TRANSACTIONS ON DATA PRIVACY 10 (2017) 175–201

Measuring Rule Retention in Anonymized Data – When One Measure Is Not Enough

Sam Fletcher*, **Md Zahidul Islam***

*School of Computing and Mathematics, Charles Sturt University, Bathurst 2795, Australia.

E-mail: sam.pt.fletcher@gmail.com, zislam@csu.edu.au

Received 4 July 2016; received in revised form 12 December 2016 and 19 March 2017; accepted 10 October 2017

Abstract. In this paper, we explore how anonymizing data to preserve privacy affects the utility of the classification rules discoverable in the data. In order for an analysis of anonymized data to provide useful results, the data should have as much of the information contained in the original data as possible. Therein lies a problem – how does one make sure that anonymized data still contains the information it had before anonymization? This question is not the same as asking if an accurate classifier can be built from the anonymized data. Often in the literature, the prediction accuracy of a classifier made from anonymized data is used as evidence that the data are similar to the original. We demonstrate that this is not the case, and we propose a new methodology for measuring the retention of the rules that existed in the original data. We then use our methodology to design three measures that can be easily implemented, each measuring aspects of the data that no pre-existing techniques can measure. These measures do not negate the usefulness of prediction accuracy or other measures – they are complementary to them, and support our argument that one measure is almost never enough.

Keywords. Machine Learning, Data Mining, Privacy, Patterns, Rules, Utility Measures

1 Introduction

Finding and analyzing patterns in data is becoming increasingly important in the data-driven society of the 21st century. Technology continues to facilitate new and efficient ways of collecting data, and extracting knowledge from the data remains an active area of research. While discovering patterns or building models can be done manually using expert domain knowledge, the size and complexity of modern datasets has led to increasing reliance on computer-driven, human-independent techniques.

Data mining – the science of producing useful information from (potentially enormous) repositories of data – covers a wide range of applications. Some data mining algorithms output humanly-understandable patterns discovered in the data [1], some produce a classification or regression model that can make predictions about the future [2], and others detect anomalies in the data [3]. These techniques can be applied on a wide range of modern datasets, such as medical data, financial data, social data, and law-enforcement data, among others. In this paper, we focus on a certain kind of pattern that can be discovered in data: classification rules, also known as "decision rules" or simply "rules".

Unfortunately, sometimes there are real-world considerations that conflict with the goals of data mining; sometimes the privacy of the people whose data is being data mined needs to be considered. For example, government legislation might require a minimum level of anonymization (in other words, de-identification) of any data that could leak sensitive information about its participants. Individuals might also refuse to provide their data if strong privacy guarantees are not promised to them. The process of modifying data to preserve the privacy of each participant in the data is known as anonymization. Of course, the point of collecting the data in the first place is often to discover interesting and useful patterns, and so the preservation of privacy needs to be done in a way that also preserves the utility of the data.

"Utility of the data" can be a nebulous term, with the definition of "utility" often depending on the expected workload of the data. In this paper, we propose a methodology for designing workload-specific utility measures that evaluate rules discovered in the data. More precisely, we compare an original dataset x to an anonymized version z, and measure how the rules discovered in the original dataset have *changed* after anonymization. We define rules as a set of antecedents predicting a consequent, $\psi \rightarrow c$. We implement our proposed methodology with three straightforward measures of rule retention, but they are by no means exhaustive. Which rules a data owner deems valuable can vary wildly depending on the owner's context and goals, and it would be misguided to blindly apply a "one size fits all" measure to all privacy-preservation scenarios, devoid of context. Our methodology enables data owners to design customized measures to meet their needs.

1.1 Our Contribution

Our contribution can be summarized as follows:

- We propose a novel methodology for measuring the rule retention of a dataset after it has been anonymized with a privacy-preservation technique.
- We implement and test three measures that use our methodology and demonstrate their sensitivity to changes in rule retention.
- Using a thought experiment, we demonstrate that other pre-existing measures are not sensitive to changes in rule retention, while our measures are.
- We also provide a correlation matrix of our three measures and three pre-existing measures, showing that there is almost no correlation between the performance of a classifier built from the anonymized data, and the retention of the original rules. This demonstrates that no single measure can be expected to inform the user (e.g. the data owner) about every change in the data after anonymization, and that our methodology captures information that no pre-existing measure does.
- We analyze the effect of anonymization on four specific rules found in the Adult dataset.
- We use a real-world differentially-private anonymization technique [4] and measure the utility of the resulting dataset.

We also make the code for our three implementations of our proposed methodology available online.[1]

[1] The code can be found at *http://samfletcher.work/code* or *http://csusap.csu.edu.au/~zislam/*.

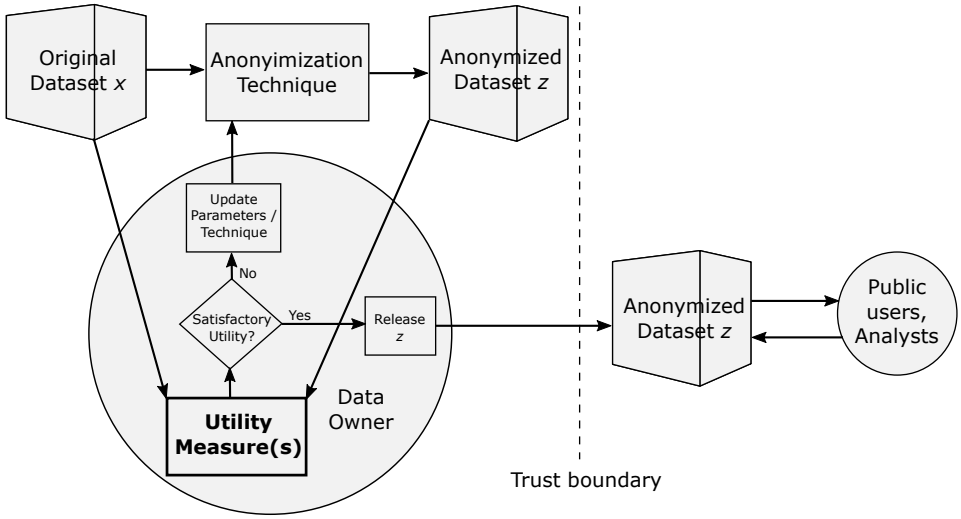

Figure 1: A high-level diagram of the scenario discussed in this paper, where a data owner is using an anonymization technique to output an anonymized version of their data to the public. In this paper, we focus on measuring the comparative utility of z compared to x, highlighted in bold.

In Section 3, we discuss related work, as well as three pre-existing utility measures that we will use in our experiments. In Section 4, we propose a generalized methodology for measuring the retention of rules in anonymized data. In Section 5, we present three implementations of the proposed methodology. In Section 6, we use a thought experiment to explore our measures alongside pre-existing measures. In Section 7, we detail our experiments, and in Section 8, we present our empirical findings. We summarize our thoughts and conclude the paper in Section 9.

2 The Setting

In this paper, we frame the problem from the perspective of the data owner, where the data owner is wishing to release an anonymized version of their data to the public that retains the rules that exist in the original version of the data. Figure 1 presents a high-level view of the scenario. The data owner applies an anonymization technique to their dataset x, outputting an anonymized version z. They can then assess the quality (that is, utility) of the anonymized data, and re-do the anonymization process with different parameters or a different technique if they are not satisfied with z's quality. When they are satisfied, the anonymized dataset is released to the public.

We define a dataset x (or z) as a two-dimensional table made up of independent rows, each defined by the values it possesses in each column. Each row represents a record $r \in x$, and each column represents an attribute $A \in \mathcal{A}$. Each A is made up of its own set of values, with each r possessing one value v per A, written as $r_A = v; \forall A \in \mathcal{A}$. When x is anonymized to preserve privacy, we denote the anonymized dataset as z. To simulate the process of predicting the labels of future records, some records are excluded from x, with a model being built using only the "training dataset" x. The model's performance can then

be tested with the "test dataset" t. Since the labels of these excluded records are already known, the user can tell if the rules discovered in x predicted the labels in t correctly [5].

In order to preserve privacy, a dataset can be anonymized in a variety of ways. If the data owner is fully releasing the data to the public after some modification, noise might be added to each participant's values in a way that maintains the overall distribution of values while hiding the original values of any single individual. Techniques that use this approach include *additive noise* [6, 7, 8] and *multiplicative noise* [9]. Alternatively, groups of values could be "generalized" to a single value, making values that were once different indistinguishable from each other. This is the approach *k-anonymity* [10] and its sibling techniques (such as *l-diversity* [11]) use. *Differential Privacy* [12, 13, 14] offers strong guarantees to each participant in that data, promising them that their participation in the dataset will be undetectable. It does so by adding Laplace noise to continuous values and by changing discrete values to other values with weighted probability. Differential Privacy can be used to generate a new dataset, where new records are created using information from the original dataset [4, 15]. We will explore the utility of a real-world example of differentially private data in Section 8.4. Alternatively, if the dataset is remaining under the control of the data owner and the public is only allowed to query the dataset interactively, differential privacy can be used to modify the results of the individual query outputs [16, 17, 18]. In this paper we focus on the former, non-interactive scenario, where the data owner releases an anonymized version of the dataset and then releases control of it, having no more say in how the dataset will be used.

Note that the specific method of anonymization is not the focus, rather the focus is on how to measure the utility of the data once the anonymization method has been used. It is also worth mentioning that for many of these anonymization techniques, such as k-anonymity and Differential Privacy, the level of privacy preservation achieved is determined by the parameters used during the anonymization process, not via measurement after the fact. For example, the level of privacy achieved by an anonymization technique using Differential Privacy is determined by the size of the privacy budget ϵ used, which mathematically bounds the probability of a participant being detected in z [13, 18]. Measurements are only required to evaluate the utility of the data; not the privacy.

Regardless of which anonymization technique is used, some degradation of dataset x's utility is unavoidable, due to z being not as truthful as x by definition. Balancing this loss of utility with privacy requirements is known as the privacy-utility trade-off, and optimizing this trade-off is key to a successful privacy-preservation technique. In order to assess the utility of a dataset anonymized to preserve privacy, it is currently common practice [8, 19, 20] to use a variety of data mining techniques to discover rules in the anonymized dataset z, and then see if those rules can correctly predict the labels of future records. Data mining techniques such as decision forests [21, 22], association rule mining [23] and frequent pattern mining [24] are some of the methods that can be used to extract rules, where the rules are in the form $\psi \rightarrow c$.

The antecedent ψ is a collection of conditions or requirements that when true for a record r, predict that a consequent $c \in C$ (in other words, a label or class value) is also true for that record. The conditions in ψ specify certain values $v \in A$ of attributes $A \in \mathcal{A}$ that some records will meet and others will not. These conditions can be of the form $r_A = v$ or $r_A \in V$ for discrete attributes, where V is some subset of A; or in the case of continuous attributes, be of the form $r_A > v$, $r_A < v$, or contain some other operator. Negation versions of these operators can also be used, such as $\neq$ and $\notin$. See Table 1 for some examples of what antecedents might look like.

i	**Antecedent,** ψ_i	**Consequent,** $Pr(C = c)$
0	$206134 > Census\ Weighting \leq 346177$ AND $Capital\ Gains \leq 4737$ AND $Age > 40.5$ AND $Capital\ Loss \leq 1836.5$ AND $Hours\ per\ Week > 52.5$	$Pr(Income \leq \$50,000) = 0.57$, $Pr(Income > \$50,000) = 0.43$
1	$244440.5 < Census\ Weighting \leq 249542$ AND $Capital\ Gains \leq 4737$ AND $Age \leq 40.5$	$Pr(Income \leq \$50,000) = 0.92$, $Pr(Income > \$50,000) = 0.08)$
2	$Census\ Weighting \leq 206134$ AND $Capital\ Gains > 5316.5$	$Pr(Income \leq \$50,000) = 0.05$, $Pr(Income > \$50,000) = 0.95$
3	$116388 > Census\ Weighting \leq 200855.5$ AND $Capital\ Gains \leq 5316.5$ AND $Capital\ Loss \leq 1198$ AND $Hours\ per\ Week \leq 53.5$	$Pr(Income \leq \$50,000) = 0.81$, $Pr(Income > \$50,000) = 0.19$

Table 1: A selection of rules discovered in the "Adult" dataset [35].

If a record r meets every condition in ψ (in other words, $r \in \sigma_\psi(x)$)[2], it is predicted to have a label $C = c$ either with certainty or with some probability (in other words, $0 \leq P(r_C = c) \leq 1$). Note that we abuse notation and simplify $\psi \rightarrow c$ to just ψ when it is clear from context that we mean the whole rule.

Which rules the data owner deems important enough to assess for changes in utility is outside the scope of this paper. What makes a rule valuable can vary depending on the expected workload; that is, the needs of the user. Measures have been developed to assess different aspects of rules, such as a rule's support or coverage [25, 26], confidence [27], conciseness [28], peculiarity [29], or many other aspects depending on the user's needs [30, 31, 32, 33]. These measures are often collectively referred to as interestingness measures. How the rules are discovered is also outside of the scope of this paper – any rules in the form $\psi \rightarrow c$ are applicable to the solution proposed in this paper. In our experiments, we arbitrarily use the CART decision tree algorithm [34] to generate a collection of rules. The number of rules that are assessed for changes in utility can be as high as the data owner likes.

We refer to the set of all rules $\{\psi_i; \forall i\}$ discovered in a dataset x as Ψ_x. If Ψ_x is used to predict C for all records in t, we write the average accuracy α of these predictions (that is, the "Prediction Accuracy") as $\alpha(\Psi_x|t)$. In words, this can be read as "The accuracy of Ψ_x at correctly predicting class labels of records in t". Some examples of rules can be seen in Table 1, including the probability of r having a label c. Our proposed methodology and its implementations are independent of methods for discovering rules – any rules in the form $\psi \rightarrow c$ are applicable, regardless of whether they were manually found, found with a decision tree or via association rule mining or frequent pattern mining, or any other method.

[2]Read σ as the mathematical symbol for *selection*. For example, $\sigma_p(q)$ is the subset of elements in q for which p is true. p can either be a statement such as $C = c$, or a set of statements such as ψ, in which case all statements in set p must be true for an element in q in order for that element to be in the set $\sigma_p(q)$.

3 Related Work

One of the most popular methods for assessing the utility of z is to build a model from it, and compare its performance to the performance of a model built from x [6, 8, 19, 20, 36, 37, 38]. Prediction Accuracy, for example, can tell the data owner how accurate the two models are at predicting the class labels of previously-unseen records $r \in t$ [5]. For models that can be written in terms of rules, such as decision tree classifiers, we can denote the two models as Ψ_z and Ψ_x. Prediction Accuracy is sometimes reversed to instead represent Prediction Error[3] [38], also known as Error Rate [39]. Other common measures are F-measure [40, 41, 42] and AUC [43, 44, 45], where again Ψ_z is compared to Ψ_x using t (refer to Figure 2 for a graphical representation of these datasets and sets of rules). These measures are not limited to measuring the utility of rules, and can be used in any classification model, such as k-Nearest Neighbor models [39]. We use Prediction Accuracy, F-measure and AUC in our experiments.

Formally, Prediction Accuracy can be written as:

$$\alpha(\Psi_x|t) = \frac{1}{|t|}\sum_{r\in t}\mathbf{1}(\Psi_x(r) = r_C) \ ,$$

where $\Psi_x(r)$ is the outputted label when record r is inputted into classifier Ψ_x (that is, the set of rules), and $\mathbf{1}(\bullet)$ is the indicator function, returning 1 if $\bullet$ is true and 0 otherwise. Thus, Prediction Accuracy is the fraction of records that have their class label correctly predicted.

AUC and F-measure are most appropriate when C is binary (in other words, $|C| = 2$), and where $c_1 \in C$ is the "important" class label, or the class label that the user is trying to correctly predict, and $c_2 \in C$ is unimportant. These labels are often referred to as the "positive" and "negative" labels, respectively. A "True Positive" (TP) label is therefore a label that was correctly predicted to be positive, a "False Positive" (FP) is a label that was predicted to be positive but was not, and similarly for "True Negative" (TN) and "False Negative" (FN).[4]

F-measure [46] can be formally written in terms of precision ($\frac{TP}{TP+FP}$) and recall ($\frac{TP}{TP+FN}$) as:

$$F_\beta = (1+\beta^2) \times \frac{precision \ \times \ recall}{\beta^2 \ \times \ precision + recall} \ ,$$

where β is very often equal to 1, so that recall and precision have equal weighting. We use $\beta = 1$ in our experiments.

Meanwhile, AUC [47] is shorthand for "Area under the ROC curve", with "ROC" in turn being short for "Receiver Operating Characteristic". The ROC curve describes the trade-off between TP (benefits) and FP (costs). Often it is plotted on axes with the TP Rate ($\frac{TP}{TP+FN}$) as the y-axis and the FP Rate ($\frac{FP}{FP+TN}$) as the x-axis. Thus, AUC is the area under this curve. It represents the probability that Ψ_x is more likely to predict a positive label as positive than to predict a negative label as positive. It has become popular in the machine learning community as of late, despite some problems it has when comparing different classifiers [48, 49].

Comparing the performance of Ψ_z to the performance of Ψ_x has a shortcoming, however: it cannot tell the user if the rules discovered in z are the same rules discovered in x. There is no way of knowing if the anonymization process applied to x caused the original rules

[3] In other words, $1 - prediction\ accuracy = prediction\ error$.

[4] Using this notation, we can actually re-write Prediction Accuracy as $\frac{TP+TN}{TP+TN+FP+FN}$.

to change (or disappear), or if weaker rules became strong enough to become more prominent. Similar problems have been identified in the past, specifically that the performance of one type of classifier does not mean that other classifiers will perform similarly [19]. In fact, if the utility of an anonymized dataset is judged based on the performance of one or more classifiers, it is recommended that the dataset is not released at all, and instead just release the classifiers [19]. Our proposed methodology solves this problem by measuring the utility of the anonymized data directly. Other problems arise from relying too heavily on Prediction Accuracy, such as when the data labels are imbalanced [50], or when the comprehensibility of the classifier needs to be considered as well [51].

Less work has been done on utility measures that specifically focus on rule retention, but there has been some. A measure known as "average relative error for aggregate counts", or RE, has been used in the past [52, 53], where datasets x and z are both queried with the same count query f, and the outputs are compared:

$$RE = \frac{|f(x) - f(z)|}{f(x)} \ .$$

A query is very similar to a rule except for the consequent; it uses a set of conditions to filter a dataset down to the subset of records that obey all the conditions. As we will demonstrate in Section 5.2, RE can be thought of as an implementation of our more general methodology for measuring rule retention.

A similar but different area of research related to our work is utility-based anonymization techniques, in which a utility measure is used as a cost function during the anonymization process [54]. Our proposed methodology differs from this approach in that it is not for designing cost functions, but instead for designing standalone, workload-specific utility measures. Some measures can be used as both cost functions and utility measures. Kullback-Leibler (KL) divergence [57], for example, has been proposed as a standalone utility measure for measuring the difference between two probability distributions; one of the original dataset x, and one generated from anonymized marginals (also known as frequency tables) of x [58]. KL-divergence has also been proposed as a cost function in a bottom-up, greedy anonymization algorithm for achieving k-anonymity [59]. Other measures exist for measuring the difference between two distributions, such as Chi-squared histogram distance [60], which we will be using in Section 5.3. The most appropriate measure to use when comparing distributions depends on the properties of the distributions, such as if they are modeling continuous numerical data, discrete numerical data or discrete categorical data; and if discrete, whether or not both distributions have the same number of buckets [60].

4 A Methodology for Measuring Rule Retention

While Prediction Accuracy is an excellent measure when evaluating the utility of a classifier or model [61, 62], care needs to be taken when extending its use to the privacy-preservation domain. It has been common in the past for the impact of privacy-preservation techniques on the utility of the data to be measured with Prediction Accuracy [6, 8, 19, 20, 36, 37, 38]. This necessitates applying a data mining technique to the anonymized data z to build a classifier (or discovering a set of rules with another technique) and then testing how well the discovered rules[5] Ψ_z can accurately predict the class label of records in a test

[5]Note that a classifier is semantically the same as a set of rules if it can be broken down into antecedents and consequents.

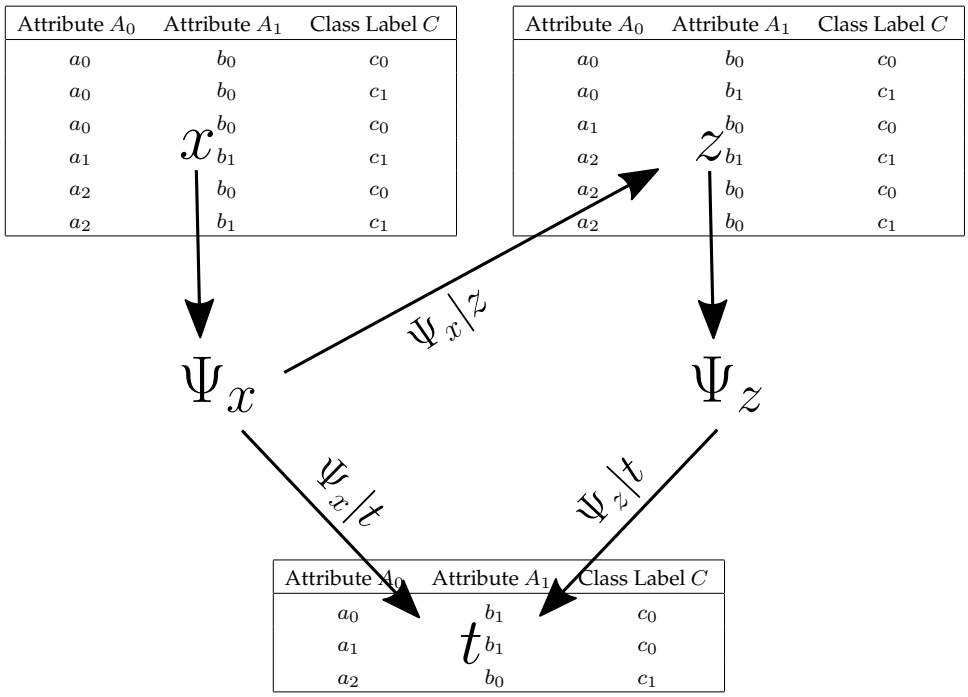

Figure 2: A diagram of the datasets and rule sets involved when comparing the utility of anonymized data z to the original data x. Ψ_x is a set of rules (in the form $\psi \rightarrow c$) discovered in x, and similarly for Ψ_z and z. $\Psi_x|z$ is our generalized notation for any assessment of Ψ_x when z is inputted into it (and similarly for other rule sets and inputs). An example of what Ψ_x and Ψ_z could look like is presented in Table 2 and Table 3, respectively.

dataset t. The accuracy α of Ψ_z can then be compared to the accuracy of Ψ_x (in other words, $\alpha(\Psi_x|t) - \alpha(\Psi_z|t)$), and the difference is considered to be how much the privacy-preservation technique has affected the data. See Figure 2 for a graphical representation of the datasets and sets of rules used in this discussion. Examples of rules that could be discovered in the x and z datasets in Figure 2 are presented in Table 2 and Table 3, respectively.

We see two problems with this current methodology:

Problem 1 It only tells the user if the particular technique used to find the rules in z produced a good classifier / model / set of rules. Perhaps some amount of implicit assumptions can be made about the ability of other techniques to perform similarly well (in other words, "technique f did well, so techniques g and h probably produce similar results"), but there is nothing explicitly said by the quality of one Ψ_z about the quality of all data mining techniques applied to z.

Problem 2 The user cannot tell if the rules in Ψ_z are the same rules that can be discovered in x (such as the rules that would be discovered if the same data mining technique was applied to x, producing Ψ_x). Some rules might still be there, while others might not, and the rules that are still there might have changed in any number of ways (such as changes in the support or confidence of the rule, or the values v used in the conditions in ψ).

i	**Antecedent,** ψ_i	**Consequent,** $Pr(C = c)$	**Support**
0	a_0	$Pr(c_0) = 0.66,\ Pr(c_1) = 0.34$	3
1	a_1	$Pr(c_0) = 0.0,\ Pr(c_1) = 1.0$	1
2	a_2 AND b_0	$Pr(c_0) = 1.0,\ Pr(c_1) = 0.0$	1
3	a_2 AND b_1	$Pr(c_0) = 1.0,\ Pr(c_1) = 0.0$	1

Table 2: An example of what Ψ_x could look like: some rules discovered from x in Figure 2. We include the confidence and support of the rules as well. These rules could be manually discovered, or discovered by a data mining algorithm, such as a decision tree.

i	**Antecedent,** ψ_i	**Consequent,** $Pr(C = c)$	**Support**
0	a_0 AND b_0	$Pr(c_0) = 1.0,\ Pr(c_1) = 0.0$	1
1	a_0 AND b_1	$Pr(c_0) = 1.0,\ Pr(c_1) = 0.0$	1
2	a_1	$Pr(c_0) = 1.0,\ Pr(c_1) = 0.0$	1
3	a_2	$Pr(c_0) = 0.34,\ Pr(c_1) = 0.66$	3

Table 3: An example of what Ψ_z could look like: some rules discovered from z in Figure 2.

A solution to Problem 1 is for the data owner to build a Ψ_z and Ψ_x with every possible data mining technique they think is worth checking [19]. This solution is extremely computationally expensive, and does not address Problem 2. If this solution is not used, then a user must either trust the implicit assumption that other data mining techniques will perform similarly, or release the collection of rules Ψ_z that they *did* test, and not release z to the public at all [19]. To the best of our knowledge, aside from our preliminary investigation [63], no solution to Problem 2 currently exists in the literature.

We therefore propose a methodology that aids in addressing Problem 1, and fully addresses Problem 2. Rather than discovering a collection of rules in z (in other words, Ψ_z) that may or may not have any relation to the rules in x (in other words, Ψ_x), we propose that the data owner defines a collection of rules found in x and evaluates if the data in z still follows those rules. A full solution to Problem 1 may be intractable, due to the difficulty of defining an exhaustive collection of rules that any conceivable user might be interested in. Having a "reasonably good" collection of rules is possible though, and it is this partial solution that we achieve in this paper. More specifically, we propose designing measures of rule retention using the following methodology:

$$\Psi_x | z \ .$$

No model Ψ_z needs to be computed in order to measure the rule retention of z. Nor is t required, unless it is desired for other, unrelated testing.

This methodology makes progress with respect to Problem 1 by not restricting the definition of "utility" to the context of a particular data mining technique being applied to z,

and instead raises the level of abstraction when defining the "utility" of z to a technique-independent level. The methodology fully solves Problem 2 since the same rules that were found in x are being used to evaluate z.

Of course, the methodology used by Prediction Accuracy and other measures of *model performance* can be written similarly: $\Psi_z|t$. When using this methodology, an important part of the process is to acknowledge that a model trained on z cannot be tested on z, lest risk over-estimating the model's performance when used in the real world. This is due to the phenomenon of over-fitting; when a machine learning algorithm is taught to distinguish between labels, the algorithm will tend to *reinforce* any rules it finds that correctly classify more labels, even if the "rules" are merely idiosyncrasies of the training data, such as noise, and do not model real life. We therefore use separate testing data t to judge if the rules the algorithm discovered are, in fact, valid. Due to how similar $\Psi_x|z$ and $\Psi_z|t$ first appear based on their notation, the reader may wonder if the same risk of over-fitting needs to be ascribed to $\Psi_x|z$. Our methodology represents a fundamentally different way of measuring properties of z however – in terms of *rule retention* – and the concept of over-fitting does not exist in this methodology. Rather than measuring how much a set of rules discovered in a sample is generalizable to the universe, we are measuring how much a perturbed version of the sample still follows the original rules. The utility of the original rules is not what is being measured in $\Psi_x|z$; that is the role of measures of *model performance*, and can be measured in the course of defining Ψ_x, before moving on to measuring rule retention in z.

The next natural question is: how exactly do we evaluate z with Ψ_x? There are many potential implementations of our proposed methodology, but we provide three examples below in Section 5. The first measure (in Section 5.1) evaluates how much of Ψ_x as a whole has been retained in z. The next two measures (in Section 5.2 and Section 5.3) evaluate the presence of each rule in Ψ_x (in other words, $\psi_i; \forall i$) separately, offering the data owner the ability to check for changes in individual rules, as well as seeing the average change. Other implementations can easily be designed to meet the needs of the data owner. Every dataset has its own nuances, and it is usually advantageous to take those nuances into account when measuring the effect of privacy-preservation techniques, rather than trying to use a "catch-all" approach. The release of data to the public will be a one-time event (once it's out there, there's no taking it back!), and so spending additional resources to properly evaluate x and z is likely worth it.

5 Implementations of our Methodology

5.1 Rule Accuracy

Introduced by us in a 2014 conference [63] and not published in a journal until now, Rule Accuracy is a simple measure that compares x and z. Like its name might suggest, it is very similar to Prediction Accuracy in that it measures the average accuracy of a collection of rules at predicting the class value C of some data. However, instead of predicting the class value of some testing data t, it predicts the class labels of the anonymized data z. If we write the Prediction Accuracy of z as $\alpha(\Psi_z|t)$, then we can write the Rule Accuracy of z using similar notation, as $\alpha(\Psi_x|z)$. In a privacy-preservation scenario the point is to compare z's performance to x, so Prediction Accuracy becomes $\alpha(\Psi_x|t) - \alpha(\Psi_z|t)$, and the Rule Accuracy equivalent is therefore $\alpha(\Psi_x|x) - \alpha(\Psi_x|z)$.[6] Note that while $\alpha(\Psi_x|x)$ should

[6] Remember that z is an anonymized version of x, with each record in z corresponding to an unaltered version in x.

not be used to assess the quality of a classifier due to the risk of over-fitting, in this paper we are not concerned with the generality of the rules. How Ψ_x is created or defined is outside the scope of this paper. Instead, the difference between $\alpha(\Psi_x|x)$ and $\alpha(\Psi_x|z)$ tells us the difference in the number of records that are contributing to the prediction made by each rule (where the prediction is the majority class label). Rule Accuracy is a way of measuring the presence of x's rules in z; if $\alpha(\Psi_x|x) - \alpha(\Psi_x|z)$ is close to zero, then the user knows that a similar number of records in x and z are contributing to the correct predictions made by the classifier built from x. Since the records in z are just anonymized versions of the records in x, this is a valuable thing to know! If $\alpha(\Psi_x|x) - \alpha(\Psi_x|z)$ is closer to one, the user knows that the records were anonymized in a way that reduced the presence of the rules found in Ψ_x. If $\alpha(\Psi_x|x) - \alpha(\Psi_x|z)$ is negative, this is actually just as bad as a positive result of similar magnitude, because x is trusted data – any random modifications made to x is further from the trusted data by definition, even if some quality metrics increase. Ideally we want every rule in Ψ_x to be just as prevalent in z as it is in x; no more, no less. Thus we define Rule Accuracy as:

$$\text{Rule Accuracy} = |\alpha(\Psi_x|x) - \alpha(\Psi_x|z)| \ . \tag{1}$$

Rule Accuracy evaluates whether Ψ_x, as a whole, can correctly predict C for records in z. Since it uses an identical process to Prediction Accuracy (with the user simply having to redirect the measure to check z rather than t), it gains all of the benefits of Prediction Accuracy such as low computation time and conceptual simplicity. What it does not do, however, is evaluate the presence of each rule individually (in other words, $\psi_i; \forall i$). There are almost always multiple rules that predict the same $c \in C$, so it is possible that some rules no longer have records in z that follow them (and instead those records follow different rules) without the Rule Accuracy result changing. As long as the record's new rule still correctly predicts c, the Rule Accuracy measure is insensitive to this change. The following two measures avoid this insensitivity by evaluating the rule retention in z on a per-rule basis, rather than evaluating the entire rule list as a whole.

5.2 Rule Support Distance (RSD)

The "support" of a rule is the number of records in a dataset that a rule covers [26, 27], and can be represented as $|\sigma_\psi(x)|$ when describing the support of rule ψ in dataset x. Whether C is predicted correctly is irrelevant when measuring support. To measure the support for ψ in x compared to z, we can calculate $|\sigma_\psi(x)|$ and $|\sigma_\psi(z)|$. By comparing these results, a user knows how much the presence of an individual rule ψ has changed due to the modifications made to x (resulting in z). This level of granularity allows the user to use their domain knowledge to make specific assessments of the status of each ψ. This can naturally be repeated for all $\psi \in \Psi_x$. To summarize the overall support retention of Ψ_x for a dataset z, the mean difference can be calculated:

$$RSD = \frac{1}{|\Psi_x| \times |x|} \sum_{\psi \in \Psi_x} | \, |\sigma_\psi(x)| - |\sigma_\psi(z)| \, | \ . \tag{2}$$

Note that each rule contributes equally to the mean difference. Rules with higher support are not assumed to be more important, since each ψ in Ψ_x should have already been assessed by the user as being important enough to worry about preserving in z, and we are now only interested in if the original support has *changed*. It should be noted though that

rules in Ψ_x with very low support cannot reduce in size by as much as rules with high support – support cannot go below 0 – so the presence of many rules with low support risks "diluting" RSD.[7] However it is normal for such small rules to be considered as idiosyncrasies of x, and not generalizing to future records (such as t), and so most data mining algorithms automatically remove them from Ψ_x [5]. This is sometimes referred to as the "minimum support threshold".

Rule Support Distance (RSD) has a defined lower and upper limit of $0 \leq RSD \leq 1$, allowing for an intuitive interpretation of the result, such as: "The average percentage change in the prevalence of a rule when anonymizing x to create z".

RSD is similar to measuring the average relative error for aggregate count queries (RE) [52, 53], since counting the number of records that match a query is the same thing as measuring the support of a rule. RE can be categorized as another implementation of our proposed generalized methodology for measuring rule retention. The main difference between RSD and RE is the problem domain; RE can be used when dealing with count queries, and RSD can be used when dealing with $\psi \rightarrow c$ rules.

The aim of privacy preservation is to (1) make any individual record difficult to identify, while (2) leaving the rules as unaffected as possible [6, 8, 64]. If the user considers support to be an important component of rules, then RSD can be used to monitor this component. The specific records that matched each ψ in x is irrelevant – ψ will still be just as prevalent in z as it was in x if other records take the place of the records that no longer follow ψ. In order for a record to change which rule it matches, its values must have changed during the anonymization process enough for it to legitimately meet the conditions of a different rule.

While Rule Accuracy indirectly measures the support of $\psi_i \in \Psi_x; \forall i$ in z, RSD does so directly, removing any uncertainty about the presence (in other words, coverage or support) of each rule in z.

5.3 Rule Label Distance (RLD)

Say a record $r \in x$ meets the conditions of a certain rule $\psi_i \in \Psi_x$ (in other words, $r \in \sigma_{\psi_i}(x)$). When x is anonymized to z, it is possible that r will be changed in a way that causes it to meet the conditions of a different rule $\psi_j \in \Psi_x$ (in other words, $r \in \sigma_{\psi_j}(z)$). If this occurs, the distribution of labels (in other words, C) will change for both ψ_i and ψ_j, since r_C has been removed from ψ_i's distribution of class labels and added to ψ_j's. The purpose of a rule is often to predict C, and so it is important to know how much that prediction might have changed in z. Rule Accuracy measures this to an extent, as ψ_i and ψ_j might predict different class labels and a maximum of one of those predictions can be correct for a record r. But it is also possible that the two rules will predict the same class label, leading to no change in the Rule Accuracy of z compared to x (at least as far as r is concerned). The consequent of any rule ψ is usually the most common class label to occur out of all the records in $\sigma_\psi(x)$, with any other class labels being ignored [5]. This has the effect of making ψ's prediction of $C = c$ appear identically confident[8] regardless of how high or low the frequency of c is in $\sigma_\psi(z)$ compared to $\sigma_\psi(x)$, as long as it remains the most frequent class label.

[7] This effect is caused whenever many small differences are averaged alongside several large differences. The presence of near-zero numbers effectively reduces the average, diluting the larger differences. There is nothing inherently wrong with this, but it is usually undesirable.

[8] "Confidence" refers to the certainty or reliability of a rule – that is, how frequent the most frequent label is [25]. If 100% of the records in a rule have the same class label, then that rule can be considered highly reliable.

To avoid these problems, we use the Chi-squared histogram distance [60] to measure differences in the distribution of C between $\sigma_\psi(x)$ and $\sigma_\psi(z)$:

$$\chi^2(\sigma_\psi(x), \sigma_\psi(z)) = \frac{1}{2} \sum_{c \in C} \frac{(f(\sigma_\psi(x), c) - f(\sigma_\psi(z), c))^2}{f(\sigma_\psi(x), c) + f(\sigma_\psi(z), c)} \ , \tag{3}$$

where $f(\sigma_\psi(x), c)$ is the relative frequency of the class label c in $\sigma_\psi(x)$ (that is, the fraction of records in $\sigma_\psi(x)$ that have $r_C = c$), and similarly for $f(\sigma_\psi(z), c)$ in respect to $\sigma_\psi(z)$:

$$f(\sigma_\psi(x), c) = \frac{\sum_{r \in \sigma_\psi(x)} \mathbf{1}(r_C = c)}{|\sigma_\psi(x)|} \ .$$

Just like with Chi-squared hypothesis testing, Chi-squared histogram distance becomes unstable if there are less than five samples. This limitation is automatically handled if a minimum support threshold for each rule ψ was applied when making Ψ_x; otherwise we recommend discounting any rules that have less than five class labels (in other words, ignoring rules $\psi \in \Psi_x$ where $|\sigma_\psi(x)| < 5$).

Other measures of distribution distance could be used, such as KL-divergence [58, 59], however none is as appropriate as Chi-squared histogram distance. For example, KL-divergence works best with continuous numerical data, while Chi-squared histogram distance is specifically for two distributions consisting of a small, equal number of discrete categorical buckets [60], which is the situation we have here where we are comparing label frequencies.

Even if the majority c value in $\sigma_\psi(x)$ occurs even more frequently in $\sigma_\psi(z)$ (and thus has increased confidence), this should not be considered as an improvement unless the anonymization process that created z was aiming to improve rule utility. In scenarios such as privacy preservation, the distribution of C for $\sigma_\psi(x)$ is considered to be the ground truth. RLD (Rule Label Distance) successfully captures this scenario, where any distance away from x is a reduction in utility by definition. The mean Chi-squared histogram distance of all rules in Ψ_x can then be easily calculated:

$$RLD = \frac{1}{|\Psi_x|} \sum_{\psi \in \Psi_x} \chi^2(\sigma_\psi(x), \sigma_\psi(z)) \ . \tag{4}$$

It should be noted that Chi-squared histogram distance is invariant to the number of records [60], and so the support of a rule (both in x and z) does not affect the result. If the support of each rule is deemed relevant by the user, $|\sigma_\psi(x)|$ can easily be taken into account as well. We do not recommend combining a rule's support difference and label distribution distance into a single result, as the results are likely to be far more informative when separate. This is true for both single rules and the mean results (RSD and RLD). Chi-squared histogram distance is also invariant to the number of labels, so it is not restricted to datasets or rules with a particularly sized C. This is often a concern with popular measures such as AUC [47] and F-measure [46], where non-binary class attributes need to be treated with care [65].

6 A Thought Experiment

We use a thought experiment to demonstrate the sensitivity of our measures to changes in the data that are not detected by pre-existing measures. We will use the toy datasets seen

	Rule Accuracy	RSD	RLD	Prediction Accuracy	AUC	F-measure
x	0.00	0.00	0.00	0.67	0.67	0.80
z	0.50	0.08	0.34	0.67	0.50	0.67
Change	**0.50**	**0.08**	**0.34**	**0.00**	**0.17**	**0.13**

Table 4: The results of six measures when the two rules seen in Table 2 undergo changes so that they now resemble what is seen in Table 3.

in Figure 2. The rules in Ψ_x have been written out in Table 2, along with their support and confidence. After anonymizing x with a privacy-preservation technique, the result is z as seen in Figure 2. The set of rules Ψ_z was then discovered from that anonymized data; we present the rules in Table 3. We then assess the quality of z using six measures: our three implementations of our proposed methodology, as well as Prediction Accuracy, AUC and F-measure. The results are tabulated in Table 4.

Several things have happened here. Firstly, Prediction Accuracy was completely incapable of detecting any changes in z compared to x. It is possible that an analyst would not care that z is different, and is only interested in being able to make good predictions on future data (and that is fine). However, if the analyst makes any assumption about the similarity between z and x with Prediction Accuracy, they have made a very dangerous mistake. As we can see in Table 2 and Table 3, Ψ_z is radically different from Ψ_x. Due to the changes present in z, the rules discovered in z are very different from the rules discovered in x. Our proposed methodology solves the issue of quantifying the intuition one has about the differences between Ψ_z and Ψ_x. Rule Accuracy, RSD and RLD were all able to accurately identify the differences between x and z that they are designed to identify: the overall retention of Ψ_x's rules, the changes in the rules' support and the changes in the rules' class label distribution, respectively.

AUC and F-measure were able to detect some changes, but it is important to recognize that these changes do not represent any connection between Ψ_x and Ψ_z. Both measures started by calculating the true and false predictions of the positive and negative labels of Ψ_x using t, and then they made similar calculations of Ψ_z using t. At no point was Ψ_z actually compared to Ψ_x, except indirectly, in much the same way that Prediction Accuracy indirectly compares them. As demonstrated by Prediction Accuracy's results in Table 4 though, there is no guarantee that any of these indirect comparisons will detect any differences at all. Even if they do, if a data owner is trying to decide whether to release z to the public (as seen in Figure 1), how do they use those results? Rule Accuracy, RSD and RLD offer concrete results about z's retention of x's rules.

7 Experiment Methodology

To empirically evaluate our three measures, we carry out the below experiments and present the results in Section 8.

1. We analyze individual rules in Section 8.1.

2. We detect the degradation of rule retention as higher anonymization requirements are used in Section 8.2.

3. We measure correlations between six different measures in Section 8.3.

4. We test the usefulness of RSD and RLD in a real-world scenario in Section 8.4.

For all experiments except for the analysis of individual rules, 10-fold cross validation is used, with each real-world dataset being split into a training dataset x and a testing dataset t. We use 17 datasets publicly available in the UCI Machine Learning Repository [35], and list their details in Table 5. When an anonymized dataset z is required in the experiments, we use one of three anonymization techniques (described in Section 7.1 and Section 7.2), applying it to x. The anonymization process is repeated 10 times, creating 10 separate z's, and the results of each measure are aggregated.

For Ψ_x and Ψ_z, we generate the rules with decision trees. Note that the rules could just have easily been manually created, generated from a different classifier, filtered using any number of interestingness measures, hand-picked from a list of generated rules, or by any other means that outputs rules in the form $\psi \rightarrow c$.

To generate a collection of rules Ψ_x (and Ψ_z) for each dataset, we run the CART algorithm [34], with a minimum leaf size (in other words, minimum support threshold) of $|x| \times 0.02$ and a maximum tree depth of 12. By generating rules in this way, we produce a set of realistic rules for each dataset, with the rules also varying in length (that is, the number of conditions in ψ). Another advantage of generating our rules in this way is that the deterministic nature of CART allows the reader to replicate our Ψ_x's exactly. To analyze individual rules in Section 8.1, four rules were manually selected from the first CART decision tree built from the Adult dataset, and presented in Table 1.

Our datasets range in size from 653 to 58000 records, 6 to 62 attributes, and 2 to 18 class labels, and include both numerical and categorical attributes. The number of rules (in other words, $|\Psi_x|$) ranges from 11 to 37. The details of the datasets are summarized in Table 5. For experiments involving AUC and F-measure (e.g. Table 6, discussed later) we limit our experiments to datasets with binary labels, where these measures are known to work best.

7.1 Toy Privacy Preservation Techniques

To simulate various anonymization techniques applied to a dataset, we add noise to the data in two simple ways. Each type of noise represents a different scenario respectively: where attribute and multi-attribute (in other words, multivariate) distributions are flattened (in other words, made more uniform); and where attribute distributions and most multi-attribute distributions are preserved. We simulate these scenarios using additive noise. Using these two scenarios, we explore what a user can learn from our three implementations of our proposed methodology, and how they compare to Prediction Accuracy, AUC and F-measure.

The two types noise addition we use are listed below. Note that these are simple toy noise addition techniques, and are not part of this paper's contribution. Neither of these noise types add noise to the label C. For each type of noise, we increase the percentage chance of changing a value in 2% increments, from 0% to 30%.

Uniform Noise (UN) A user-defined percentage of values in the dataset x are changed, with the result being z. If a value r_A is changed and A is a continuous attribute, the new

Name	Records	Continuous Attributes	Discrete Attributes	Labels	Majority Label %
Banknotes	1372	4	0	2	55%
Vehicle	846	18	0	4	26%
RedWine	1599	11	0	6	43%
Spambase	4601	57	0	2	60%
Wilt	4839	5	0	2	95%
WallSensor	5456	4	0	4	40%
PageBlocks	5473	10	0	5	90%
OptDigits	5620	62	0	10	10%
PenWritten	10992	16	0	10	10%
GammaTele	19014	10	0	2	65%
Shuttle	58000	9	0	7	79%
Credit	653	6	9	2	55%
Parkinsons	1040	28	1	2	50%
Yeast	1484	7	1	10	31%
Cardio	2126	21	1	3	78%
Adult	30162	6	5	2	75%
Bank	45211	7	9	2	88%

Table 5: Details of the datasets used in our experiments. The columns are, in order: the number of records in x; the number of continuous attributes in x; the number discrete attributes in x; the number of labels (class values) for C in x; and the relative frequency of the most common label in C.

value is selected from a uniform distribution between the minimum and maximum values of A. If A is a discrete attribute, r_A is changed to any unique value in the set A, with each value having an equal probability of being selected. Values are randomly selected, with the original value having no effect on the new value. This has the effect of flattening the distribution of values for all attributes, as well as flattening all multivariate distributions.

Gaussian Noise (GN) A user-defined percentage of values in the dataset x are changed, with the result being z. If a value r_A is changed and A is a continuous attribute, a random number is selected from a Gaussian distribution with a mean of zero and a variance equal to A's variance, and added to r_A. If A is a discrete attribute, r_A is changed to a value that is randomly selected from A's original set of values (including repeated values). GN therefore maintains the distribution of values for both numerical and categorical attributes. Additionally, continuous values are changed in a way that takes into account the original value. This means that each record's continuous values are likely to remain close to their original values, and thus the multivariate distribution of the dataset is likely to be preserved.

7.2 Real-World Differential Privacy

To simulate a real-world example, we anonymize the data of the Banknotes dataset using the differentially-private technique proposed by [4]. This involves partitioning the data into disjoint subsets, using noisy count queries to return the number of records in each subset, and then generating new records based on the attribute domains in each subset. The count queries are made noisy using Laplace noise (as is common in differential privacy [13]), where the amount of noise is dictated by the size of the privacy budget; the smaller the budget, the more noise that needs to be added to maintain differential privacy. Similar to how we can increase the amount of noise induced by UN and GN, we can increase the amount of noise added by DP (differential privacy) by decreasing the privacy budget. Common values for the privacy budget parameter range from 0.001 to 1.0. We encourage the reader to refer to [13] and [4] for more details on the mechanisms underlying differential privacy; they are outside the scope of this paper.

7.3 Pearson's Correlation Coefficient

For this experiment, we use the results of six different measures: Rule Accuracy, RSD, RLD, Prediction Accuracy, AUC, and F-measure. By comparing the results of each measure as noise increases, for each dataset, we calculate the measures' correlation to each other using Pearson's correlation coefficient (in other words, Pearson's **r** value) [66]. We calculate their correlation for each noise type separately. The coefficient has a range of $-1 \leq \mathbf{r} \leq 1$, where a result close to 1 indicates a high positive correlation (as one measure increases, so does the other measure), a result close to -1 indicates a high negative correlation (as one measure increases, the other decreases), and a result close to 0 indicates low correlation (the result of one measure has little bearing on the result of the other). To standardize the results of different datasets, we look at the *difference* between each measure's result on z compared to x. In other words, for each level of noise, we subtract the result that the measure achieved when there was zero noise. This has no effect on our proposed measures (which always equal 0 when there is no noise), and simply causes Prediction Accuracy, AUC and F-measure to be reported as the difference between the "true" result and the "noisy" result.

8 Our Implementations in Practice

In this section, we run the experiments outlined in Section 7, demonstrating how the implementations of our proposed methodology work in practice.

8.1 Analyzing Individual Rules

To demonstrate the information a user can learn about individual rules, Figure 3 presents the support and Chi-squared histogram distance of the example rules shown in Table 1, as UN increases. In this example, we can see that the four rules are affected quite differently by the noise addition. Some of the observations a data scientist could make about these four rules are:

- ψ_3 has gone from representing over 8000 of the of 30162 records in Adult to representing only 3000 records in the anonymized version of Adult by the time UN has reached 30%.

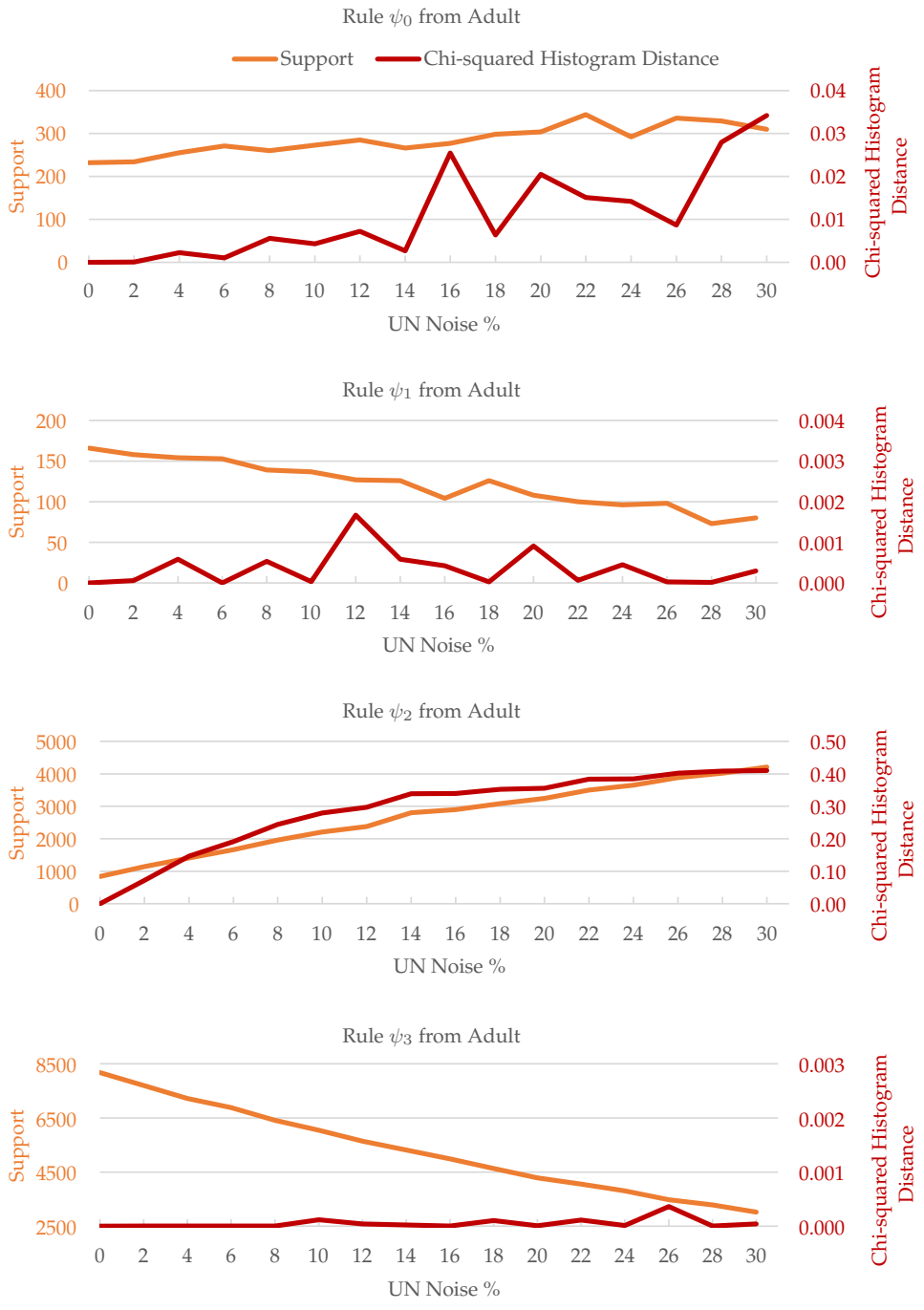

Figure 3: The Support and Chi-squared Histogram Distance of the example rules shown in Table 1 (discovered in the Adult dataset), as UN increases. The left y-axis measures the Support and the right y-axis measures the Chi-squared Histogram Distance.

- Despite this massive change in support, the distribution of the class labels in ψ_3 is actually almost exactly the same at all noise levels.
- The same cannot be said for ψ_2, where a massive change in support (from less than 1000 to roughly 4000) has been accompanied by a massive change in the distribution of class labels as well.
  - If this observation caused the user to investigate further, they would find that ψ_2's change in label distribution was enough to completely flip the prediction the rule is making! At 30% noise, the reported probability of a record having each label is $Pr(Income \leq \$50,000) = 0.68$, $Pr(Income > \$50,000) = 0.32$, compared to the probabilities shown in Table 1: $Pr(Income \leq \$50,000) = 0.05$, $Pr(Income > \$50,000) = 0.95$. It would be incredibly damaging to any analysis performed with z if the user trusted this rule.
- ψ_0 and ψ_1 represent a much smaller proportion of the Adult dataset, and have undergone moderate changes in support. ψ_0 has grown larger, while ψ_1 has become smaller, but neither saw enough change in label distribution to cause concern.
- These rules in Adult were discovered with a decision tree, along with 31 other rules that underwent a variety of changes in support and label distribution similar to the changes shown in Figure 3.

8.2 Detecting Rule Retention as Noise Increases

After averaging the support distance and Chi-squared histogram distance of all rules and thus calculating RSD and RLD, we can compare their assessments of z's rule retention for each dataset. We also compare RSD and RLD to the assessment made by Rule Accuracy. For each dataset, we present the results of RSD, RLD and Rule Accuracy as UN increases in Figure 4. Note that for Rule Accuracy, we present the percentage of cases where Ψ_x *incorrectly* predicts the label of records in z so that lower values signify better rule retention for all three measures.[9] As more noise is added, we observe that all three measures trend upwards as expected, but upon closer inspection we can see that they do not do so at identical rates.

8.3 Correlations Between Utility Measures

The differences in trends seen in Figure 4 are quantified by the correlations between the measures, seen in Table 6. The correlations are calculated using Pearson's correlation coefficient [66] as described in Section 7.3, and Prediction Accuracy, AUC and F-measure are included as well. Unlike Figure 4, the correlations are calculated using the nine datasets with binary labels for the benefit of AUC and F-measure.[10]

One observation we can make about Table 6 is that despite all the measures using the same data, they do not always agree with each other. Just because Prediction Accuracy decreases does not mean that F-measure also decreases, for example. Another observation is that Prediction Accuracy, F-measure and AUC have very weak correlations with any of

[9] That is, $rule\ accuracy\ error = 1 - rule\ accuracy$.

[10] The correlations among RLD, RSD, Rule Accuracy and Prediction Accuracy when using the datasets shown in Figure 4 are similar to those shown in Table 6.

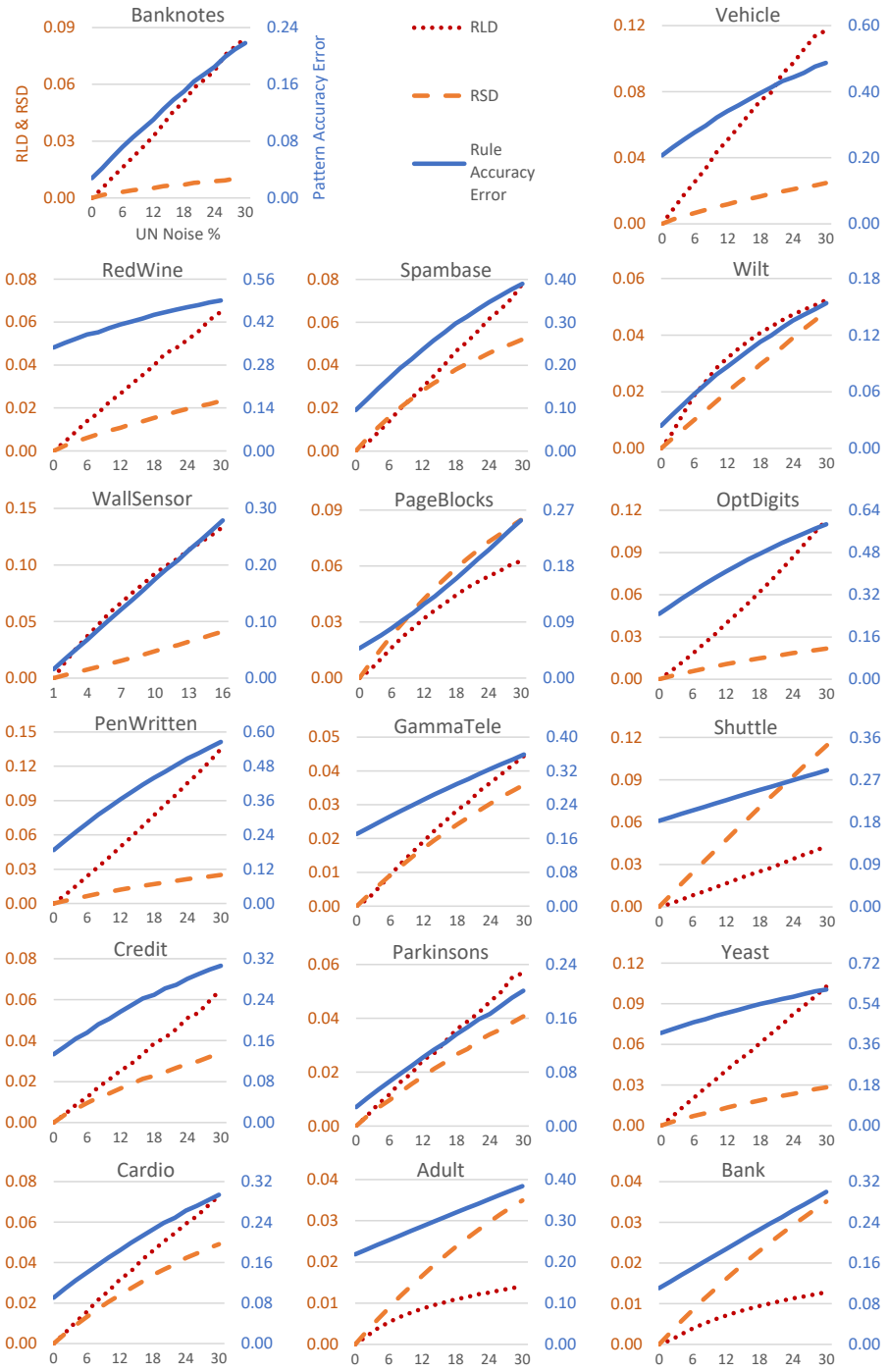

Figure 4: The mean results of RSD, RLD and Rule Accuracy Error as UN increases. The left-hand y-axis corresponds to RLD and RSD. The right-hand y-axis corresponds to Rule Accuracy Error. The x-axis is the percentage of noise from 0% to 30%.

Measure	RLD	RSD	Prediction Accuracy	AUC	F-measure
			UN		
Rule Accuracy	-0.77 (0.00)	-0.83 (0.00)	0.36 (0.00)	0.34 (0.00)	0.26 (0.00)
RLD		0.58 (0.00)	-0.32 (0.00)	-0.23 (0.01)	-0.09 (0.33)
RSD			-0.21 (0.02)	-0.35 (0.00)	-0.28 (0.00)
Prediction Accuracy				0.77 (0.00)	0.47 (0.00)
AUC					0.91 (0.00)
			GN		
Rule Accuracy	-0.86 (0.00)	-0.77 (0.00)	0.19 (0.04)	0.20 (0.02)	0.21 (0.02)
RLD		0.44 (0.00)	-0.04 (0.67)	-0.08 (0.35)	-0.09 (0.34)
RSD			-0.13 (0.14)	-0.27 (0.00)	-0.26 (0.00)
Prediction Accuracy				0.72 (0.00)	0.50 (0.00)
AUC					0.93 (0.00)

Table 6: A matrix of correlations for each combination of two measures, for all two noise types. We include the p value of each correlation in brackets (in other words, the probability of observing a result at least as extreme as the one reported by chance, assuming there is zero correlation).

our implementations of our proposed methodology. This is interesting, and confirms our suspicions that just because a good classifier (that is, a classifier that achieves good results) can be made from noisy data, does not mean that the rules in the noisy data have the same properties as the original rules, or even that the original rules are in the noisy data at all. For example if a user observed a particular amount of Prediction Accuracy loss after anonymizing x to z, there is no way to tell from this single number how much the support of the rules in x might have changed.

8.4 Real-World Differentially Private Data

In this experiment, we apply a real-world technique proposed in [4] that creates a differentially private version z of some dataset x. Here we use the Banknotes dataset as x. The details of the technique can be found in the original paper, suffice to say that new records are created based on the original records. The amount of noise incurred during this process depends on the "privacy budget"; the less budget there is to spend, the more noise that needs to be added to preserve privacy.

Figure 5 displays the results of our experiment. Four utility measures are presented: Rule Accuracy, Prediction Accuracy, RLD, and RSD. Several observations can be made when

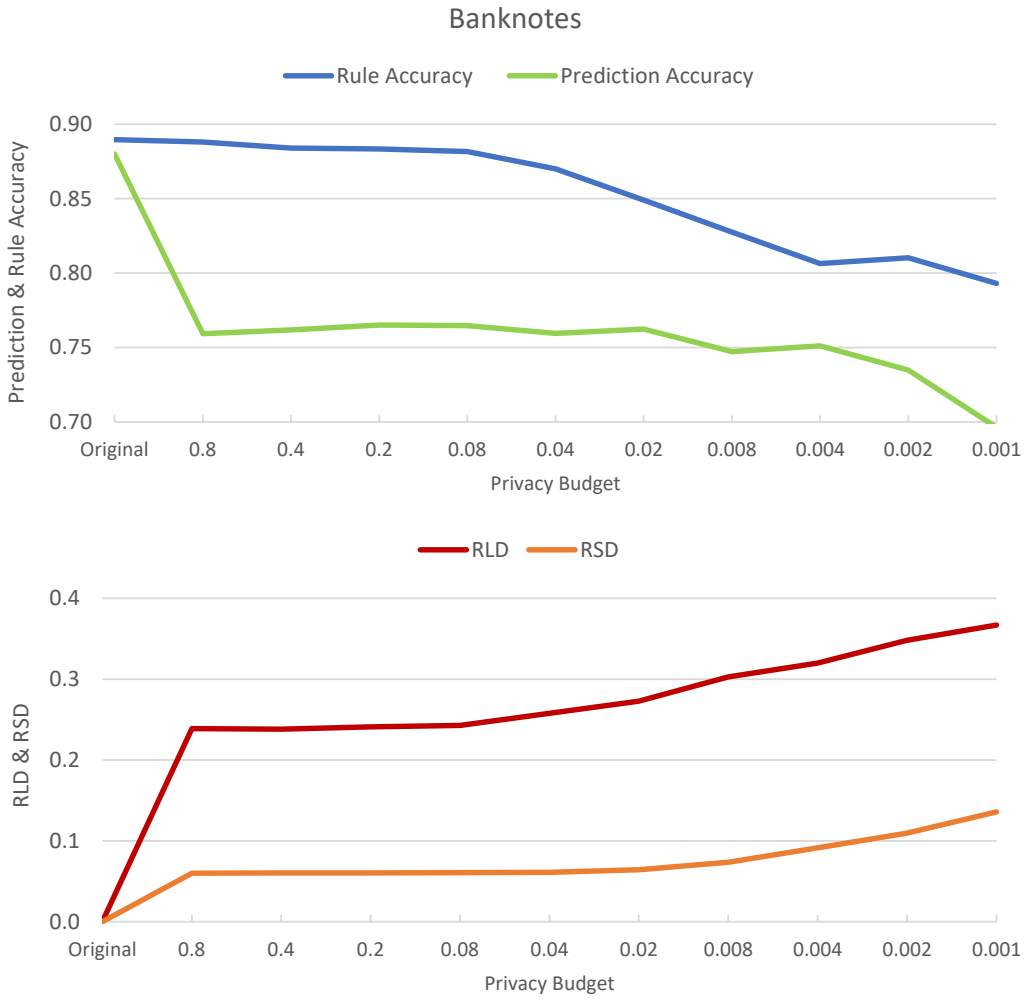

Figure 5: The mean results of Rule Accuracy, Prediction Accuracy, RLD and RSD as the privacy budget decreases. We use the differentially-private technique proposed by [4] to generate new data based on the Banknotes dataset.

looking at the results. Firstly, all four measures report that the quality of z worsens as the privacy budget decreases, as expected. However, we can see that aside from the initial drop between the original data (which has a functional privacy budget of infinity) and a privacy budget of 0.8, Prediction Accuracy does not detect any loss of quality until the budget drops to below 0.02. Rule Accuracy, on the other hand, detects a difference in quality when the budget drops to 0.08. Interestingly, Rule Accuracy reports that almost all of the records in z can be correctly predicted by the original rules until the budget is 0.08. Of course, neither of these results are "wrong" – they are measuring different things. The results reported by Prediction Accuracy tell us that with budgets between 0.8 and 0.02, a CART decision tree can be created without any significant difference in their ability to predict the label of future records. However even with a budget of 0.8, CART cannot produce a classifier with anywhere near as good as the one produced from x, even though Rule Accuracy tells us that z obeys the original rules. This observation may encourage the data owner running these experiments in the real world to manually view the CART tree to see where the rules are changing, and to test with other classifiers to see if the problem persists.

Similar observations can be made with RLD and RSD. The average Chi-squared distance between the labels in x and z that obey the rules found in x steadily rises as the privacy budget decreases. The average support of each of the original rules, on the other hand, stays the same until the budget decreases to 0.02. A budget of 0.02 is the same budget when the CART classifier started to deteriorate, which may be enough evidence for the data owner to investigate exactly which rules are changing in support, and whether or not they are the rules responsible for worsening the classifier.

Utility measures such as the four presented in Figure 5 (and hopefully many more) provide the data owner with the information they need to make informed decisions about the data. Based on the four measures provided, if the owner had to decide what data was of acceptable quality to release to the public, which trying to maximize privacy preservation, they may decide to use a budget of 0.02.

9 Discussion

None of our proposed measures can tell a user if a good classifier can be made from z. They are not trying to! If a user wishes to learn that, they can use machine learning algorithms on z and see if the resulting classifier has good performance, using measures such as Prediction Accuracy. Doing so, however, will not tell them if those machine learning algorithms found the same rules that existed in x. That is where our proposed methodology – and our implementations of that methodology – come in.

Rule Accuracy, RSD and RLD should not be interpreted as exhaustively measuring all aspects of rule retention. Rather, they are examples of quantifying specific effects a privacy-preservation technique can have on data. It is the responsibility of the data scientist performing the anonymization of x to assess what properties of a dataset are relevant or important, and then to measure how those properties might have changed after anonymization. Rule Accuracy measures the overall retention of the original rules; RSD measures changes in rule support, per rule; RLD measures changes in label distribution per rule; other measures might focus on quantifying changes in rule conciseness or peculiarity or any number of other properties that might make rules interesting to a user.

Prediction Accuracy is currently heavily relied upon in privacy-preservation research. While the measure itself is very useful, it should not be viewed as an all-encompassing measure of the quality of anonymized data, but rather as another example of quantifying a

specific property – the ability for accurate classifiers to be built using a variety of machine-learning algorithms.

Measuring properties of Ψ_x in z is straightforward, both conceptually and computationally, and can be easily used in conjunction with Prediction Accuracy and other measures. It enables the user to quantify aspects of M that previously could only be assessed with experience or intuition.

References

[1] A. Vellido, J. D. Martin-Guerroro, P. J. Lisboa, Making machine learning models interpretable, in: European Symposium on Artificial Neural Networks, i6doc, Bruges, Belgium, 2012, pp. 163–172.

[2] A. Criminisi, J. Shotton, E. Konukoglu, Decision forests for classification, regression, density estimation, manifold learning and semi-supervised learning, Tech. rep., Microsoft (2011).

[3] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, ACM Computing Surveys 41 (3) (2009) 1–15.

[4] Y. Xiao, L. Xiong, C. Yuan, Differentially private data release through multidimensional partitioning, Lecture Notes in Computer Science 6358 (1) (2010) 150–168.

[5] J. Han, M. Kamber, J. Pei, Data mining: concepts and techniques, Morgan Kaufmann Publishers, 2006.

[6] R. Agrawal, R. Srikant, Privacy-preserving Data Mining, in: ACM SIGMOD Conference on Management of Data, ACM, Dallas, Texas, 2000, pp. 439–450.

[7] D. Agrawal, C. Aggarwal, On the design and quantification of privacy preserving data mining algorithms, in: 20th ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems., ACM, 2001, pp. 247–255.

[8] M. Z. Islam, L. Brankovic, Privacy preserving data mining: A noise addition framework using a novel clustering technique, Knowledge-Based Systems 24 (8) (2011) 1214–1223.

[9] K. Liu, H. Kargupta, J. Ryan, Random projection-based multiplicative data perturbation for privacy preserving distributed data mining, IEEE Transactions on Knowledge and Data Engineering 18 (1) (2006) 92–106.

[10] L. Sweeney, k-anonymity: A model for protecting privacy, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10 (5) (2002) 557–570.

[11] A. Machanavajjhala, D. Kifer, J. Gehrke, M. Venkitasubramaniam, l-diversity: Privacy beyond k-anonymity, ACM Transactions on Knowledge Discovery from Data 1 (1) (2007) 3.

[12] C. Dwork, Differential Privacy, in: Automata, languages and programming, Vol. 4052, Springer, Venice, Italy, 2006, pp. 1–12.

[13] C. Dwork, A. Roth, The Algorithmic Foundations of Differential Privacy, Vol. 9, Now Publishers, 2013.

[14] F. McSherry, K. Talwar, Mechanism Design via Differential Privacy, in: 48th Symposium on Foundations of Computer Science, IEEE, 2007, pp. 94–103.

[15] M. Hardt, K. Ligett, F. D. McSherry, A Simple and Practical Algorithm for Differentially Private Data Release, in: Advances in Neural Information Processing Systems, 2012, pp. 2339–2347.

[16] S. Fletcher, M. Z. Islam, A Differentially Private Decision Forest, in: 13th Australasian Data Mining Conference, Conferences in Research and Practice in Information Technology, Sydney, Australia, 2015, pp. 1–10.

[17] S. Fletcher, M. Z. Islam, A Differentially-Private Random Decision Forest using Reliable Signal-to-Noise Ratios, in: 28th Australasian Joint Conference on Artificial Intelligence, Lecture Notes

in Computer Science, Springer, 2015, pp. 192–203.

[18] G. Jagannathan, K. Pillaipakkamnatt, R. N. Wright, A practical differentially private random decision tree classifier, Transactions on Data Privacy 5 (2012) 273–295.

[19] J. Brickell, V. Shmatikov, The cost of privacy: destruction of data-mining utility in anonymized data publishing, in: 14th SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2008, pp. 70–78.

[20] B. Fung, K. Wang, P. Yu, Top-down specialization for information and privacy preservation, in: 21st International Conference on Data Engineering, IEEE, 2005, pp. 205–216.

[21] L. Breiman, Random forests, Machine learning 45 (1) (2001) 5–32.

[22] M. Z. Islam, H. Giggins, Knowledge discovery through SysFor: a systematically developed forest of multiple decision trees, in: 9th Australasian Data Mining Conference, Australian Computer Society, Inc., Ballarat, Australia, 2011, pp. 195–204.

[23] A. Evfimievski, R. Srikant, R. Agrawal, J. Gehrke, Privacy preserving mining of association rules, Information Systems 29 (4) (2004) 343–364.

[24] J. Han, H. Cheng, D. Xin, X. Yan, Frequent pattern mining: current status and future directions, Data Mining and Knowledge Discovery 15 (1) (2007) 55–86.

[25] P.-N. Tan, V. Kumar, J. Srivastava, Selecting the right interestingness measure for association patterns, in: 8th SIGKDD International Conference on Knowledge Discovery and Data Mining, Vol. 2, ACM, New York, USA, 2002, p. 32.

[26] G. Webb, D. Brain, Generality is predictive of prediction accuracy, in: Pacific Rim Knowledge Acquisition Workshop, 2002, pp. 117–130.

[27] E. Dasseni, V. Verykios, A. K. Elmagarmid, E. Bertino, Hiding Association Rules by Using Confidence and Support, in: Information Hiding, Purdue University, Springer Berlin Heidelberg, 2001, pp. 369–383.

[28] B. Padmanabhan, A. Tuzhilin, Small is beautiful: discovering the minimal set of unexpected patterns, in: 6th SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2000, pp. 54–63.

[29] N. Zhong, Y. Yao, M. Ohshima, Peculiarity oriented multidatabase mining, IEEE Transactions on Knowledge and Data Engineering 15 (4) (2003) 952–960.

[30] S. Fletcher, M. Z. Islam, Measuring Information Quality for Privacy Preserving Data Mining, International Journal of Computer Theory and Engineering 7 (1) (2015) 21–28.

[31] L. Geng, H. J. Hamilton, Interestingness measures for data mining: a survey, ACM Computing Surveys 38 (3) (2006) 1–32.

[32] P.-N. Tan, V. Kumar, J. Srivastava, Selecting the right objective measure for association analysis, Information Systems 29 (4) (2004) 293–313.

[33] B. Vaillant, P. Lenca, S. Lallich, A clustering of interestingness measures, in: Discovery Science, Springer, 2004, pp. 290–297.

[34] L. Breiman, J. Friedman, C. Stone, R. Olshen, Classification and regression trees, Chapman & Hall/CRC, 1984.

[35] K. Bache, M. Lichman, UCI Machine Learning Repository (2013).
URL `http://archive.ics.uci.edu/ml/`

[36] E. Bertino, D. Lin, W. Jiang, A survey of quantification of privacy preserving data mining algorithms, Privacy-Preserving Data Mining (2008) 1–20.

[37] S. Fletcher, M. Z. Islam, An Anonymization Technique using Intersected Decision Trees, Journal of King Saud University - Computer and Information Sciences 27 (3) (2015) 21.

[38] B. Fung, K. Wang, R. Chen, P. Yu, Privacy-preserving data publishing: A survey of recent developments, ACM Computing Surveys 42 (4) (2010) 1–53.

[39] K. Mancuhan, C. Clifton, K -Nearest Neighbor Classification Using Anatomized Data, Computing Research Repository (CoRR) abs/1610.0 (2016) 10.

[40] B. Fung, K. Wang, L. Wang, M. Debbabi, A framework for privacy-preserving cluster analysis, in: International Conference on Intelligence and Security Informatics, IEEE, 2008, pp. 46–51.

[41] B. Fung, K. Wang, A. W.-C. Fu, P. S. Yu, Introduction to Privacy-Preserving Data Publishing: Concepts and Techniques, CRC Press, 2010.

[42] S. Mukherjee, Z. Chen, A. Gangopadhyay, A privacy-preserving technique for Euclidean distance-based mining algorithms using Fourier-related transforms, VLDB Journal 15 (4) (2006) 293–315.

[43] J. Herranz, S. Matwin, J. Nin, V. Torra, Classifying data from protected statistical datasets, Computers and Security 29 (8) (2010) 875–890.

[44] S. Rana, S. K. Gupta, S. Venkatesh, Differentially private random forest with high utility, in: IEEE International Conference on Data Mining, IEEE, 2016, pp. 955–960.

[45] S. Yu, G. Fung, R. Rosales, S. Krishnan, Privacy-preserving cox regression for survival analysis, in: 14th SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2008, p. 1034.

[46] C. van Rijsbergen, Information Retrieval, Butterworth, 1979.

[47] J. Hanley, B. McNeil, The meaning and use of the area under a receiver operating characteristic (ROC) curve, Radiology 143 (1) (1982) 29–36.

[48] D. J. Hand, Measuring classifier performance: a coherent alternative to the area under the ROC curve, Machine Learning 77 (1) (2009) 103–123.

[49] J. M. Lobo, A. Jiménez-valverde, R. Real, AUC: A misleading measure of the performance of predictive distribution models, Global Ecology and Biogeography 17 (2) (2008) 145–151.

[50] G. Menardi, N. Torelli, Training and assessing classification rules with imbalanced data, Data Mining and Knowledge Discovery 28 (1) (2014) 92–122.

[51] A. Freitas, Comprehensible classification models: A position paper, ACM SIGKDD Explorations Newsletter 15 (1) (2013) 1–10.

[52] K. Wang, P. Wang, A. W. Fu, R. C. W. Wong, Generalized bucketization scheme for flexible privacy settings, Information Sciences 348 (1) (2016) 377–393.

[53] X. Xiao, Y. Tao, Personalized privacy preservation, in: SIGMOD International Conference on Management of Data, ACM, 2006, pp. 229–240.

[54] M. Hua, J. Pei, A Survey of Utility-based Privacy-Preserving Data Transformation Methods, in: Privacy-Preserving Data Mining: Models and Algorithms, Springer, 2008, Ch. 9, pp. 207–237.

[55] J. R. Quinlan, C4.5: programs for machine learning, 1st Edition, Morgan kaufmann, 1993.

[56] A. Friedman, A. Schuster, Data Mining with Differential Privacy, in: 16th SIGKDD Conference on Knowledge Discovery and Data Mining, ACM, Washington, DC, USA, 2010, pp. 493–502.

[57] S. Kullback, R. Leibler, On information and sufficiency, The Annals of Mathematical Statistics 22 (1) (1951) 79–86.

[58] D. Kifer, J. Gehrke, Injecting utility into anonymized datasets, in: SIGMOD International Conference on Management of Data, ACM, New York, New York, USA, 2006, pp. 217–228.

[59] Y. Kameya, K. Hayashi, Bottom-Up Cell Suppression that Preserves the Missing-at-random Condition, in: International Conference on Trust and Privacy in Digital Business, Springer, 2016, pp. 65–78.

[60] O. Pele, M. Werman, The quadratic-chi histogram distance family, Lecture Notes in Computer Science 6312 (2) (2010) 749–762.

[61] C. Ferri, J. Hernández-Orallo, R. Modroiu, An experimental comparison of performance measures for classification, Pattern Recognition Letters 30 (1) (2009) 27–38.

[62] M. Sokolova, G. Lapalme, A systematic analysis of performance measures for classification tasks, Information Processing & Management 45 (4) (2009) 427–437.

[63] S. Fletcher, M. Z. Islam, Quality evaluation of an anonymized dataset, in: 22nd International Conference on Pattern Recognition, IEEE, Stockholm, Sweden, 2014, pp. 3594–3599.

[64] V. Estivill-Castro, L. Brankovic, Data Swapping: Balancing Privacy Against Precision in Mining for Logic Rules, in: Data Warehousing and Knowledge Discovery, Springer, Berlin, 1999, pp. 389–398.

[65] M. Felkin, Comparing classification results between n-ary and binary problems, in: Quality Measures in Data Mining, Springer, 2007, Ch. 12, pp. 277–301.

[66] K. Pearson, On lines and planes of closest fit to systems of points in space, The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 2 (11) (1901) 559–572.