Tree Traversal to Achieve Generalization for Data De-identification

Jose C. D. Hernandez, David A. Garcia, and Farshad Rabbani

***Abstract*—Every day data is published of different types and from various sources. Data de-identification protects the privacy of most of this data before its publication. Over the recent years, a technique proposed by Dr. Sweeney, known as k-anonymization as a means for privacy protection has gained great popularity. There has been intensive research involving this method and many alterations, in the hope to find an optimal solution in real-time to the generalization problem. To achieve either generalization or suppression, researchers have used different types of heuristics, most of them being tree-based. Although this is a heavily investigated area, there is no simple method to prepare data for generalization; in theory, there are infinite methods for data preparation and partitioning. In this research, we first propose the use of commonly known algorithms to prepare data and achieve generalization. We then introduce the use of tree-based algorithms and tree traversal as the mechanism to achieve data generalization. We further investigate them, by comparing the quality of generalization sets obtained in each traversal method, in the hope to determine which method is best.**

***Keywords*—Privacy, De-identification, Re-identification, Data Generalization, Generalization Heuristics, Tree Traversal, Data Transformation, k-anonymity, Privacy Metrics, Data Publication.**

I. Introduction

As the era of big data, the twenty-first century has more than 2.5 quintillion data [1] generated each day from various sources. Every day the massive amounts of data that is collected and published are primarily due to the advances and unquestionable development of information technologies in recent years [2]. Governmental, local and international institutions, either public or private are increasingly required to make their data electronically available [3], either to satisfy some legal requirements or as part of some business process [4]. When that data becomes publicly available, it can be used by anyone and for any purpose. Government agencies and commercial industries [4], like insurance companies, health organizations, retailers, are just a few of the many entities that can make use of publicly available data. These companies are free to use the public data in literally any process they desire.

Data is always collected or published for specific purposes, which led to a delicate trade-off between the use of digital data (for security, progress, research, competitiveness) and the privacy of the citizens and other entities involved in its release [5]. To address this privacy issue, data publishers usually apply data privacy mechanisms to the data before its public release. These mechanisms must be implemented correctly otherwise, it can lead to severe ramifications on the privacy of the different entities involved [6]-[8]; evidenced by high profile cases [9]: the Massachusetts medical records released in the mid-1990s, the AOL Research released, and the Netflix Prize Competition in 2006, among others. When not appropriately sanitized, the public release data may unwittingly disclose sensitive [10], or personal information [11] to the public and the chance of misuse of the data is extremely high [12], [13].

For the public release data to be useful, there needs to be a proper balance between the privacy risks and data usability left on the release version [14]; which is no easy task to address. This difficulty of preserving privacy in data publishing has gained significant attention within the research community [15] and has caused data custodians to be more concerned about the different security issues related to data publishing. This is such a serious issue that many governments have implemented different legislations that take into account this particular issue, for example, the US Health Insurance Portability and Accountability Act (HIPAA) [16], and the European Directive on Data Protection [17], are just two of the many legislations that have taken into account data privacy protection when publishing data. Since there is no perfect method to determine that optimal balance for privacy and usability of the data, companies will usually choose a specific threshold that they are willing to tolerate [18] and will base their privacy protection on that threshold.

The most common and widespread mechanism for privacy protection in public releases of data is de-identification (also known as anonymization). De-identified data are records that have had enough personally identifiable information (PII) removed or obscured (also referred to as masked or obfuscated) with the goal that the rest of the data does not distinguish an individual and there is no sensible premise to trust that the data can be utilized to identify an individual [19]. When implemented, the data de-identification process varies from data custodian to data custodian (further details of the actual process in Section II). There are no defined guidelines

Manuscript received October 8, 2018.

Jose C. D. Hernandez is with the University of Belize, Belmopan, Belize (email: jose.hernandez@ub.edu.bz).

David A. Garcia is with the University of Belize, Belmopan, Belize (phone: +501-822-3680; e-mail: dgarcia@ub.edu.bz).

Farshad Rabbani is with the University of Belize, Belmopan, Belize (e-mail: frabbani@ub.edu.bz).

to achieve de-identification. There are however different techniques (described in Section II-A) and methods (described in Section II-B) that aid in the implementation.

From the many privacy protection techniques, generalization is considered one of the most popular methods to achieve de-identification of data [7], [20]. That is why in this research, our privacy implementation technique will be data generalization. However, the actual process of achieving generalization (described in detail in Section II-A1) can also vary, and there are many ways to accomplish it. The different methods are referred to as heuristics, or more specifically, data generalization heuristics. A heuristic is an approach or algorithm that leads to a correct solution of a programming task by non-rigorous or self-learning means [2]. In this research, we will be working with only non-semantic heuristics; ignoring the semantics of the generalized data in its totality.

Most non-semantic heuristics utilize the tree abstract data type to generate the different generalization sets, even though in the literature their actual structures are not detailed. In this research, we implement basic tree-based algorithms to achieve data generalization and try to determine which type of tree traversal is better when using heuristics that generate a tree for data de-identification. To properly accomplish this, we have conducted several experiments outlined in Section V of this paper.

II. PRIVACY RISKS IN PUBLIC DATA

Data publication is a must in every sector, and as technology progresses, the demand will increase. Currently, public data is available for almost every area imaginable; medical healthcare datasets (for example, clinical studies, and hospital and discharge databases), demographic datasets (for example, census and sociology studies), systems data (for example, logs and bugs reports), social network (for example, Facebook and Twitter), e-commerce (for instance, Netflix movie ratings and Amazon products details) and the list continues. A fictional case is used to illustrate the process and reasoning of data de-identification. Assume that a clinic is considering the release of patient record information. This information is described in Table I.

TABLE I
PATIENT RECORDS

ID	Name	Age	Sex	Zip code	Disease
10001	JAMES	30	Male	881001	AIDS
20002	JOHN	25	Male	881002	Cholera
30003	MICHAEL	20	Female	881003	Dengue
40004	ROBERT	25	Male	881004	Hepatitis
50005	MARY	20	Female	881005	Malaria

TABLE II
PATIENT RECORDS WITHOUT PIIS

Age	Sex	Zip code	Disease
30	Male	881001	AIDS
25	Male	881002	Cholera
20	Female	881003	Dengue
25	Male	881004	Hepatitis
20	Female	881005	Malaria

The clinic decided to remove all the PIIs, in this case, the columns name, and ID, producing Table II. This new table is then published, however with the use of other existing data, that is available also publicly, for example, voter records, it is possible to identify the owner of the records, by combining the different fields in each table. This action is likely due to the presence of quasi-identifiers (QIs) [6], [7], [22]-[24] in the data release. Quasi-identifiers [10] are fields combined with information acquired from other sources to reveal the original data, such as {Age, Sex, Zip code}. That process (the use of QIs to reveal the original data) is known as re-identification of data [11]; which is one of the significant effects of identification risk [25]-[28] associated in every public release of data. Re-identification is also defined as merely the reverse process of de-identification, to obtain the original data [11]. Identification risk associated with every data release is the probability of a specific entity to be identified by the use of public data [26]. The term privacy, in its broadest sense, is the right of an individual to have specific details about them not available to the public [25].

Before making data publicly available, to prevent privacy-related issues in data releases, organizations sanitize the information with techniques such as de-identification [29]. The objective is to make the data anonymous and reduce privacy-related risks [27]. Data anonymization (also known as de-identification) is just one of the many techniques used to sanitize a dataset for public release; sanitizing it in a way that prevents attackers from breaching the privacy of the different entities in the public release [23]. The following section outlines some of the many different techniques that can be used to transform the original data into de-identified data.

A. Privacy Protection Techniques

The following are the most common techniques used to achieve data de-identification. Generalization is a process of making information less precise, for example, by the grouping of continuous values. Suppressing Data is accomplished by deleting an entire record, or certain parts of records. Introducing Noise into the Data is a technique that adds small amounts of variation to the selected data. Swapping the Data exchanges the data contained in the same data field from different but comparable records (e.g., swapping the postal codes of two records). Replacing Data with the Average Value is a method that replaces data field values with the average value of the group of data to which the data field belongs. From all of these techniques, Generalization is considered one

of the most popular methods to achieve de-identification of data [7], [20]. The following section describes this method in greater detail.

1) Data Generalization

Data generalization is the process of making information less precise, such as grouping continuous values and replacing the original value with a group of values [30]; this approach generates generalization sets (GSs) that replaces the actual data values. If we look at Table II, by grouping the values in the age column, the grouping replaces the actual values in that column.

For example, the age of patient Mary can be generalized to {20-25} or {10-50} or {19-21} or any other selected range. Inherently, this is why generalization cannot stand alone, primarily because there are no limits to define. Limits like, how big are the GSs? Which GSs replaces which values? To answer these questions and to provide different limits, Professor Latanya Sweeney in 2002 came up with a concept referred to as k-anonymity; a model for privacy protection in data releases [23]. However, Meyerson & Williams [31] proved that optimal generalization sets for k-anonymity is NP-hard, under both the generalization and suppression models.

Although, the k-anonymity model has drawn considerable interest in the research community for the last few years resulting in many proposed algorithms that either modify or extend its capabilities [32] (to view the various k-anonymization approaches, see [33]); the actual process of obtaining the GS varies, since its primary property utilizes either generalization or suppression.

2) K-Anonymity

The primary goal of k-anonymity is to prevent an adversary from identifying an individual with a probability of 1/k [34]. Therefore, the probability of an adversary being able to find out the identity of a k-anonymized tuple is at most 1/k [34]. The k-anonymity technique uses either generalization or suppression to protect privacy. With generalization, less concrete forms replace QIs [20]. For example, value 15 in a dataset is generalized to the range {15-30}, then all occurrences of the value 15 are replaced by the range {15-30}. On the other hand, with suppression, the value is completely removed, and instead of the value, something ambiguous, like asterisks is put in its place [20].

One of the most popular methods studied by the community for k-anonymity has been homogeneous generalization [34]. This approach entails partitioning of the tuples in the table under study into groups called equivalence classes which are also known as generalization classes or sets (GS). The entire GS replaces all values found in each GS, or a range developed based on the GS generated during the partitioning process.

Returning to our example of Table I, with the assumption that the data custodian is willing to tolerate a threshold of 2, using k-anonymity, yields at least two records of each, repeated throughout the entire table. The first step is to generalize the data into different sets, in our example, the complete data is H= {30, 25, 20, 25, 20}, the different GS generated will be directly related to the heuristic used to achieve generalization.

In our example, age 25 and 20 already meets the k=2 requirements. Therefore, those values are left alone. However there are no values to group age 30 with, but upon executing heuristics A, the resulting GSs were, A= {20}, B= {25, 30} and those GSs replaces the actual values, as displayed in Table III.

TABLE III
AGE K-ANONYMITY, WITH K=2

Age	age (2-anonymity)
30	{25,30}
25	{25,30}
20	{20}
25	{25,30}
20	{20}

Note that, if heuristic B was executed instead of heuristic A, the GSs might have been A= {20, 30}, B= {25} and perhaps heuristic C would have had results A= {20, 30, 25}. These multiple options pose the question – which heuristic is best? This uncertainty is indeed a problem. This paper proposes the use of commonly known tree base algorithms to prepare data and then traverse them to obtain the data order for generalization.

III. TREE-BASED GENERALIZATION

For this paper, we will be exploring three commonly known algorithms; (1) The Huffman Coding Algorithm [35], (2) The Binary Search Tree Algorithm [36] and (3) The Georgy Adelson-Velsky and Evgenii Landis' Tree Algorithm, also known as an AVL Tree [37].

The Huffman Coding Algorithm solves the problem of finding an optimal codebook (variable length codes) for an arbitrary probability distribution of symbols [35]. That is, while there might exist many other codebooks, none will have a smaller average code length [35] than that generated by Huffman's - the primary reason for choosing this heuristic. Huffman provides detailed instructions on how to construct Huffman Trees such as that shown in Fig. 1 [38].

A *Binary Search Tree* is a rooted binary tree, whose internal nodes each store a key and is recursively defined as either being empty or consisting of a node called the root (top), together with two rooted binary trees called the left and right subtrees, respectively. Those sub-trees can both be empty.

What makes this data structure special, is the fact that it satisfies the binary search property; binary search requires fast access to two elements, specifically the median elements above and below the given node [36]. The order among sibling nodes matters in rooted trees. Therefore the value that represents the left node is different from that of the right [36]. This characteristic that for a given node X, all nodes in the left subtree of X have keys < X, while all nodes in the right subtree of X have keys > X [36], of this heuristic, made it an obvious choice.

On the other hand, an *AVL Tree* is a self-balancing binary search tree where the heights of child subtrees of any node

differ by at most one; if at any given time they differ by more than one, it will automatically re-balance itself in order to maintain that property [39].

A. Tree-Based Heuristics Construction

To achieve generalization with these algorithms the frequency of each unique value in the dataset is used to build the trees.

For example, for the dataset of J= {47, 42, 51, 35, 33, 51, 37, 33, 38, 36}, [x] represents the frequency of a given value. Outlined below are the trees generated using the heuristics proposed:

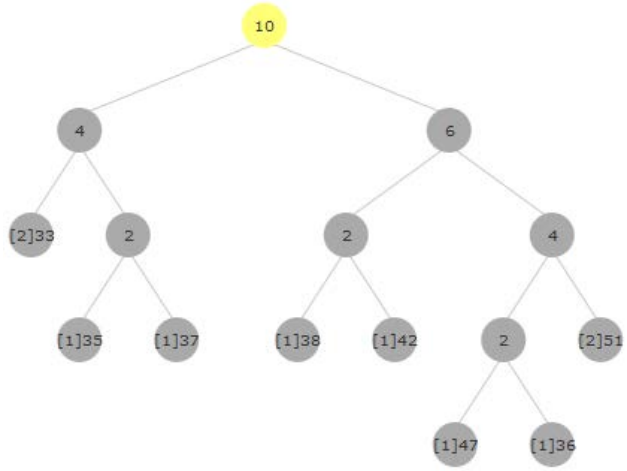

Fig. 1. Huffman Tree

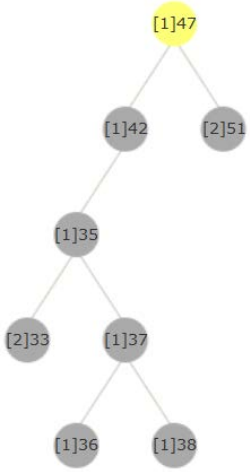

Fig. 2. Binary Search Tree

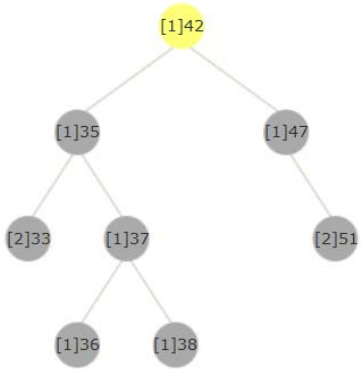

Fig. 3. AVL Tree

Once constructed, each tree should be traversed [36] by either; (1) Pre-order traversal, (2) In-order traversal, or (3) Post-order traversal. By ignoring all those values whose frequency is node>=K, and by traversing only those values that are of frequency node<K; the resulting data from the tree traversals are k-anonymous. The traversal result will determine the data order, once that has been obtained it is just a matter of getting the different GSs based on the K specified.

B. The Generalization Algorithm

There are many methods to achieve the actual division of the different generalization sets, for this paper and this research, we have developed a generalization algorithm based on three different concepts derived from the classical problems of the Knapsack family of algorithms [40]: (1) Bin Packing, (2) Partition Problem, and (3) the Subset Sum problem.

These three classical problems can be summarized as follows: the Bin Packing problem solves the scenario of packing objects of different volumes into a finite number of bins of specific capacities in a way that produces a minimum for the number of bins utilized. The Partition Problem, also known as number partitioning, basically carries out the task of deciding whether the partitioning of a given set of positive integers can yield two subsets of the equal sum. The Subset Sum problem determines if there is a subset of a specific sum in a master set.

Our algorithm follows the following logic:

Given a dataset $E=\{e_1,\ldots,e_T\}$ where "T" denotes the total number of elements. $tam(E)=tam(e_1)+tam(e_2)+\ldots+tam(e_T)$; where "tam" is size/frequency; therefore $tam(e_i)=n_i$; where "n_i" is the total of the individual element. Therefore, a partition is defined as; "P_j" given that $P_j=\{e_{1j},e_{2j},\ldots,e_{aj}\}$; which is a subset of "E" with "a" elements. This leads to the definition of $tam(P_j)=tam(e_{1j})+tam(e_{2j})+\ldots+tam(e_{aj})$; given a "K" where $K \in N$ and $K \leq tam(E)$.

With the objective to minimize the function $tam(P_i)\ \forall_{i=1,\ldots,M}$ and to find $M \in N$; referred to as "Number of partitions" that satisfies the following properties: $P_1 \cup P_2 \cup \ldots \cup P_M = E$ and $P_i \cap P_j = \emptyset\ \forall_{i \neq j}$; we will refer to it as "the equation to determine partitions for M"; which are subject to: $tam(P_1)+tam(P_2)+\ldots tam(P_M)=tam(E)$ and $tam(P_i) \geq K\ \forall_{i=1,\ldots,M}$; which will determine the elements to be generalized given a

particular K.

In summary, our algorithm obeys the following four rules:

- Rule 1: Partition into sum => k, where K is any given integer.
- Rule 2: Merge the last incomplete partition with the one above to maintain cluster closeness.
- Rule 3: When the entire set is not able to achieve sum => K, then request the first node that is = k and then merge this node with the rest.
- Rule 4: If rule four returns null, then request first node >k and merge set with it.

The resulting datasets from this algorithm are the results of the tree traversals. There are different methods of traversing a tree; the question that arises is: if one traversal is better than the others? To answer such a question, a mechanism to measure the results is needed. The following privacy metric will be used to evaluate the quality of the different generalization sets obtained by the different traversal methods.

IV. PRIVACY PROTECTION METRICS

In general, the quantification used to measure data privacy is the degree of uncertainty, according to which original private data can be inferred [41]. The degree of uncertainty is the amount of private information that can be discovered from data mining results [41], for example, the degree of uncertainty for the k-anonymity is at least 1/k. The higher the degree of uncertainty achieved by an algorithm, the better the data privacy is protected by that algorithm [41]. For this paper, we have chosen the following two metrics.

A. Information Loss Metric

Information loss (IL) is used to measure the amount of information lost due to k-anonymization. There are a variety of methods that have been proposed and used during the past years [4], [28], [33], [41]-[44]. The measurement of information loss in this research will be done using the one described by [4], which was later used by [45] and [32]. From all the different metrics available, the one by [4] is considered most appropriate due to the properties it uses to measure the information loss; which is based on GS and not the different levels of generalization hierarchies (GH), this study will generate GS and not GH. Thus this metric is considered the most appropriate in this particular scenario.

The idea behind this metric is to determine the information loss by each anonymized value and then average all the individual information loss to determine the total information loss per the given column. This research uses the same assumption utilized by Iyengar [4] when measuring information loss. That assumption is that there is the same generalization loss for ambiguity between any two distinct values.

For example, given the column age, which contains the following dataset age = {24,46,51,28,28,23,55,45,35,40}, this dataset has 9 unique values. When the dataset is anonymized using K=4, we obtain; GS1 = {24,40,55,51} and GS2 = {45,23,28,46,35}. To obtain the amount of information loss per GS, equation 1

$$\text{Information Loss per GS} = \frac{\text{Amount of GS elements}}{\text{Unique Values in dataset}-1} \quad (1)$$

is used which in return will give us; $\text{GS1} = \frac{4}{9-1} = \frac{4}{8} = 0.5$ and $\text{GS2} = \frac{5}{9-1} = \frac{5}{8} = 0.625$. In this case, the information loss for the column age is given by equation 2, which returns 0.575 for this example.

Information Loss per Column =
Average of all IL per GS used in column (2)

B. Utility Loss Metric

Measuring the utility of the released data is another tough task [46]. This metric measures the amount of information retained in a generalized table [47]. The utility of an anonymized table increases as the distance between the anonymized table and the original table decreases [47]. The utility depends on the variability of the confidential attribute within the group: the more similar the confidential attribute values, the more knowledge for the user (when the anonymized data makes more sense), but also the higher the risk of attribute disclosure [46].

Several metrics have been proposed to measure utility, also referred to as measurements of data quality [48], with the most popular ones being the discernibility metric [24] and the misclassification metric [4]. Both of these metrics assign a penalty to each tuple based on how many tuples in the transformed dataset are indistinguishable from it [41]. In 2012, a new metric known as research value (RV) was proposed by [49]; which encapsulates the utility of each attribute concerning specific conditions.

For this study, the utility metric will be the Hierarchical Discernibility Metric (HDM) [50] due to the many improvements described by [50] over the discernibility metric. The fact that HDM evaluates the utility based on generalization class and not on the generalization of hierarchical levels makes it most appropriate for this work. The following example outlines HDM.

Assume there are 50 records 'X' and 200 records 'Y'. If a GS has the two values G1= {X, Y} (the GS created using X and Y) and there is a total of 1000 records, the calculation of the HDM for each value uses equation 3.

$$\text{HDM}v = \frac{(\text{N}e - \text{N}v)}{(\text{N} - \text{N}v)} \quad (3)$$

Where **N** is the total number of records.
$\mathbf{N_e}$ is the number of records that have values in the group e.
$\mathbf{N_v}$ is the number of records that have value v.

For this example, $N = x + y = 50 + 200 = 250$; using N=250 $\text{HDM}x = \frac{250-50}{1000-50} = \frac{4}{19}$ (the utility loss for value X) and $\text{HDM}y = \frac{250-200}{1000-200} = \frac{1}{16}$ (the utility loss for value Y).

Therefore 'X' has a HDM of 4/19, where 'Y' has a HDM of 1/16. Now, to determine the total HDM for G1 all HDM per values within the GS are average together [50], thus HDM_{G1} is determine by equation 4.

$$HDM_{GS} = \frac{\sum_{i=0}^{n} n_i}{n} \quad (4)$$

Where **n** is the total number of elements.

In this example, the total DDM (using equation 4) for G1 is $\frac{4/19 + 1/16}{2} = 0.136516$. If this example had resulted in more GSs, then the total utility loss per the entire table would be the average of each GS [51].

V. EXPERIMENTS

Besides demonstrating that basic algorithms can be used to achieve generalization via tree traversal, we were also interested in determining which, if any is the best method of traversing a tree. Best in the sense that it achieves the k property requested and yet, loses less information and utility compared to the other methods of traversing the tree.

A. *Experiment Setup*

To achieve such a task, we designed, developed and ran two "3X3X3" factorial designs with three independent variables and two dependent variables. The independent variables were (1) the method of traversing a tree, (2) the heuristics and (3) the value of K.

The heuristics, in this case, were: Huffman coding, Binary Search Tree, and AVL tree. The values of K used were; 3, 5, and 10; chosen because they were the common values used by [28], [43], [49] in their experiments using k-anonymity and the same dataset utilized by this research.

The data source for these experiments was the Adult dataset from the UC Irvine Machine Learning Repository [52]; stored in a MySQL database. This dataset is a de facto benchmark for k-anonymity related experiments and investigations [53]. Some researchers have gone as far as to call this dataset a standard benchmark for k-anonymization studies [20]. Possible columns under consideration from the dataset (QIs) are the following; age, work class, education, occupation, hours work per week, and native country. Those fields contain enough data to evaluate the generalization outcome.

We used only the columns age and hours work per week for the experiments; each one with 80 runs, which generated 2160 records per column. Each traversal result underwent our generalization algorithm described in Section III-B and was evaluated using the privacy metrics described in section 4. The following section provides an outline of the analysis of the resulting data.

B. *Results*

Our results showed that for both, the age and hours work per week (H/W) column, the in-order traversal method produced less information and utility loss, shown in Fig. 4-7; Table IV and V are summaries of the results.

TABLE IV
TREE TRAVERSAL RESULTS FOR COLUMN AGE

Traversal Method	Information Loss		Utility Loss	
	Mean	StDev	Mean	StDev
In-Order	**0.044998**	**0.001180**	**0.009517**	**0.004366**
Post-order	0.045579	0.001140	0.009846	0.004626
Pre-order	0.046576	0.003001	0.010190	0.004637

TABLE V
TREE TRAVERSAL RESULTS FOR COLUMN H/W

Traversal Method	Information Loss		Utility Loss	
	Mean	StDev	Mean	StDev
In-Order	0.081889	0.012812	0.009763	0.005515
Post-order	0.082183	0.013070	0.010015	0.005759
Pre-order	0.082719	0.013527	0.010436	0.006295

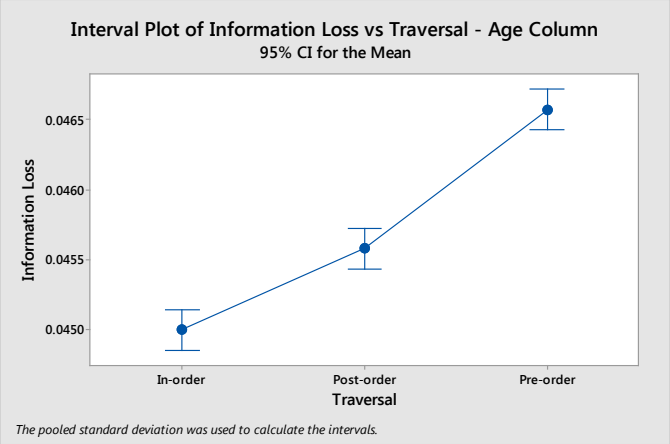

Fig. 4. Information loss due to traversal for column age

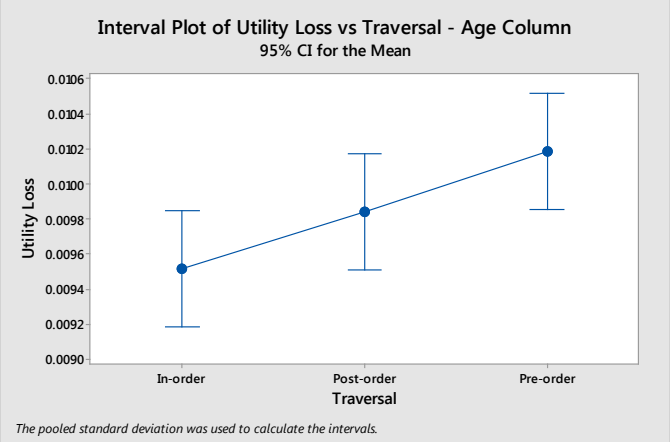

Fig. 1. Utility loss due to traversal for column age

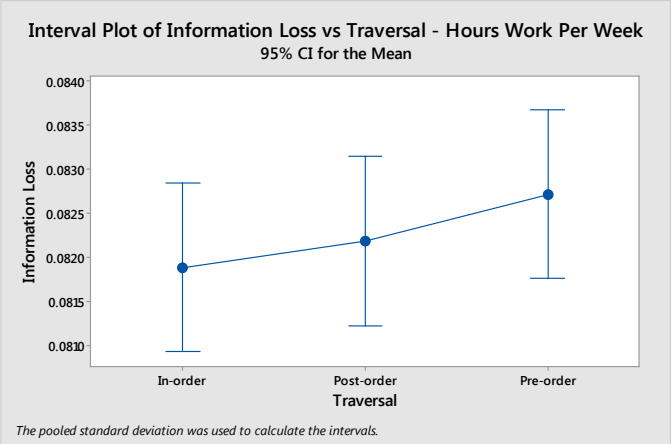

Fig. 6. Information loss due to traversal for column H/W

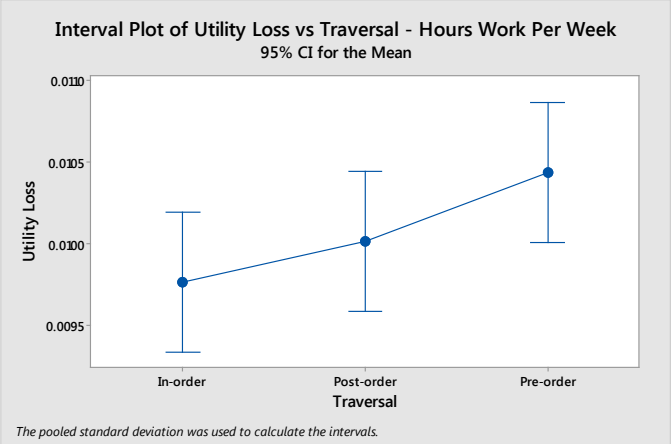

Fig. 2. Utility loss due to traversal for column H/W

VI. DISCUSSION

We have demonstrated that by using any tree-based algorithm, it is possible to achieve generalization. Based on the results of the experiments, it is clear that the method used in traversing the tree for data preparations plays a significant role in the quality of the resulting generalized data as shown in Fig. 8 and Fig. 9. At this moment, however, we are unable to state which of the three different heuristics used obtains the best generalization overall, further experiments are required to determine this.

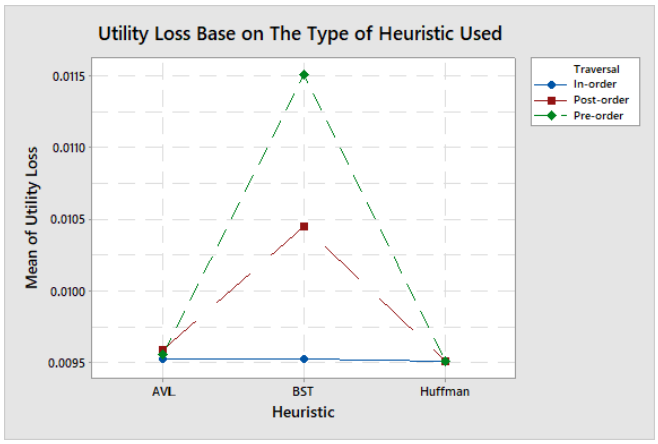

Fig. 3. Utility Loss Based on the Type of Heuristic Used

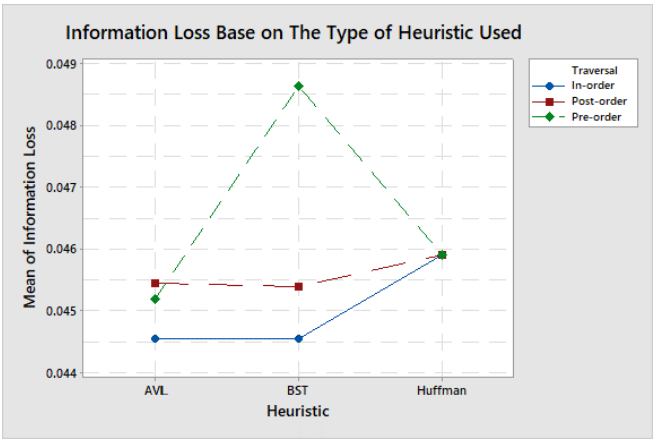

Fig. 4. Information Loss Based on the Type of Heuristic Used

An interesting fact is that with Huffman coding, the traversal method used is irrelevant; all the different tree traversals produced the same amount of information and utility loss, as can be seen in Fig. 8 and Fig. 9. This fact was true for both experiments. For all other heuristics in the study, the traversal method did give different results. We believe this was due to the property that Huffman coding has, which is the ability to generate new nodes by combining two existing nodes.

Once obtained, the GSs can in return replace the actual values and thus achieve de-identification. Each record in the de-identified dataset has a probability 1/k of being re-identified [43].

The importance of knowing which traversal method is better can be instrumental when making data generalization by the use of tree-based heuristics because it will permit the data custodian the ability to use the traversal method that will guarantee the k value property, yet obtaining less information and utility loss during the process. Also to a certain degree, knowing which traversal method is better, makes the de-identified data more useful when conducting different analyses.

VII. CONCLUSION

This research was an attempt to address the critical problem of transforming data so that the dual goals of usefulness and privacy can be satisfied to a certain degree. We achieved this by investigating tree-based heuristics and determined the best method of traversing the tree to achieve generalization. Based on the results of our experiments, we can conclude with confidence that the best tree traversal method is in-order traversal; best in the sense that it has the least information and utility loss.

We have successfully implemented tree-based algorithms to achieve data generalization. We were able to demonstrate the viability of the use of traversal methods to prepare data for generalization, and we were able to show a clear difference between them.

REFERENCES

[1] K. Yang, X. Jia, K. Zhang, and others, "Privacy-Preserving Data Publish-Subscribe Service on Cloud-based Platforms," 2014.

[2] L. Jiang, *Proceedings of the 2011 International Conference on Informatics, Cybernetics, and Computer Engineering (ICCE2011) November 19-20, 2011, Melbourne, Australia: Volume 1: Intelligent Control and Network Communication*, vol. 110. Springer Science & Business Media, 2011.

[3] R. Turn, "Information privacy issues for the 1990s," in *Research in Security and Privacy, 1990. Proceedings., 1990 IEEE Computer Society Symposium on*, 1990, pp. 394–400.

[4] V. S. Iyengar, "Transforming data to satisfy privacy constraints," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2002, pp. 279–288.

[5] J. Nin and J. Herranz, *Privacy and Anonymity in Information Management Systems*. Springer, 2010.

[6] A. Manta, "Literature survey on privacy preserving mechanisms for data publishing," 2013.

[7] A. Saroha, "Survey of k-Anonymity," 2014.

[8] C. Thomas and D. Thomas, "A Survey on Privacy Preservation in Data Publishing."

[9] O. Heffetz and K. Ligett, "Privacy and data-based research," *Journal of Economic Perspectives*, vol. 28, no. 2, pp. 75–98, 2014.

[10] J. Cao and P. Karras, "Publishing microdata with a robust privacy guarantee," *Proceedings of the VLDB Endowment*, vol. 5, no. 11, pp. 1388–1399, 2012.

[11] S. L. Garfinkel, "De-identification of personal information," *NISTIR*, vol. 8053, pp. 1–46, 2015.

[12] M. Callahan, "Us dhs handbook for safeguarding sensitive personally identifiable information," *Washington, DC*, 2012.

[13] G. Navarro-Arribas and V. Torra, *Advanced research in data privacy*, vol. 567. Springer, 2014.

[14] F. Kohlmayer, F. Prasser, and K. A. Kuhn, "The cost of quality: Implementing generalization and suppression for anonymizing biomedical data with minimal information loss," *Journal of biomedical informatics*, vol. 58, pp. 37–48, 2015.

[15] B. C. Fung, K. Wang, A. W.-C. Fu, and S. Y. Philip, *Introduction to privacy-preserving data publishing: Concepts and techniques*. Chapman and Hall/CRC, 2010.

[16] B. C. Fung, K. Wang, and P. S. Yu, "Top-down specialization for information and privacy preservation," in *Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on*, 2005, pp. 205–216.

[17] Z. Huang, "Extensions to the k-means algorithm for clustering large data sets with categorical values," *Data mining and knowledge discovery*, vol. 2, no. 3, pp. 283–304, 1998.

[18] D. Riboni, L. Pareschi, and C. Bettini, "Preserving Privacy in Sequential Data Release against Background Knowledge Attacks," *arXiv preprint arXiv:1010.0924*, 2010.

[19] E. McCallister, *Guide to protecting the confidentiality of personally identifiable information*. Diane Publishing, 2010.

[20] L. Sweeney, "Achieving k-anonymity privacy protection using generalization and suppression," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 571–588, 2002.

[21] M. Press, *Microsoft Press computer dictionary: the comprehensive standard for business, school, library, and home*. Microsoft Pr, 1994.

[22] K. El Emam and F. K. Dankar, "Protecting privacy using k-anonymity," *Journal of the American Medical Informatics Association*, vol. 15, no. 5, pp. 627–637, 2008.

[23] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.

[24] A. et al. Machanavajjhala, "l-diversity: Privacy beyond k-anonymity," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, p. 3, 2007.

[25] P. Samarati, "Protecting respondents identities in microdata release," *IEEE transactions on Knowledge and Data Engineering*, vol. 13, no. 6, pp. 1010–1027, 2001.

[26] W. E. Yancey, W. E. Winkler, and R. H. Creecy, "Disclosure risk assessment in perturbative microdata protection," in *Inference control in statistical databases*, Springer, 2002, pp. 135–152.

[27] P. Samarati and L. Sweeney, "Generalizing data to provide anonymity when disclosing information," in *PODS*, 1998, vol. 98, p. 188.

[28] J.-W. Byun, A. Kamra, E. Bertino, and N. Li, "Efficient k-anonymization using clustering techniques," in *International Conference on Database Systems for Advanced Applications*, 2007, pp. 188–200.

[29] L. Willenborg and T. De Waal, *Elements of statistical disclosure control*, vol. 155. Springer Science & Business Media, 2012.

[30] S. Chawla, C. Dwork, F. McSherry, A. Smith, and H. Wee, "Toward privacy in public databases," in *Theory of Cryptography Conference*, 2005, pp. 363–385.

[31] A. Meyerson and R. Williams, "On the complexity of optimal k-anonymity," in *Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, 2004, pp. 223–228.

[32] M. E. Kabir, H. Wang, and E. Bertino, "Efficient systematic clustering method for k-anonymization," *Acta Informatica*, vol. 48, no. 1, pp. 51–66, 2011.

[33] V. Ciriani, S. D. C. Di Vimercati, S. Foresti, and P. Samarati, "k-anonymous data mining: A survey," in *Privacy-preserving data mining*, Springer, 2008, pp. 105–136.

[34] W. K. Wong, N. Mamoulis, and D. W. L. Cheung, "Non-homogeneous generalization in privacy preserving data publishing," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, 2010, pp. 747–758.

[35] S. Pigeon, "Huffman coding," *Lossless Compression Handbook*, pp. 79–100, 2003.

[36] S. S. Skiena, *The algorithm design manual: Text*, vol. 1. Springer Science & Business Media, 1998.

[37] K. Mehlhorn and P. Sanders, *Algorithms and data structures: The basic toolbox*. Springer Science & Business Media, 2008.

[38] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proceedings of the IRE*, vol. 40, no. 9, pp. 1098–1101, 1952.

[39] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2009.

[40] S. Martello, "Knapsack problems: algorithms and computer implementations," *Wiley-Interscience series in discrete mathematics and optimization*, 1990.

[41] E. Bertino, D. Lin, and W. Jiang, "A survey of quantification of privacy preserving data mining algorithms," in *Privacy-preserving data mining*, Springer, 2008, pp. 183–205.

[42] J.-L. Lin and M.-C. Wei, "An efficient clustering method for k-anonymization," in *Proceedings of the 2008 international workshop on Privacy and anonymity in information society*, 2008, pp. 46–50.

[43] R. J. Bayardo and R. Agrawal, "Data privacy through optimal k-anonymization," in *Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on*, 2005, pp. 217–228.

[44] A. Solanas, F. Sebé, and J. Domingo-Ferrer, "Micro-aggregation-based heuristics for p-sensitive k-anonymity: one step beyond," in *Proceedings of the 2008 international workshop on Privacy and anonymity in information society*, 2008, pp. 61–69.

[45] G. Ghinita, P. Karras, P. Kalnis, and N. Mamoulis, "Fast data anonymization with low information loss," in *Proceedings of the 33rd international conference on Very large data bases*, 2007, pp. 758–769.

[46] J. Soria-Comas, "Improving data utility in differential privacy and k-anonymity," *arXiv preprint arXiv:1307.0966*, 2013.

[47] C. Fang and E.-C. Chang, "Information leakage in optimal anonymized and diversified data," in *International Workshop on Information Hiding*, 2008, pp. 30–44.

[48] K. B. Frikken and Y. Zhang, "Yet another privacy metric for publishing micro-data," in *Proceedings of the 7th ACM workshop on Privacy in the electronic society*, 2008, pp. 117–122.

[49] S. M. Morton, "An Improved Utility Driven Approach Towards k-anonymity Using Data Constraint Rules," 2013.

[50] T. Li and N. Li, "Towards optimal k-anonymization," *Data \& Knowledge Engineering*, vol. 65, no. 1, pp. 22–39, 2008.

[51] G. T. Duncan and S. Mukherjee, "Optimal disclosure limitation strategy in statistical databases: Deterring tracker attacks through additive noise," *Journal of the American Statistical Association*, vol. 95, no. 451, pp. 720–729, 2000.

[52] "UC Irvine Machine Learning Repository: Adult Data Set:" [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Adult. [Accessed: 04-Jul-2016].

[53] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, “Incognito: Efficient full-domain k-anonymity,” in *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, 2005, pp. 49–60.