 will be confined to non-numeric attributes. The limitations of ID3 approach was over-fitting of data. CART uses cost-complexity pruning to avoid the over-fitting problem. Another advantage of CART is, it constructs binary trees in which each internal node contains exactly two nodes that help in better data splitter than ID3. Outliers will also be handled by CART. So, CART is proven to be more efficient than ID3 in many ways. The Gini index of the sample Dataset (D_s) of age and the Gini index of sample Dataset (D_p) and (D_u) are same. From this we can say that $\Delta Gini_{Age}(D_s) \approx \Delta Gini_{Age}(D_p) + \Delta Gini_{Age}(D_u)$.

Hence it is verified that the decision tree obtained from the un-realized samples by a Modified CART is proved to be same as the original decision tree constructed from the traditional CART algorithm.

Output Accuracy

According to the experimental results, MCART and traditional algorithms were compared in terms of execution time (Table 1 and Figure 3), decision tree (Figure 2) and analysis parameters (Table 2).

The output of this article is the decision tree generated from data sets. The decision trees have been constructed using CART and Modified CART within 2.23 ms where as traditional algorithm constructed with in 2.59 ms on an average. The decision trees obtained from both data sets, i.e., the original data set and the unrealized data set, were the same,

when the CART and modified CART were applied. Thus, our method has generated accurate outputs.

Storage complexity

The required storage is based on the data set compliment approach. The storage requirement will be also decreased from $2 * D_p + D_u$ to $2 * (D_p - 4) + D_u$.

Optimization of size of data set

In this article with the help of modified unrealized algorithm the size of the data set has been reduced from 10 values to 6 values. It helps in better computational times as well as reduces storage complexity.

Privacy risk

The risk of data loss has been reduced with the help of Modified unrealized algorithm and Modified CART algorithm. By the using data set compliment approach in unrealized data set reduces the privacy risk in most cases. In addition, with Modified CART, thus data is secured.

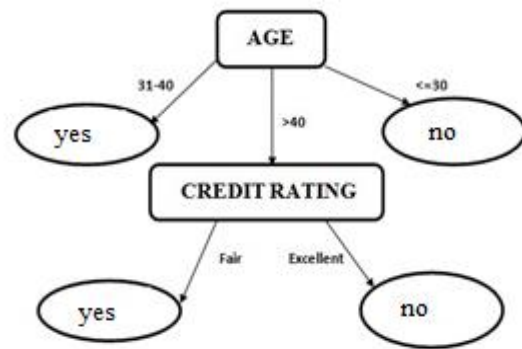


Figure 2. Decision tree obtained from MCART and traditional algorithm

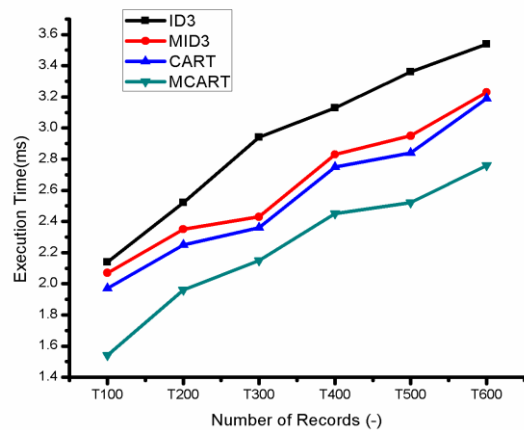


Figure 3. Line comparison between MCART and traditional algorithms

Table 1. Execution time comparison between MCART and traditional algorithms

Number of Records	ID3(ms)	MID3(ms)	CART(ms)	MCART(ms)
T100	2.14	2.07	1.97	1.54
T200	2.52	2.35	2.25	1.96
T300	2.94	2.43	2.36	2.15
T400	3.13	2.83	2.75	2.45
T500	3.36	2.95	2.84	2.52
T600	3.54	3.23	3.19	2.76

Table 2. Analysis parameter comparison between MCART and traditional algorithms

Analysis Parameter	ID3	MID3	CART	MCART
Time Complexity	$O(dm \log m)$	$O(dm^2 \log m)$	$O(dm \log m)$	$O(dm \log m)$
Output Accuracy on an average (ms)	2.93 ms	2.64 ms	2.56ms	2.23ms
Computational Complexity	High	High	Low	Low
Storage	High	High	High	Low
Dataset Complementation	Yes	Yes	Yes	Yes
Un-realized Algorithm	Yes	Yes	Yes	No
Modified Un-realized Algorithm	No	No	No	Yes
Privacy Risk	High	High	Low	Low

9. CONCLUSION

In this paper, a modified un-realization algorithm has been proposed to generate an optimal distorted dataset by removing duplicate elements. This reduces the computational complexity in data reconstruction. Further, a novel CART algorithm has also been proposed to achieve Privacy Preserving Decision Tree Learning. In this article the privacy is preserved by the combination of CART, Modified CART and Modified Un-realization Algorithms. The proposed method has given better results than the traditional Un-realization Algorithms and classification Algorithms by comparing the decision trees and their Gini values. In future, this article may be enhanced by applying advanced classification techniques like C4.5, CHAID, SVM etc.

REFERENCES

- [1] Agrawal, R., Srikant., R. (2000). Privacy preserving data mining. Proceedings of ACM SIGMOD Conference, 29(2): 439-450. <https://doi.org/10.1145/342009.335438>
- [2] Aggarwal, C.C., Yu, P.S. (2004). A condensation approach to privacy preserving data mining. Proceedings of International Conference on Extending Database Technology (EDBT), 2992: 183-199. https://doi.org/10.1007/978-3-540-24741-8_12
- [3] Aggarwal, C., Yu, P. (2008). Privacy preserving data mining models and algorithms. Springer Book. <https://doi.org/10.1007/978-0-387-70992-5>
- [4] Breiman, L. (1984). Classification and regression trees. The Wadsworth and Brooks-Cole Statistics Probability Series. Chapman & Hall.
- [5] Doud, J., Xu, S., Zhang, W. (2006). Privacy preserving decision tree mining based on random substitutions. Proceedings of International Conference Emerging trends in information and computer security (ETRICS '06), pp. 145-159. https://doi.org/10.1007/11766155_11
- [6] Agrawal, R., Srikant, R. (2000). Privacy-preserving data mining. Proceedings of the ACM SIGMOD International Conference on Management of Data, 29(2): 439-450. <https://doi.org/10.1145/342009.335438>
- [7] Sweeney, L. (2002). K-Anonymity-A model for protecting privacy. Uncertainty, Fuzziness and Knowledge-Based System, 10: 557-570. <https://doi.org/10.1142/S0218488502001648>
- [8] Lindell, Y., Pinkas, B. (2002). Privacy preserving data mining. Journal of Cryptology, 15(3): 177-206. <https://doi.org/10.1007/s00145-001-0019-2>
- [9] Ma, Q., Deng, P. (2008). Secure multi-party protocols for privacy preserving data mining. proc. Third Int'l Conf. Wireless Algorithm, Systems, and Applications (WASA '08), pp. 526-537. https://doi.org/10.1007/978-3-540-88582-5_49
- [10] Kisilevich, S., Elovici, Y., Shapira, B., Rokach, L. (2009). kACTUS 2: Privacy preserving in classification tasks using k-anonymity. Protecting Persons While Protecting the People, 63-81. https://doi.org/10.1007/978-3-642-10233-2_7
- [11] Fong, P.K. (2008). Privacy preservation for training data sets in database: Application to decision tree learning. Master's thesis, Dept. Of Computer Science, University of Victoria. <https://doi.org/10.1109/TKDE.2010.226>
- [12] Fong, P.K., Weber-Jahnke, J.H. (2012). Senior member. privacy preserving decision tree learning using unrealized data sets. Proceeding of the IEEE Transactions on Knowledge and Data Engineering, 24(2): 353-364. <https://doi.org/10.1109/TKDE.2010.226>
- [13] Teki, S.M., Varma, M.K., Yadav, A.K. (2019). Brain tumour segmentation using U-net based adversarial networks. Traitement du Signal, 36(4): 353-359. <https://doi.org/10.18280/ts.360408>
- [14] Gopalan, N.P., Satyanarayana Murthy, T., Venkateswarlu, Y. (2018). Hiding critical transactions using un-realization approach. International Journal of Pure and Applied Mathematics, 118(7): 629-633. <https://acadpubl.eu/jsi/2018-118-7-9/articles/7/83.pdf>
- [15] Gopalan, N.P., Satyanarayana Murthy, T., Venkateswarlu, Y. (2018). An efficient method for hiding association rules with additional parameter metrics. International Journal of Pure and Applied Mathematics, 118(7): 285-290.
- [16] Satyanarayana Murthy, T., Gopalan, N.P., Athira, T.R. (2019). Hiding critical transactions using modified un-realization approach. International Journal of Information and Computer Security, Inderscience Publishers.
- [17] Satyanarayana Murthy, T., Gopalan, N.P., Yakobu, D. (2019). An efficient un-realization algorithm for privacy preserving decision tree learning using McDiarmid's bound. International Journal of Innovative Technology and Exploring Engineering (IJITEE), 8(4S2): 499-502.
- [18] Satyanarayana Murthy, T., Gopalan, N.P. (2018). A novel algorithm for association rule hiding. International Journal of Information Engineering and Electronic Business (IJIEEB), 10(3): 45-50. <https://doi.org/10.5815/ijieeb.2018.03.06>
- [19] Satyanarayana Murthy, T., Gopalan, N.P., Gunturu, S. (2018). A novel optimization based algorithm to hide sensitive item-sets through sanitization approach. International Journal of Modern Education and Computer Science (IJMECS), 10(10): 48-55. <https://doi.org/10.5815/ijmecs.2018.10.06>
- [20] Satyanarayana Murthy, T., Gopalan, N.P., Alla, D.S.K.

- (2018). The power of anonymization and sensitive knowledge hiding using sanitization approach. *International Journal of Modern Education and Computer Science (IJMECS)*, 10(9): 26-32. <https://doi.org/10.5815/ijmecs.2018.09.04>
- [21] Satyanarayana Murthy, T., Gopalan, N.P., Preethi, G. (2018). An efficient way of anonymization without subjecting to attacks using secure matrix method. 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), pp. 1462-1465. <https://doi.org/10.1109/ICCONS.2018.8663048>
- [22] Satyanarayana Murthy, T., Gopalan, N.P., Pudi, P. (2018). An efficient meta-heuristic chemical reaction optimization based algorithm for association rule hiding using an advanced perturbation approach. 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), pp. 1466-1471. <https://doi.org/10.1109/ICCONS.2018.8662983>
- [23] Gopalan, N.P., Satyanarayana Murthy, T. (2017). Association rule hiding using chemical reaction optimization. *Soft Computing for Problem Solving*, 249-255.
- [24] Verykios, V.S., Elmagarmid, A.K., Bertino, E., Saygin, Y., Dasseni, E. (2004). Association rule hiding. *IEEE Transactions on Knowledge and Data Engineering*, 434-447. <https://doi.org/10.1109/TKDE.2004.1269668>