Learning Decision Trees from Anonymized Data

Jianhua Shao and Jasmin Beckford
School of Computer Science and Informatics
Cardiff University, Cardiff, UK
{shaoj, beckfordj@cardiff.ac.uk}@cardiff.ac.uk

***Abstract*—There is much interest in developing solutions for protecting data privacy in recent years, and many privacy models and data sanitization methods have been proposed. However, relatively little has been done to understand how existing data analysis techniques may be adapted to work with sanitized data. In this paper we report a study on learning decision trees from anonymized data. We sanitize data using the Mondrian algorithm to satisfy k-anonymity and adapt the ID3 algorithm to learn decision trees from sanitized data. Our preliminary experiments show that accurate decision trees can be learnt from anonymized data, and degradation of classification accuracy is no more than 2% with typical settings.**

I. Introduction

With widespread and increased deployment of data capturing applications and services, more and more data is being generated, collected and analyzed. While analyzing such data is beneficial and important to businesses and organizations, the data may contain sensitive information about individuals and their privacy must be respected. Unfortunately, simply removing individual identification information such as social security number from a dataset is not sufficient to protect privacy. To address this, there is much interest in developing solutions for protecting data privacy in recent years and many privacy models and data sanitization methods have been proposed [4].

A privacy model specifies certain properties that a dataset must satisfy, in order to protect the data against a particular type of privacy risk. For example, k-anonymity is a well-known privacy model which requires each individual contained in a dataset to be not distinguishable from at least $k-1$ other individuals based on a set of so-called quasi-identifiers (QIDs) such as gender, postcode and age [13]. When a dataset to be released does not satisfy a particular privacy model, data sanitization is carried out to perturb or modify the data as necessary. For example, the age of a person may be generalized into a range, so that k individuals can have the same age range in order to satisfy k-anonymity.

While much study has been carried in the last decade or so to derive data sanitization methods that can anonymize data to achieve different levels of privacy protection, including various heuristic data utility optimization measures, surprisingly little has been done to consider how some common data analysis tasks may be conducted using anonymized data. As a step in this direction, we consider in this paper how a decision tree may be learnt from sanitized data and assess the effect of data sanitization on classification accuracy.

How data analysis tasks may be performed on sanitized data will depend heavily on how data is sanitized to achieve a level of privacy protection. The goal of data sanitization is to ensure that while privacy requirements are met, the utility of sanitized data is optimized. That is, the sanitized data should be as useful as possible in subsequent data analysis studies. To achieve this, different heuristics have been proposed. Some heuristics use generic measures to guide data sanitization, for example, to ensure that modification to the data is minimum [8]. By keeping sanitized data as close to original data as possible, it is hoped that the sanitized data will be as useful as the original data in analytic studies. Other methods take the characteristics of intended data analysis task into account when sanitizing the data, for example, to produce sanitized data specifically optimized for learning a decision tree [3].

Given that the purpose of anonymizing a set of data is to enable its publication to support analytic studies, the approach that optimizes data sanitizaton by taking data analysis requirements into account is appealing. However, this is not realistic in practice. Consider, for example, a medical researcher who is to analyze data collected from several hospitals. Due to privacy concerns, it is unlikely that hospitals will be willing to share their original data, but the researcher may be granted access to anonymized data. Clearly, it would not be reasonable to assume that hospitals will know exactly what the researcher's analytic requirements may be, and they may not be willing to sanitize data per analytic study either.

Thus, instead of attempting to sanitize data with some intended analysis in mind, it is desirable that data analysis techniques can be adjusted to work with generically sanitized data. In this paper we propose to modify a decision tree learning method to work with generically sanitized data. Our approach does not require data sanitization methods to know decision tree learning requirements [3], nor do we require the anonymization process to release extra statistical information to aid the derivation of a decision tree [6]. As a proof of concept, we experimented our idea on the simple ID3 decision tree learning algorithm, and we tested our method on the *Adult* dataset [10], a de facto benchmarking dataset for testing privacy solutions, in order to determine the extent to which privacy preservation of data truly affects its usefulness. Our results show that accurate decision trees can be learnt from anonymized data and degradation of classification accuracy is small with typical settings when compared to the trees derived from the original data.

The rest of the paper is organized as follows. In Section 2, we give the necessary definitions that we use in the paper and the background material necessary to understand our approach. In Section 3, we introduce our modified ID3. Section 4 briefly discusses experimental results and finally in Section 5, we conclude the paper.

II. Related Work

Most work on assessing and determining the usefulness of sanitized data is based on some generic, task-independent measures. For example, earlier measures rely purely on some data characteristics such as the size of an equivalence group in k-anonymization [1], [8], [15], [7], as an indication of how likely the data will be useful in subsequent data analysis tasks. These measures in fact are measures of how much information is lost during data sanitization, rather than how useful the sanitized data will be in practice. Workload based measures have also been proposed [14], [5], trying to link sanitized data to the likely tasks to be performed on them, thereby giving more plausible measures. In so doing, these measures resort to some generic operations that a data analysis task is likely to perform, for example, counting the number of instances in a group. Clearly, these measures are a guess of how sanitization may affect a data analysis task, but not an accurate reflection of how useful such data will be in real applications. We do not propose yet another generic measure of how likely data will be useful, but study how data analysis techniques may be modified to work with sanitized data, thereby assessing the effect of data sanitization on data analysis.

There are also works that attempt to optimize a sanitization process by taking intended data analysis tasks into account. For example, Inan et al [6] proposed a method where in addition to publishing the sanitized data, some associated statistics such as the mean of an equivalence group is also released. Such statistics can then help derive more accurate classifiers from sanitized data. Fung et al [3] proposed a method which sanitize the data in a top-down specialization fashion, where each step is guided by class entropy measure. As such, the data is sanitized with a specific type of classifier learning method in mind. In contrast, our approach does not require the data publisher to release any extra information apart from the sanitized data itself, and we do not require the data to be sanitized specifically for a particular task. Instead, we adapt an existing decision tree learning method to work with generically sanitized data.

III. Preliminaries

A. k-Anonymity and Set-Based Generalization

Without loss of generality, we assume that data is contained within a single table $T(A_1, A_2, \ldots, A_m, C)$, where each $A_j, 1 \leq j \leq m$, is a numerical or categorical attribute and C is a categorical class attribute. We may drop C from T when it is not needed in the discussion. We also assume that the first q attributes in T are quasi-identifiers (QIDs). That is, they are publicly available (e.g. from a voters list) and may be used by an adversary to identify an individual contained in T. Other attributes are deemed as sensitive attributes.

Definition 1 (k-Anonymity): Let $T(A_1, A_2, \ldots, A_m)$ be a table and $v = \langle v_1, v_2, \ldots, v_m \rangle$ be any tuple in T. T satisfies *k-anonymity* if

$$|\{t \mid t \in T, t.A_1 = v_1, t.A_2 = v_2, \ldots, t.A_q = v_q\}| \geq k$$

That is, for any tuple in T, there are at least $k-1$ other tuples that have the same QID values. We call each such group an *equivalence group*.

When k-anonymity is not satisfied, data needs to be modified prior to its publication to ensure that equivalence groups form across the entire dataset. Different methods have been proposed to do so and one approach is set-based generalization [9].

Definition 2 (Set-based Generalization): Let $T(A_1, A_2, \ldots, A_m)$ be a table, $A_1, A_2, \ldots, A_q$ be QIDs, and $G = \{t_1, t_2, \ldots, t_k\}$ be a set of tuples in T. A *set-based generalization* of G is a replacement of each value $a_{ij}, 1 \leq i \leq k, 1 \leq j \leq q$, in G by $\tilde{a}_{ij} = \bigcup_{a_{ij} \in A_j} a_{ij}$, the union of all values in A_j occurred in G. We denote a set-based generalization of G by $\tilde{G}$ and call each $\tilde{a}_{ij}$ a *generalized value*.

To explain the concepts of k-anonymity and set-based generalization, consider this example. Suppose that we have Table I that is required to be 3-anonymized, where *Age*, *Gender* and *Postcode* are assumed to be QIDs and *Disease* is the only sensitive attribute.

TABLE I
Original table

Age	Gender	Postcode	Disease
20	M	1032	HIV
23	M	2113	Flu
27	F	5632	HIV
25	F	1023	Obesity
27	F	1132	Cancer
29	M	3232	Heart Attack

As the table currently does not satisfy 3-anonymity, we apply set-based generalization to the table to generate the 3-anonymized table in Table II, where a generalized value is represented by listing its values in bracket and we interpret it as representing any of its members, e.g. (*M, F*) may represent *M* or *F*.

TABLE II
A 3-anonymised table

Age	Gender	Postcode	Disease
(20, 23, 27)	(M, F)	(1032, 2113, 5632)	HIV
(20, 23, 27)	(M, F)	(1032, 2113, 5632)	Flu
(20, 23, 27)	(M, F)	(1032, 2113, 5632)	HIV
(25, 27, 29)	(M, F)	(1023, 1132, 3232)	Obesity
(25, 27, 29)	(M, F)	(1023, 1132, 3232)	Cancer
(25, 27, 29)	(M, F)	(1023, 1132, 3232)	Heart Attack

A key issue in set-generalizing a table T is to find an optimal partition of T into groups of k tuples, and then apply the generalization given in Definition 2 to each group. Many different methods have been proposed. In this paper we follow the Mondrian algorithm [8], though our approach is applicable to other sanitization methods too.

B. The Mondrian Algorithm

Mondrian was originally proposed by LeFevre et al to anonymize a set of data to satisfy k-anonymity using a multidimensional global recoding [8]. It works by recursively choosing an attribute and partitioning the dataset at the median of the chosen attribute until k-anonymity is violated, that is, when the partition will result in a group that has less than k tuples. The Mondrian method is given in Algorithm 1.

Algorithm 1 *Mondrian*

```
input: a dataset $D$
output: a partition of $D$
1. if $D$ cannot be split then
2.     return $D$
3. else
4.     $dim \leftarrow ChooseAttribute(D)$
5.     $fs \leftarrow CalculateFrequency(D, dim)$
6.     $SplitVal \leftarrow FindMedian(fs)$
7.     $D_L \leftarrow \{t | t.dim \leq splitVal\}$
8.     $D_R \leftarrow \{t | t.dim > splitVal\}$
9. return $Mondrian(D_L) \cup Mondrian(D_R)$
```

The algorithm takes a dataset D as input. If a partition can be made, i.e. the dataset is large enough to be cut into two subsets each having at least k tuples (step 1), then a best cut along one of the attributes will be sought in the following steps. First, an attribute with the widest normalized range of values will be selected (step 4). Then the frequency of each distinct value in the selected attribute is calculated, and these frequencies are placed in an ordered set (step 5). Note that as we use the median value to split D, the ordering of the values in the frequency set is significant. For numerical attributes, the values are placed in ascending order. For categorical attributes, any ordering can be used but the same ordering must be consistently used for the attribute throughout the partitioning process. For simplicity we use alphabetical ordering for all categorical attributes. For example, the frequency set for the *Race* attribute in the *Adult* dataset [10] is as follows:

Value	frequency
Amer-Indian-Eskimo	286
Asian-Pac-Islander	895
Black	2817
Other	231
White	25,933

This frequency set is then used to determine a point at which we will partition D (step 6). Two modes of partitioning are possible: strict partitioning and relaxed partitioning. Strict partitioning requires that when D is split, its two resultant subsets must not contain any overlapping values within the attribute concerned. This means that the set of values containing the median must be taken in its entirety and placed in one partition only. Relaxed partitioning, on the other hand, allows the two partitions to contain the same values. This would mean that in our example the value *White* could be split into 10,852 instances in one partition and the other 15,081 instances in another. Due to our set-based generalization, strict partitioning will be used.

The final step is simply to split D and recursively run the algorithm using the two partitions as inputs. The algorithm stops when there are no further cuts can be made to any of the partitions. After the algorithm returns a set of partitions, we apply set-based generalization to each partition, i.e. the QID values for the instances contained with each partition are unioned and each individual value is replaced by the generalized value.

C. The ID3 Algorithm

The ID3 algorithm is one of simplest decision tree learning algorithms. As the purpose of our study is to assess the impact of data sanitization on classification accuracy, rather than attempting to develop a new decision tree learning algorithm that can produce as accurate classification as possible, a simple method such as ID3 suffices. The method is shown in Algorithm 2, which is based on a version given in [11].

Algorithm 2 *ID3*

```
input: a dataset $D$ with attributes $A = \{A_1, A_2, \ldots, A_m\}$ and
       class values $C = \{c_1, c_2, \ldots, c_s\}$
output: a decision tree
1.  $dim \leftarrow ChooseAttribute(D)$
2.  $\{D_1, D_2, \ldots, D_h\} \leftarrow partition(D, dim)$
3.  make each $D_j$ a child node of $dim$
4.  for each $D_j$ do
5.      if all class values in $D_j$ are $c_s$ then
6.          label $D_j$ with $c_s$
7.      else if $|A| = 1$ then
8.          label $D_j$ with the majority of $C$
9.      else
10.         ID3($A - \{A_j\}, D_j$)
```

The algorithm works as follows. Each time one attribute is chosen to be the root of the tree, and the choice is based on entropy heuristic: the attribute that has the most gain in entropy will be chosen as the root (step 1). The dataset D is then split into subsets based on the distinct values of the root attribute (step 2), and they are made children of the root (step 3). Each subset is then considered in turn: if all the tuples in it belong to the same class, then we label it with the class value; if dim is the last attribute, then we label the subset with the majority class value in the group. Otherwise, the ID3 algorithm is recursively called with the subset and reduced attribute list as input.

IV. Privacy-Enabled Decision Tree Learning

In this section we introduce our modified ID3 algorithm to work with data sanitized by the Mondrian algorithm with set-based generalization. As some of the original values have been replaced by generalized values by Mondrian in order to satisfy k-anonymity, we need to consider how such generalized values are dealt with in two key steps of the ID3 algorithm: the calculation of entropy when selecting the root of the tree and the partition of D into subsets. That is, we need to consider how steps 1 and 2 of the ID3 algorithm given in Algorithm 2 may be modified.

A. Calculating Entropy

In our sanitized data, a generalized value can represent a number of possible values. Therefore, it is necessary to consider how this can be taken into account and result in an accurate interpretation of the entropy for a given attribute. There are three possible ways of doing so, and we will explain them using Table III as an example.

TABLE III
Example for Entropy Calculation

Sex	Salary
(M, F)	≤ 50
F	≤ 50
F	> 50
F	≤ 50
M	> 50
(F, M)	≤ 50
F	≤ 50
(M, F)	> 50

One possible solution is to consider a generalized value such as (M, F) as representing a value in its own right. That is, in a tuple containing this value, we interpret it as an individual who is either male or female, but we are not certain if it is male or female. Each time it appears in the dataset, instead of attempting to determine whether a particular instance actually possesses the *M* or *F* value, it is simply left as (M, F). This way, the entropy value of the *Sex* attribute can be calculated as normal, and we simply have three possible values rather than two for the *Sex* attribute.

Using this method to compute the entropy for the *Sex* attribute given in Table III, we obtain the following:

$$\begin{aligned}
E(Sex_F) &= -\frac{1}{4}\log_2\frac{1}{4} - \frac{3}{4}\log_2\frac{3}{4} = 0.811 \\
E(Sex_M) &= -\frac{1}{1}\log_2\frac{1}{1} - \frac{0}{1}\log_2\frac{0}{1} = 0 \\
E(Sex_{(M,F)}) &= -\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} = 0.918
\end{aligned}$$

Note that with this method, a new value label, (M, F), would be introduced into the decision tree, as the *Sex* attribute would have a three-way split. This would pose no problem if unseen cases to be classified are also generalized in the same way, but the new label must be carefully interpreted when using the tree to classify cases that are not generalized.

Another method for calculating an attribute's entropy is to convert generalized values into original values a priori. For example, (M, F) could be mapped randomly to either *M* or *F*. Once the generalized dataset has been converted this way, entropy can then be calculated as normal. While this allows entropy calculation to be carried out normally and does not suffer the classification problem that the first method has, it also has a problem. When *M* and *F* are generalized into (M, F), the distribution of these values within the equivalence group is lost. Mapping a generalized value back to one of the original values can essentially be seen as a process of reconstructing the original data from its generalized version, and without some relevant statistics available [6], this approach can attract significant error in the calculation of entropy.

In this paper we propose to treat each generalized value as equally likely to represent any of its members, and factor this likelihood into entropy calculation. We consider a generalized value such as (M, F) in Table III to indicate that there is an equal probability of 0.5 for it to be *M* or *F*. In general, if a generalized value contains r values, then we consider that there is a probability of $\frac{1}{r}$ to be one of the original values contained in it. When totalling the number of each value in the dataset, we have 5.5 for *F* and 2.5 for *M*, and the original size of the dataset is preserved.

Using this method to compute the entropy for the *Sex* attribute given in Table III, we obtain the following:

$$\begin{aligned}
E(Sex_F) &= -\frac{1.5}{5.5}\log_2\frac{1.5}{5.5} - \frac{4}{5.5}\log_2\frac{4}{5.5} = 0.845 \\
E(Sex_M) &= -\frac{1.5}{2.5}\log_2\frac{1.5}{2.5} - \frac{1}{2.5}\log_2\frac{1}{2.5} = 0.971
\end{aligned}$$

As can be seen, with this approach we basically distribute the amount of uncertainty equally towards the certain cases. This has two advantages. First, it is unlikely that such a mapping will cause some significant bias as the random mapping method we described above may produce. Second, as the uncertainty is equally distributed, the overall entropy of a particular attribute will still be determined by the distribution of the values that are not generalized. Thus, the entropy calculation, and hence the choice of root node, will favour those attributes that have incurred less generalizations. This is intuitively welcome as using such attributes will lead to a decision tree that is more likely to resemble the tree that would be learnt from the original data.

B. Converting Generalized Values

Once a root attribute is selected, we need to use its values to partition the dataset into subsets, and then continue to build the decision tree with these subsets. As we keep generalized values in entropy calculation but do not wish to label branches of the tree with generalized values, we need to consider how to map a generalized value to one of its original values when we create a decision tree. There are alternative solutions:

- **Random Selection.** Out of the set of possible values, we can select one value at random to become the instance's value in this attribute and use it to create the tree. This would mean that each time the algorithm is run, different values could be assigned to each instance, resulting in varied final decision trees. Note that this randomness is different from mapping a generalized value to a original value randomly before entropy calculation: this mapping will affect mainly how tuples in the dataset may be partitioned, rather than the choice of attributes to be the root of a subtree.
- **Highest Frequency.** Out of a generalized set of possible values, we can select the value that occurs the most throughout the dataset to become the instance's value. While this method is plausible, it could result in some values that would never get selected, and as such their representation in the decision tree could be skewed.
- **Replicate Original Distribution**. We can calculate the distribution of the values of a generalized value in the original dataset and replicate this during the assignment of instance's value. For example, a distribution of 45% males and 55% females in the original dataset would be maintained when generalized values are mapped to original values. This method would require the data sanitization process to release some relevant statistics. Furthermore, there would be no guarantee that generalized values are mapped back to the correct instances in the original dataset, even though the overall distribution is maintained.

In this paper we use the random selection method to map a generalized value into a single value to be an instance's value when building a tree. Note that all these methods will have an element of randomness in attempting to map a generalized value to its original value, and this level of randomness or uncertainty should be expected as otherwise our data sanitization method would not have done its job. This could result in a slightly different decision tree being created each time when the algorithm is run on a given dataset. However, given that our objective is to study how much degradation of classification accuracy may result from set-based generalization, a basic method such as random selection is useful as it would serve as a baseline study for future work to be built upon. That is, any improvement on the mapping process will only improve the quality of the decision tree learnt from the sanitized data.

V. Experimental Results

In this section we report some preliminary experimental results. Our general approach is shown in Fig. 1. We begin with a dataset to be classified. Using this dataset as training data, a decision tree will be derived. The dataset will then be anonymized using a chosen privacy preservation technique to produce its sanitized form. The sanitized dataset will then be used as training data to derive a second decision tree. Both decision trees will be used to classify the same test data in order to compare their classification accuracy.

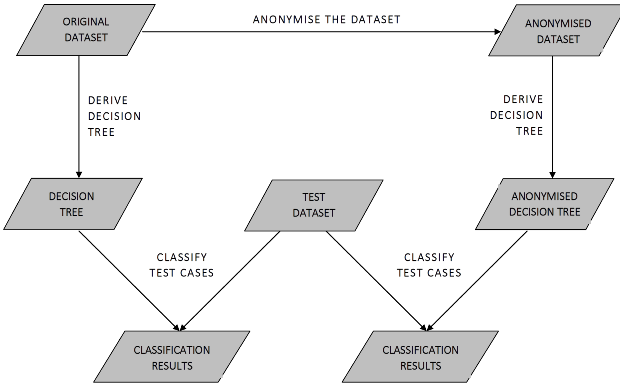

Fig. 1. Experiment Approach

For decision tree derivation, we implemented our ID3 algorithm based on a version given in [11] and the implementation of the Mondrian algorithm followed the one given in [8]. Experiments were performed on *Adult* [10], a real-world dataset which is widely used in privacy protection studies. We compare classification accuracy of the decision trees derived from the original dataset and its anonymized version.

The *Adult* dataset contains 6 numerical and 8 categorical attributes. Similar to other studies, we removed all the instances with missing values, resulting in 30,162 training cases and 15,060 test cases. The dataset was anonymized using the Mondrian algorithm with different numbers of QIDs and varying values of k, but only categorical attributes were used in deriving decision trees due to the nature of the ID3 algorithm. For each setting, we ran our experiments three times and average their classification accuracy. This is necessary because our modified ID3 algorithm is not deterministic in dealing with generalized values, and can generate slightly different decision trees in different runs.

A. *Effect of Varying k*

We first tested our modified ID3 algorithm by varying the values of k from 4 to 64 while fixing the QIDs to $\{$*Age, Education, Hours-Per-Week, Native-Country, Capital-Gain, Work-Class*$\}$ which is the same setting used by Inan et al. [6] in their study. We ran our ID3 algorithm on the original dataset to derive a decision tree and used this decision tree to classify the test instances to obtain an accuracy of 80.96%. This is then used as a benchmark to compare to all other tests to see how classification accuracy was affected by anonymization. The results are shown in Figure 2.

As can be seen, an increase in the value of k led to a decrease in classification accuracy. This is expected as a lower k will require less generalization to be made to the data. However, it is worth observing that when $k = 4$, the accuracy is only decreased by 0.09% when compared to the

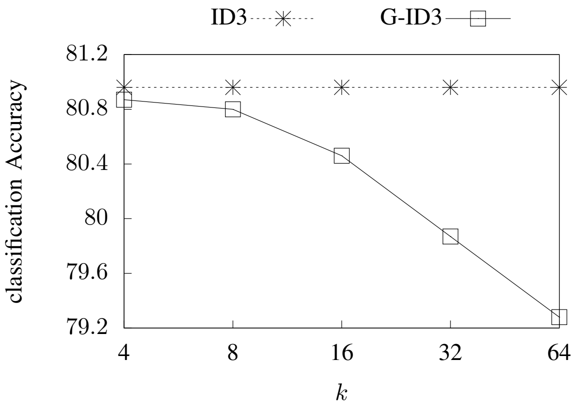

Fig. 2. Classification accuracy with varying k

classification accuracy using the original dataset, and when $k = 64$, a decrease of accuracy was only at 1.68%. The maximum variation in classification accuracy in different runs is only 0.56, so the fact that our solution involves an element of randomness and can generate slightly different trees each time when it is run on a given set of data does not seems to impact our classification accuracy. These are significant as $k = 5$ is often recommended in practice for k-anonymizing a set of data [2], so a good performance at $k = 4$ is important. On the other hand, setting $k = 64$ for the size of *Adult* dataset is equivalent to a case where a smaller k is used for a smaller dataset, that is, a case where substantial generalization on original data is required due to the size and k ratio. As our experiments show, in all the cases we are able to derive decisions trees from anonymized data with no more than 2% degradation in classification accuracy.

We have also observed from experiments that due to the heuristics used by the Mondrian algorithm in sanitizing the data, values are usually generalized only with one or two other values. As such, the likelihood that a generalized value will be mapped back to its original value during the creation of the decision tree is relatively high. Therefore, the semantic relationships between the attributes can be preserved, leading to a classification accuracy which is similar to the accuracy obtained through the use of the original dataset.

B. Effect of Varying QIDs

We then tested our modified ID3 algorithm by varying the number of QIDs while fixing the value of k at 64. In varying the QIDs, we start with the set of 6 QIDs used in the varying k tests. We then added one extra attribute as QID at a time, starting with a categorical one that has most distinct values and ending with the numerical attributes. The *Fnlwgt* and *Education-num* attributes were not used in this test because *Fnlwgt* is not useful from the classification point of view and *Education-num* carries similar information to the *Education* attribute. Therefore, the maximum number of QIDs used in this investigation was 12.

By selecting the QIDs this way, we hope to test the effect of using more QIDs better. This is because with more distinct values, it is likely that more generalization will be exercised during data sanitization and there will be a lesser chance of generalized value being mapped back to their original value during the creation of a decision tree. Thus, selecting QIDs this way will test how the resultant classification accuracy will be affected, even when the least number of QIDs is used. The results are shown in Figure 3.

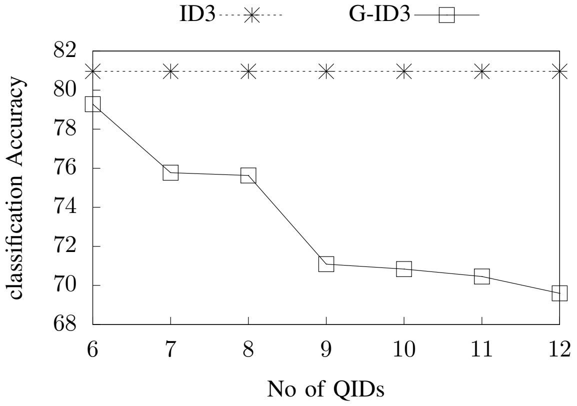

Fig. 3. Classification accuracy with varying QIDs

As can be seen, as the number of QIDs used in the anonymization process increases, the classification accuracy decreases. This again is expected since the more QIDs that are specified, the rarer the value combinations become across these attributes. This means that the combinations of values appear less frequently, requiring more generalization to be carried out in order to enforce k-anonymity. This in turn leads to more instances to have generalized values and more values to be included in a generalized value. Both of these result in a lesser likelihood that in the creation of a decision tree, a generalized value will be mapped back to the original one, hence lowering down classification accuracy since the semantic relationships between the attribute values may have been distorted. Again, the maximum variation in classification accuracy in different runs was relatively small at 2.16.

To understand the amount of classification accuracy decreasing with each addition of a new QID, we analyzed the decision trees produced from the anonymized datasets from different runs. It was observed that with the addition of each new QID, the decision tree tends to favour an attribute that has not been selected as QID, as the root of the tree. Table IV shows the number of QIDs used for each anonymized dataset along with the attribute that was added to the QID list each time and the attribute that was selected as the root of the decision tree.

The first six QIDs shows that the best attribute for the root is *Relationship*. This continued with *Occupation* and *Marital Status* are added as QIDs. The values of *Relationship* have still not been generalized at this point, though the classification accuracy decreased when more QIDs were used. The *Relationship* attribute was then generalized when it was

TABLE IV
Addition of New QID

Number of QIDs	New QID	Root
6		Relationship
7	Occupation	Relationship
8	Marital Status	Relationship
9	Relationship	Sex
10	Race	Sex
11	Sex	Native Country
12	Capital Loss	Native Country

used as a QID too. At this point, the decision tree no longer used *Relationship* as its root and instead opted for *Sex*. The same pattern persisted with the remaining QIDs being added. i.e *Sex* continues to be selected as the root of the decision tree until it is added as QID, and *Native Country* is selected when it is not used as a QID. This confirmed our intuition that the way we deal with generalized values in entropy calculation should favour attributes that are not generalized. Due to this, the decision tree is able to maintain reasonably satisfactory classification accuracy despite an increasing number of QIDs used.

From looking at the chart in Figure 3, it appears that there are not big differences in classification accuracy between some consecutive numbers of QIDs. For example, the difference between using 7 and 8 QIDs is a decrease in classification accuracy of only 0.14%. However, it does look that including more attributes as QIDs may not result in uniform degradation of classification accuracy, suggesting that the quality of a decision tree learnt from sanitized data is also related to the characteristics of the QIDs that were added.

VI. Conclusions

The majority of the existing work on data privacy protection focus on developing privacy models and data sanitization methods. Relatively little has been done to consider how data analysis techniques may be adapted to work with sanitized data. In this paper we have reported a study on how a simple decision tree learning algorithm may be designed to work with set-generalized data that satisfies k-anonymity. Our results show that good classifiers could be build even with some simple adaptation. Thus, we believe that current solutions on data sanitization is useful and sanitized data can reasonably be expected to support data analysis tasks well.

Our work is at a preliminary stage and much more could be done. Our entropy calculation and generalized data mapping are relatively basic, and more advanced methods can help improve the accuracy of learnt classifiers. There is also an opportunity to study how other data analysis techniques may be adapted to work anonymized data.

References

[1] J-W. Byun and A. Kamra and E. Bertino and N. Li. Efficient k-anonymisation using clustering techniques. Proceedigs of rhe 12th International Conference on Database Systems for Advanced Applications, pp 188-200, 2007.

[2] K. E. Emam and F. K. Danlar. Protecting Privacy Using k-Anonymity. JAMIA, 15(5), pp 627-637, 2008.

[3] B. C. M. Fung and K. Wang and P. S. Yu. Top-Down Specialization for Information and Privacy Preservation. Proceedings of the 21st International Conference on Data Engineering, pp 205-216, 2005.

[4] B. C. M. Fung and K. Wang and R. Chen and P. S. Yu. Privacy-Preserving Data Publishing: a survey of recent developments. ACM Computing Surveys. 42(10), pp 14:1-14:53, 2010.

[5] X. Han and M. Wang and X. Zhang and X. Meng. Differentailly private top-k query over map-reduce. Proceedings of 4th international workshop on cloud data management. pp 25-32, 2012.

[6] A. Inan and M. Kantarcioglu and E. Bertino. Using Anonymized Data For Classification. Proceedings of the 25th International Conference on Data Engineering, pp 429-440, 2009.

[7] M. Last and T. Tassa and A. Zhmudyak and E. Shmueli. Improving accuracy of classification models induced from anonymized datasets. Information Sciences: an International Journal. vol. 256, pp 138-161, 2014.

[8] K. LeFevre and D. J. DeWitt and R. Ramakrishnan. Mondrian Multidimensional k-anonymity. Proceedings of the 22nd International Conference on Data Engineering, 2006.

[9] G. Loukides and A. Gkoulalas-Divanis and J. Shao. Efficient and flexible anonymization of transaction data. Knowledge and Information Systems, 36(1), pp 153–210, 2013.

[10] D. Newman and S. Hettich and C. Blake and C Merz. UCI repository of machine learning databases. 1998.

[11] C. Roach. Building Decision Trees in Python. *O'Reilly Media*. 2006.

[12] L. Sweeney. Datafly: a system for providing anonymity in medical data *Database Security XI, IFIP Advances in Information and Communication Technology*. pp 356-381, 1998.

[13] L. Sweeney. k-anonymity: a model for protecting privacy. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems. 10(5), pp 557-570, 2002.

[14] X. Xiao and Y. Tao. Anatomy: Simple and Effective Privacy Preservationy. Proceedings of the 32nd International Conference on Very Large Data Bases, pp 139-150, 2006.

[15] J. Xu and W. Wang and J. Pei and X. Wang and B. Shi and A. W. Fu. Utility-Based Anonymisation Using Local Recoding. Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery And Data Mining, pp 785-790, 2006.