PRIVACY PRESERVING FOR EXPERTISE DATA USING K-ANONYMITY TECHNIQUE TO ADVICE THE FARMERS

1T. S. N. MURTHY, 2VANDANA BABU. T, 3D.NAGABHUSHANAM, 4P.SESHU BABU

1,2,3,4Vignan University, Vadlamudi
Email:murthyteki@gmail.com,vandana9925@gmail.com,bhushanamdot@gmail.com,seshu.babu08@gmail.com

Abstract - Privacy preserving for an expertized dataset using security techniques to advice the farmers. Earlier protects the privacy of the data by perturbing the data through a random process. The second approach uses cryptographic techniques to perform secure multi-party computation. In this paper, we propose privacy preservation for a data classification using K-anonymity algorithm. This paper includes a protection model named k-anonymity and a set of accompanying policies for deployment. K-anonymity Protection means the information for each person contained in the release cannot be distinguished from at least k-1 individuals whose information also appears in the release. Finally, we show that our method contributes new and efficient ways to anonymize data and preserve patterns during anonymization .This intelligent system using decision tree algorithm. The decision trees generated by ID3 can be used for classification.ID3 builds decision trees from a set of training data in the same way as ID3, using the concept of information entropy. Using this ID3 Algorithm we developed a new ‘Privacy Preserving for expertise data’. This system is mainly aimed for identifying the diseases and disease management in Lemon fruits and Lemon plants to advise the farmers through online in the villages to obtain standardized yields and provide protection. The present advisory system is also used privacy preserving technique.

Keywords - Expert Systems, K-anonymity, Machine Learning, ID3 Algorithm, Lemon Crop, JSP & MYSQL

I. INTRODUCTION TO EXPERT SYSTEMS

An Expert System is a computer program conceived to simulate some forms of human reasoning (by the intermediary of an inference engine) and capable to manage an important quantity of expert knowledge. A system that uses human knowledge captured in a computer to solve problems that ordinarily require human expert. A computer program designed to model the problem solving ability of a human expert. The computer program that uses knowledge and inference procedures to solve problems that was difficult enough to acquire significant human expertise for their solutions. An expert system is a computer application that solves complicated problems that would otherwise require extensive human expert. To do so it simulates the human reasoning process by applying specific knowledge and interface. This report explained on the expert system for decision making of giving the best solution to solve Medical Diagnosis problems. Lemon is one of the most commonly used fruits in India. Its scientific name is citrus. It is the leading fruit crop in India and it is considered to be the king of fruits. Lemon occupies around 56% of the total crop production in India Even though lemon is cultivated in all over India but Karnataka, Tamil Nadu, Andhra Pradesh and united states are the premium producers of lemon in India. There are about popular varieties in lemons. Lemons contain Naringenin is found to have a bio-active effect on human health as antioxidant, free radical scavenger, anti-inflammatory, and immune system modulator. They also contain a small level of vitamin A, and other flavonoid anti-oxidants such as a, and b-carotenes, beta-cryptoxanthin, zea-xanthin and lutein.

These compounds are known to have antioxidant properties. Vitamin A also required for maintaining healthy mucus membranes and skin and is also essential for vision. Consumption of natural fruits rich in flavonoids helps the body to protect from lung and oral cavity cancers . ID3 is an algorithm used to generate a decision tree [1] developed by Ross Quinlan. ID3 builds decision trees from a set of training data in the same way as ID3, using the concept of information entropy. The information gain is based on the decrease in entropy after a dataset is split on an attribute. First the entropy of the total dataset is calculated. The dataset is then split on the different attributes. The entropy for each branch is calculated. Then it is added proportionally, to get total entropy for the split. And next we can use privacy preserving technique for K-anonymity.

II. PROPOSED SYSTEM

ID3 is an algorithm used to generate a decision tree developed by Ross Quinlan. The decision trees generated by ID3 can be used for classification algorithms. Applying privacy using K-anonymity is one of the most accepted models for privacy in real-life applications, and provides the theoretical basis for privacy related legislation. This is for several important reasons:
1) The k-anonymity [7], [8] model defines the privacy of the output of a process and not of the process itself. This is in sharp contrast to the vast majority of privacy models that were suggested earlier, and it is in this sense of privacy that clients are usually interested.
2) It is a simple, intuitive, and well understood model. Thus, it appeals to the non-expert who is the end client of the model.
3) Although the process of computing a k-anonymous table may be quite hard, it is easy to validate that an outcome is indeed k-anonymous. Hence, non-expert data owners are easily assured that they are using the model properly.
4) The assumptions regarding separation of quasi-identifiers, mode of attack, and variability of private data have so far withstood the test of real-life scenarios.

We begin our discussion by defining a private database and then defining a model of that database.

Definition 1: (A Private Database) A private database \( T \) is a collection of tuples from a domain \( D = A \times B = A_1 \times \ldots \times A_2 \times B_1 \times \ldots \times B_2 \). \( A_1, \ldots, A_k \) are public attributes and \( B_1, \ldots, B_k \) are private attributes. We denote \( A = A_1 \times \ldots \times A_k \) the public subdomain of \( D \). For every tuple \( x \in D \), the projection of \( x \) into \( A \), denoted \( xA \), is the tuple in \( A \) that has the same assignment to each public attribute as \( x \). The projection of a table \( T \) into \( A \) is denoted \( TA = \{xA : x \in T \} \).

Definition 2: (A Model) A model \( M \) is a function from a domain \( D \) to an arbitrary output domain \( O \). Every model induces an equivalence relation on \( D \), i.e., \( \forall x, y \in D, x = y \Leftrightarrow M(x) = M(y) \). The model partitions \( D \) into respective equivalence classes such that \( [x] = \{ y \in D : x = y \} \). In the Lemon decision tree example, the decision tree is a function that assigns bins to tuples in \( T \).

Definition 3: (A Release) Given a database \( T \) and a model \( M \), a release \( MT \) is the pair \( (M, pT) \), where \( pT \) (for population) is a function that assigns to each equivalence class induced by \( M \) the number of tuples from \( T \) that belong to it, i.e., \( pT([x]) = |T^x| \).

Definition 4: (A Public Identifiable Database) A public identical database \( TID = \{(idx, xA) : x \in T \} \) is a projection of a private database \( T \) into the public subdomain \( A \), such that every tuple of \( TA \) is associated with the identity of the individual to whom the original tuple in \( T \) pertained.

Definition 5: (A Span) Given a model \( M \), the span of a tuple \( a \in A \) is the set of equivalence classes induced by \( M \), which contain tuples \( x \in D \), whose projection into \( A \) is \( a \). Formally, \( S(a) = \{[x] : x \in D \land xA = a \} \).

When \( M \) is evident from the context, we will use the notation \( S(a) \).

Definition 6: (Linking attack using a model) A linking attack on the privacy of tuples in a table \( T \) from domain \( A \times B \), using a release \( MT \), is carried out by
1. Taking a public identifiable database \( TID \) which contains the identities of individuals and their public attributes \( A \).
2. Computing the span for each tuple in \( TID \).
3. Grouping together all the tuples in \( TID \) that have the same span. This results in sets of tuples, where each set is associated with one span.
4. Listing the possible private attribute value combinations for each span, according to the release \( MT \).

The tuples that are associated with a span in the third step are now linked to the private attribute value combinations possible for this span according to the fourth step.

Definition 7: (k-anonymous release) A release \( MT \) is k-anonymous with respect to a table \( T \) if a linking attack on the tuples in \( T \) using the release \( MT \) will not succeed in linking private data to fewer than \( k \) individuals.

Proof Assume an attacker associated an individual’s tuple \((idx, xA) \in TID \) with its span \( S(xA) \). We will show that if one of the conditions holds, the attacker cannot compromise the k-anonymity of \( x \). Since this holds for all tuples in \( T \), the release is proven to be k-anonymous.

1. \( S(xA) = \{[x]\} \). Since the equivalence class \([x]\) is the only one in \( S(xA) \), then according to Claim 2, tuples whose span is \( S(xA) \) belong to \([x]\) regardless of their private attribute values. Therefore, no private attribute value can be associated with the span, and the attacker gains no private knowledge from the model in this case. In other words, even if the attacker manages to identify a group of less than \( k \) individuals and associate them with \( S(xA) \), no private information will be exposed through this association.

2. \( |S(xA)| \geq k \). In this case, the model and the equivalence class populations might reveal to the attacker as much as the exact values of private attributes for tuples in \( T \) that belong to equivalence classes in \( S(xA) \). However, since \( |S(xA)| \geq k \), the...
number of individuals (tuples) that can be associated with the span is k or greater.

III. K-ANONYMIZED LEMON DATA

<table>
<thead>
<tr>
<th>part</th>
<th>climate</th>
<th>drainage</th>
<th>diesuddenly</th>
<th>decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>leaf</td>
<td>hot</td>
<td>good</td>
<td>no</td>
<td>notaffected</td>
</tr>
<tr>
<td>leaf</td>
<td>hot</td>
<td>good</td>
<td>yes</td>
<td>notaffected</td>
</tr>
<tr>
<td>seed</td>
<td>*</td>
<td>*</td>
<td>no</td>
<td>notaffected</td>
</tr>
<tr>
<td>stem</td>
<td>*</td>
<td>*</td>
<td>yes</td>
<td>notaffected</td>
</tr>
<tr>
<td>stem</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>notaffected</td>
</tr>
<tr>
<td>leaf</td>
<td>warm</td>
<td>*</td>
<td>no</td>
<td>notaffected</td>
</tr>
<tr>
<td>leaf</td>
<td>moist</td>
<td>*</td>
<td>*</td>
<td>notaffected</td>
</tr>
<tr>
<td>leaf</td>
<td>warm</td>
<td>*</td>
<td>no</td>
<td>notaffected</td>
</tr>
<tr>
<td>leaf</td>
<td>moist</td>
<td>*</td>
<td>yes</td>
<td>notaffected</td>
</tr>
<tr>
<td>seed</td>
<td>moist</td>
<td>*</td>
<td>yes</td>
<td>notaffected</td>
</tr>
<tr>
<td>seed</td>
<td>moist</td>
<td>*</td>
<td>yes</td>
<td>notaffected</td>
</tr>
<tr>
<td>seed</td>
<td>hot</td>
<td>*</td>
<td>yes</td>
<td>affected</td>
</tr>
<tr>
<td>seed</td>
<td>hot</td>
<td>*</td>
<td>yes</td>
<td>affected</td>
</tr>
</tbody>
</table>

Fig. 1. Magnetization as a function of applied field.

IV. ID3 ALGORITHM

Use ID3 algorithm for classification. In pseudo code the algorithm is:
1. Check for base cases
2. For each attribute a
3. Let a_best be the attribute with the highest information gain
4. Create a decision node that splits on a_best
5. Recurs on the subsists obtained by splitting on a_best, and add those nodes as children of node
   a) Avoiding over fitting the data (Determining how deeply to grow a decision tree)
   b) Reduced error pruning.
   c) Rule post-pruning.
   d) Handling continuous attributes.
   e) Choose one by one attribute selection measure.
   f) The training data with missing attribute values in a given table.
   g) Handling different costs of attributes.
   h) Improve computational efficiency.

To describe the operation of ID3, we use a Data for fruit classification example. The Learning Data for lemon classification example:

i). Calculate the Entropy

\[
\text{Entropy}(S) = -P(\text{Positive}) \log_2 P(\text{Positive}) - P(\text{Negative}) \log_2 P(\text{Negative})
\]

P(positive): positive examples in lemon table
P(negative): negative examples in lemon table
if S is (0.5+, 0.5-) then Entropy(S) is 1,
if S is (0.67+, 0.33-) then Entropy(S) is 0.92,
if P is (1+, 0-) then Entropy(S) is 0.

ii). The beginning state of the training data in table (with the decision classification) is that (5/14) are notedeffect, (9/14) are affected. Thus the entropy of the starting set is

\[
\text{Info}(\text{decision}) = -9/14 \log_2(9/14) - 5/14 \log_2(5/14) = 0.940
\]

iii). Choosing the part as the splitting attribute, there are given below

\[
\text{Info(\text{part})} = -5/14 \log_2(5/14) - 3/14 \log_2(3/14) \approx 0.694
\]

iv). The ID3 algorithm must determine what the gain in information is by using this split, to do this, we calculate the weighted sum of these last two entropies to get gain value is given

\[
\text{Gain(\text{part})} = \text{Info(\text{decision})} - \text{Info(\text{part})} = 0.246
\]

v). The gain in entropy by using the split attribute is thus

\[
\text{Gain(\text{part})} = 0.940 - 0.694 = 0.246
\]

vi). Looking at the diesuddenly attribute, there have been two tuples that are

\[
\text{Info(\text{diesuddenly})} = \frac{6}{14} \log_2(6/14) = 0.892
\]

vii). The gain in diesuddenly attribute is given below

\[
\text{Gain(\text{diesuddenly})} = \frac{6}{14} \log_2(6/14) - \frac{8}{14} \log_2(8/14) = 0.048
\]

viii). The climate as the gain value is given below

\[
\text{Gain(\text{climate})} = \frac{1}{4} \log_2(1/4) = 0.000
\]

ix). Calculate the GAIN RATIO in ID3 Algorithm.

\[
\text{Gain Ratio} = \frac{\text{Gain}}{\text{Info(\text{attribute})}}
\]

Algorithm for k-Anonymous Decision Tree uses some formulas of ID3

1. procedure Make Tree (T, A, k)
   2. r ← root node.
   3. c and List ← \{(a, r):a ∈ A\}
   4. while c and List contains candidates with positive gain do
   5. best C and ← candidate from c and List with highest gain.
   6. if best C and maintains k-anonymity then
   7. Apply the split and generate new nodes N.
   8. Remove candidates with the split node from c and List.
   9. c and List ← c and List U \{(a, n):a ∈ A, n ∈ N\}. 
10. else 
11. remove best C and from c and List.
12. end if
13. end while
14. return generated tree.
15. end procedure

The decision tree is a function that maps points in the original domain to the leaves of the tree, and inside the leaves, to bins, according to the class value. Hence those bins constitute partitions of the domain – each bin forms an equivalence class and contains all the tuples that are routed to it.

<table>
<thead>
<tr>
<th>part</th>
<th>climate</th>
<th>drainage</th>
<th>decison</th>
</tr>
</thead>
<tbody>
<tr>
<td>leaf</td>
<td>hot</td>
<td>good</td>
<td>not effected</td>
</tr>
<tr>
<td>leaf</td>
<td>hot</td>
<td>good</td>
<td>not effected</td>
</tr>
<tr>
<td>seed</td>
<td>warm</td>
<td>good</td>
<td>not effected</td>
</tr>
<tr>
<td>stem</td>
<td>moist</td>
<td>good</td>
<td>not effected</td>
</tr>
<tr>
<td>stem</td>
<td>warm</td>
<td>poor</td>
<td>not effected</td>
</tr>
<tr>
<td>stem</td>
<td>warm</td>
<td>poor</td>
<td>not effected</td>
</tr>
<tr>
<td>stem</td>
<td>warm</td>
<td>poor</td>
<td>not effected</td>
</tr>
<tr>
<td>seed</td>
<td>moist</td>
<td>good</td>
<td>not affected</td>
</tr>
<tr>
<td>seed</td>
<td>hot</td>
<td>poor</td>
<td>not affected</td>
</tr>
<tr>
<td>seed</td>
<td>hot</td>
<td>poor</td>
<td>not affected</td>
</tr>
<tr>
<td>stem</td>
<td>moist</td>
<td>good</td>
<td>not affected</td>
</tr>
<tr>
<td>stem</td>
<td>moist</td>
<td>good</td>
<td>not affected</td>
</tr>
</tbody>
</table>

V. RESULTS

The decision tree is a function that maps points in the original domain to the leaves of the tree, and inside the leaves, to bins, according to the class value. Hence those bins constitute partitions of the domain – each bin forms an equivalence class and contains all the tuples that are routed to it.

CONCLUSION

In the present developed privacy preserving for expertisedata providing privacy using K-Anonymity Algorithm .This system is for finding the better solutions from to the symptoms submitted to the system by the user. The process of anonymization is oblivious to any future analysis that would be carried out on the data. Therefore, during anonymization, attributes critical for the analysis may be suppressed whereas those that are not suppressed may turn out to be irrelevant. When there are many public attributes the problem is even more difficult, due to the curse of dimensionality. In that case, since the data points are distributed sparsely, the process of k-anonymization reduces the effectiveness of data mining algorithms on the anonymized data and renders privacy preservation impractical. Its main emphasis is to have a well-designed interface for giving cultivation related advices and suggestions in the area of Horticulture (Lemon) field by providing facilities like dynamic interaction between expert sys and the user without the need of expert (Farmer) at any times. By through interaction with the users and benefits the functionality of the system can be extended further many more areas in and around many areas.

REFERENCES