Towards Machine Learning-Assisted Output Checking for Statistical Disclosure Control

Josep Domingo-Ferrer and Alberto Blanco-Justicia

Universitat Rovira i Virgili,
Department of Computer Engineering and Mathematics,
CYBERCAT-Center for Cybersecurity Research of Catalonia,
UNESCO Chair in Data Privacy,
Av. Països Catalans 26, 43007 Tarragona, Catalonia
{josep.domingo,alberto.blanco}@urv.cat

Abstract. There is an increasing demand by researchers to access the microdata (data on individual persons or enterprises) collected by national statistical institutes or other data controllers. If microdata are personally identifiable information, the most usual way for data controllers to share them in a way compliant with the privacy legislation (notably the EU General Data Protection Regulation) is to release anonymized microdata. Yet, data analysts often need access to the original microdata in order to avoid the information loss caused by anonymization. To answer that need, safe access centers (on physical premises or on-line) have been set up by several national statistical institutes. In these centers, users can run their analyses on original data using the controller's software, and the controller checks the outputs of the users' analyses before returning those outputs to them, in order to make sure users do not take home any result that might leak the confidential microdata on which it has been computed. Output checking is currently implemented with human checkers, which is expensive and slow, especially because checkers need to have specific statistical expertise. In this work, we explore the use of machine learning to partially automate output checking. We follow the rule-based approach and our empirical results show that our system can generalize the rules it is trained on. In conclusion, output checking assisted by machine learning seems to work well and should be trialed in safe access centers and decentralized data marketplaces.

Keywords: Output checking · Statistical disclosure control · Safe data access centers · Privacy · Machine learning

1 Introduction

Researchers want data that are as accurate as possible to reach meaningful and trustworthy conclusions. Microdata, that is, data at the level of individual persons or enterprises, are in increasing demand. Privacy legislation, epitomized by the European Union's General Data Protection Regulation [3], prevents data controllers from sharing for secondary use microdata that contain

personally identifiable information (PII). The most usual solution is for the controller to anonymize microdata before releasing them for secondary use [2,5]. Yet, anonymization entails information loss and hence the analyses on anonymized data may not be entirely trustworthy. For this reason, researchers often require access to the original microdata.

Some data controllers, such as those involved in the nascent decentralized data markeplaces, such as Ocean [6], intend to sell not only anonymized data (data-as-a-service) but also the possibility of running computations on the original data (compute-to-data). Yet, they offer no solution to avert possible data leakages associated with the results of computations.

Other data controllers, especially national statistical institutes and data archives, have set up safe access centers as an alternative for those situations in which resarchers cannot use anonymized data. A safe access center may be a physical facility to which the researcher must travel or an on-line service that the researcher can remotely access. Whatever the case, it is a controlled environment in which the researcher runs her analyses using software provided by the controller and is under monitoring by the controller's staff during her entire work session.

A salient feature of safe access centers is that any output of the researcher's analysis is checked by the data controller's staff before returning it to the researcher [1,4]. The purpose of output checking is to make sure that the researcher will not take home any result that might leak the confidential microdata on which it has been computed.

Highly expert output checkers can follow the so-called principles-based model [1, 4]. In this model, no output is ruled in or out in advance. Rather, checkers collaborate with researchers and take the entire context of the analysis into account to make a decision on whether an output is safe enough to be returned or not. Although this model is quite costly, it minimizes the probabilities of false positives (labeling an output as safe when in fact it leaks sensitive information) and false negatives (labeling as unsafe an output that actually leaks no confidential information).

An easier alternative that requires less interaction and expertise on the checker's side is the rule-based model. In this case, the checker uses simple rules of thumb to label an output as safe or unsafe. The price paid is a higher probability of false positives and false negatives.

Contribution and plan of this paper

Output checking currently relies on human checkers. Even if they guide themselves by rules rather than principles, checking is time-consuming and hence expensive and slow. Besides, it is not easy for data controllers to appoint dedicated output checkers: staff with the required statistical expertise are difficult to recruit and output checking is often not regarded as a core task.

We propose to relieve some of the burden of output checking by (partially) automating it via a machine learning approach. This can be useful to all kinds

of controllers, from national statistical institutes to decentralized data marketplaces.

The principles-based model is definitely very difficult to automate, because it requires contextual input to be obtained from the interaction between checkers and researchers. In contrast, the rule-based approach is more amenable to automation, as rules can easily be learned using machine learning.

Taking as a starting point the rules set forth in [1], we create synthetic output checking log files based on different subsets of rules. Then we train deep learning models on each synthetic log file, and we examine how well the rules used to generate the log file have been learned and, more importantly, how the rules *not* used to generate the log file have also been learned. Our results show that our deep learning approach can generalize the rules embedded in the training data, and hence captures the general flavor of safe and unsafe outputs. Admittedly, our system does not completely eliminate the need for human checking, but it can be used to reduce the human workload to filtering out any false positives, that is, outputs labeled as safe by our system which turn out to be unsafe under a more sophisticated checking.

The rest of this paper is organized as follows. Section 2 rewrites the checking rules proposed in [1] in view of using them to create synthetic output checking logs. Section 3 describes how to generate synthetic training data and test data from the rewritten rules. Section 4 reports experimental work and assesses how the deep learning models learned can generalize the rules embedded in the training data. Conclusions and future work suggestions are summarized in Section 5.

2 Rewriting checking rules for synthetic log generation

In [1, 4], rules of thumb are proposed to decide whether an output can be safely returned to the researcher. Both documents propose similar rules based on similar rationales.

For the sake of concreteness, we take the rules proposed in [1], because they are easier to automate than those in [4]. We rewrite the rules in terms of the following attributes: *AnalysisType*, *Output*, *Confidential*, *Context* and *Decision*. In this way, we get:

RULE 1.

AnalysisType: FrequencyTable
Output: Number of units in each cell.
Confidential: YES/NO (YES means the data on which the frequency table is computed are confidential).
Decision: YES/NO
The decision is NO, that is, the output is not returned if data are confidential AND {some cell contains less than 10 units OR a single cell contains more than 90% of the total number of units in a row or column}.

RULE 2.

AnalysisType: MagnitudeTable

Output: Magnitudes in each cell (average or total).
Confidential: YES/NO
Context: Number of units in each cell, and percentage of cell total represented by the maximum contribution to the cell.
Decision: YES/NO
The decision is NO if data are confidential AND {some cell contains less than 10 units OR a single cell contains more than 90% of the units in a row or column OR in some cell the largest contributor contributes more than 50% of cell total}.

RULES 3a/3b/3c.
AnalysisType: Maximum/Minimum/Percentile
Output: Value of Maximum/Minimum/Percentile.
Confidential: YES/NO
Decision: YES/NO
The decision is NO if data are confidential.

RULE 4.
AnalysisType: Mode
Output: Modal value
Confidential: YES/NO
Context: Sample size.
Decision: YES/NO
The decision is NO if {data are confidential AND the frequency of the modal value is more than 90% of the sample size}.

RULES 5a/5b/5c/5d.
AnalysisType: Mean/Index/Ratio/Indicator
Output: Value of the statistic.
Confidential: YES/NO
Context: Sample size, percentage of sample total represented by the largest value in the sample.
Decision: YES/NO
The decision is NO if {data are confidential AND {sample size < 10 OR a single contribution accounts for more than 50% of the sample total}}.

RULE 6.
AnalysisType: ConcentrationRatio
Output: Value of the ratio.
Confidential: YES/NO
Context: Sample size, percentage of sample total represented by the largest value in the sample.
Decision: YES/NO
The decision NO if {data are confidential AND {sample size < 10 OR a single contribution accounts for more than 90% of the sample total}}.

RULES 7a/7b/7c.
AnalysisType: Variance/Skewness/Kurtosis
Output: Value of the statistic.
Confidential: YES/NO
Context: Sample size.

Decision: YES/NO
The decision is NO if {data are confidential AND sample size < 10}.

RULE 8.
AnalysisType: Graph
Output: Graph
Confidential: YES/NO
Decision: YES/NO
The decision is NO if data are confidential.

RULES 9a/9b.
AnalysisType: LinearRegressionCoefficients/
NonLinearRegressionCoefficients
Output: Value of coefficients.
Confidential: YES/NO
Context: Intercept is to be returned?
Decision: YES/NO
The decision is NO if {data are confidential AND intercept is one of the coefficients to be returned}

RULE 10.
AnalysisType: RegressionResiduals/RegressionResidualsPlot
Output: Values of residuals/Plot of residuals.
Confidential: YES/NO
Decision: YES/NO
The decision is NO if data are confidential.

RULES 11a/11b.
AnalysisType: TestStatistic_t/TestStatistic_F
Output: Value of statistic.
Confidential: YES/NO
Context: Degrees of freedom.
Decision: YES/NO
The decision NO if {data are confidential AND degrees of freedom < 10}.

RULE 12.
AnalysisType: FactorAnalysis
Output: Factor scores
Decision: YES

RULE 13.
AnalysisType: Correlations
Output: Matrix of correlation coefficients.
Confidential: YES/NO
Context: Number of units contributing to each correlation coefficient.
Decision: YES/NO
If data are confidential, then the decision is NO for those coefficients that are -1,0,-1 OR that have been computed on less than 10 units.

RULE 14.
AnalysisType: CorrespondenceAnalysis
Output: Loadings of factors
Confidential: YES/NO

Context: Number of variables, sample size
Decision: YES/NO
The decision is NO if {data are confidential AND {number of variables < 2 OR sample size < 10}}

3 Generation of synthetic training and test data

In this section we discuss how to generate synthetic data from the above rules that can be used to train and test a deep learning model. On the one side, we will generate training data from a subset of the rules and, on the other side, we will generate test data from *all* the rules. The purpose of the test data is to enable an assessment of how well the model trained with the training data has been able to learn not only the rules embedded in the training data but also the rules that were not embedded (by generalizing the former).

We realize that records in a training data set should be of fixed length, which allows them to be fed to the inputs of, say, a neural network model. To that end, we need to derive a record schema of fixed length that can describe the decisions made by all the above rules.

However, there are some outputs in the above rules that have a variable number of components: frequency tables (Rule 1), magnitude tables (Rule 2), linear regression coefficients (Rule 9a), non-linear regression coefficients (Rule 9b), regression residuals (Rule 10), factor scores (Rule 12), correlation coefficients (Rule 13) and loadings of factors (Rule 14). To deal with that problem, we will split those rules into rules that *separately apply to each single output component*, *e.g.* to each single table cell, regression coefficient, regression residual, factor score, correlation coefficient or factor loading. Then we can always create a post-processing rule whereby the entire output is only returned to the researcher if all its components are labeled as safe. In what follows we will denote by Rule i_s, the *split* version of Rule i for a single output component.

With the above arrangement, we have a single output component for all split rules. Yet, the number of context components for split rules increases with respect to the original rules, because more context on a split output (*e.g.* a cell) within the total output (*e.g.* a table) is required to make a decision on the split output. Let us assess the number of context attributes required for each rule:

- For Rule 1_s, we need a context attribute *PercentageRows* that contains the maximum percentage the cell total represents over all the totals of rows and columns the cell is part of.
- For Rule 2_s, we need a context attribute *CellUnits* that stores the number of units in the cell, a second context attribute *PercentageRows* that contains the maximum percentage the cell total represents over all the totals of rows and columns the cell is part of, and a third context attribute *PercentageCellTotal* that contains the percentage of the cell total represented by the maximum contribution to the cell.
- For Rules 3a, 3b, 3c and 8, no context attributes are needed.

- For Rule 4, one context attribute *SampleSize* is needed to store the sample size.
- For Rules 5a, 5b, 5c, 5d and 6 a context attribute *SampleSize* is needed to store the sample size and another context attribute *PercentageSampleTotal* is needed to store the percentage of the sample total represented by the largest value in the sample.
- For Rules 7a, 7b and 7c, a context attribute *SampleSize* is needed to store the sample size.
- For Rules $9a_s$ and $9b_s$, one context attribute *Intercept* is needed that stores YES if the corresponding coefficient is the intercept and NO otherwise.
- For Rules 10_s (split rule for each residual) and 12_s (split rule for each factor score) no context attributes are needed.
- For Rule 11a, one context attribute *DegreesOfFreedom* is needed to store the degrees of freedom for the t test statistic.
- For Rule 11b, two context attributes *DegreesOfFreedom* and *DegreesOfFreedom2* are needed to store the degrees of freedom of the F test statistic.
- For Rule 13_s (split rule for each correlation coefficient), one context attribute *CellUnits* is needed that contains the number of units contributing to the correlation coefficient.
- For Rule 14_s (split rule for each factor loading), a context attribute *NumberOfVariables* is needed for the number of variables and another attribute *SampleSize* is needed for the sample size.

Therefore, the schema that can describe the decisions made by all rules is formed by the following superset of attributes: *AnalysisType*, *Output*, *Confidential*, *CellUnits*, *PercentageRows*, *PercentageCellTotal*, *SampleSize*, *PercentageSampleTotal*, *Intercept*, *DegreesOfFreedom*, *DegreesOfFreedom2*, *NumberOfVariables* and *Decision*.

Given the above schema, a synthetic record to describe an instance of a certain Rule i can be generated as follows:

1. Initialize *AnalysisType* to the analysis corresponding to Rule i.
2. Randomly choose an output that is compatible with the analysis type. *E.g.* in Rule 13_s the output is a correlation coefficient and hence it must lie in the interval $[-1, 1]$.
3. Randomly set *Confidential* to YES or NO.
4. Randomly choose context attributes that fit the expected semantics for the analysis type. *E.g.* in Rules $9a_s$ and $9b_s$, *Intercept* must be YES/NO, whereas the other context attributes must be blank; in Rules 11a and 11b, the context attributes must be natural numbers expressing degrees of freedom for the t and F distributions, respectively.
5. Finally, compute *Decision* according to the decision algorithm for the rule.

Given any subset of rules, the above procedure can be applied for each rule in the subset as many times as desired. The result can be used as synthetic training data for a deep learning model to learn the rules in the set.

To obtain test data, the above procedure should be used for the entire set of rules. In this way, one can test how well the rules in the training data have been learnt, and how well the deep learning model can generalize to capture the rules not present in the training data.

4 Experimental work

We took the rules identified in the previous section, and we unified similar rules having the same decision algorithm. That is, we merged Rules 3a, 3b, 3c into a Rule 3*, Rules 5a, 5b, 5c, 5d into a Rule 5*, Rules 7a, 7b, 7c into a Rule 7*, Rules $9a_s$, $9b_s$ into a Rule 9^*_s, and Rules 11a, 11b into a Rule 11*. This left us with 14 total rules.

Then, following the procedure in Section 3, we generated a synthetic data set with $200,000$ training samples, with each of the 14 rules contributing approximately $14,700$ samples, half of which with positive decisions (the analysis results can be released) and half with negative decisions. As an example, in Table 1 we list a few generated records.

Table 1. Sample of generated records

AnalysisType	Output	Confidential	CellUnits	PercentageRows	PercentageCellTotal	SampleSize	PercentageSampleTotal	Intercept	DegreesOfFreedom	DegreesOfFreedom2	NumberOfVariables	Decision
2	0.658	0	0	0	0	508	0	0	0	0	0	True
3	0.613	1	0	0	0	358	0	0	0	0	0	True
11	0.568	1	0	0	0	9	0	0	0	0	0	False
7	0.564	0	0	0	0	955	68.5	0	0	0	0	True
9	0.337	1	0	0	0	763	5.8	0	0	0	0	False

As a machine learning model, we chose a neural network because deep neural networks have demonstrated the capacity to generalize well, and several frameworks allow for easy prototyping and deployment of neural models. Other machine learning models, such as gradient boosting, have also shown good performance on structured data, such as microdata sets. Thus, we trained feedforward neural network with 2 hidden layers of 64 units each and obtained a 94.08% accuracy, a 4.2% false positive rate and a 7.4% false negative rate. Note that false positives indicate outputs that should not be released but whose decision is YES (release) and false negatives indicate outputs that could be released without privacy risk but whose decision is NO (do not release). We are mainly interested in

a low false positive rate (FPR), since false positives are those that are dangerous for the privacy of the respondents on whose data the outputs are computed.

Next, we conducted a series of experiments to find out if a neural network can generalize when exposed to samples generated from rules it has not been exposed to during training. First, we generated a testing data set that contains samples generated using the 14 rules. This testing data set was used throughout all experiments. Then, from a number n of rules ranging from 1 to 14 we generated 100 training data sets using random subsets of n rules. That is, we built 100 data sets with samples generated from random subsets of one rule, 100 data sets with samples generated from random subsets of 2 rules, and so on, which yielded $1,400$ data sets with $200,000$ training samples each. We trained a neural network like the one described above for each of the training data sets and tested it against the previously described single testing data set whose samples were generated using all 14 rules. The source code and the results of our experiments are available in GitHub[1].

Figure 1 displays the distributions of obtained accuracies with respect to the number n of rules used to generate the training data sets. The figure shows how the accuracy of the trained models increases with the number n of rules. For a single rule ($n = 1$), however, the figure indicates that some data sets result in accuracies over 80%, although the median accuracy sits below 70% and the mode is below 60%. From $n = 6$ rules or more, most of the generated training data sets result in accuracies over 80%.

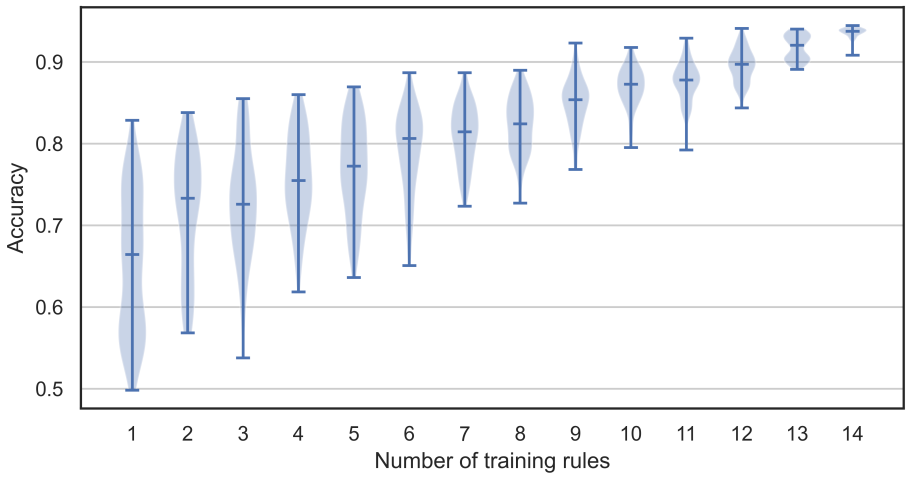


Fig. 1. Accuracy of the models with respect to the number n of rules used to generate the training sets

[1] `https://github.com/ablancoj/output-checking`

Figure 2 displays the false positive rate (red) and the false negative rate (blue) for a number of rules used to generate the training data sets ranging from $n = 1$ to $n = 14$.

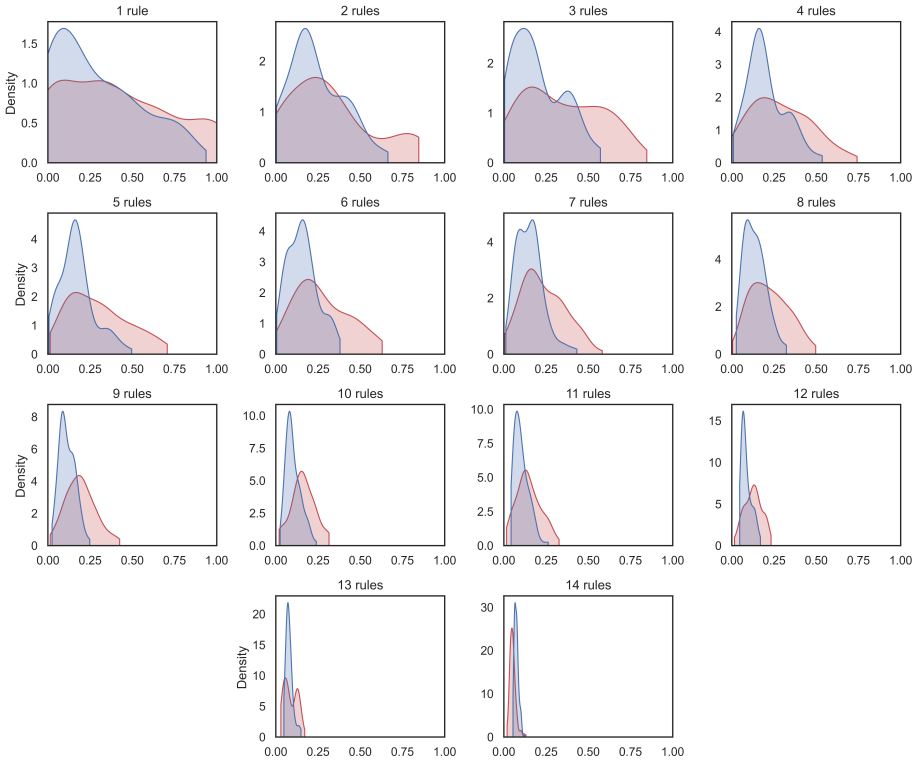


Fig. 2. False positive rate (red) and false negative rate (blue) for a number of rules used to generate the training sets ranging from $n = 1$ to $n = 14$

As mentioned above we are especially interested in the FPR, since it corresponds to released outputs that might reveal private information about the respondents they were computed on. While the false positive rate measures the privacy risk, the false negative rate measures the utility loss, because it corresponds to outputs that are not released even though they could be usefully returned to analysts without privacy risk. As expected, both rates decrease as more rules are considered when generating the training data sets. We also see that for $n = 8$ rules or more, the FPR stays below 50% and concentrates around $15 - 20\%$.

5 Conclusions and future research

We have presented an approach that leverages machine learning to assist human experts in output checking at safe data access centers. Our system follows the rule-based model, and we have shown that it can generalize the rules it is trained on. In our opinion, automating output checking is a pressing need for safe access centers and decentralized data marketplaces to take off.

A limitation of the presented research is that it does not use real log files obtained by the current manual output checking services, because such data are scarce and not public. Future research will strive to gather such data to further validate our approach. We also aim at increasing the level of automation of the entire process. Ideally, given the code of the analysis submitted by the analyst, it should be possible to automatically derive all the inputs required to make rule-based decisions. Optimizing the set of rules for maximum coverage and minimum overlap is another possible direction. Finally, extending automation to the principles-based model is also an important and daunting challenge.

Acknowledgments

Partial support is acknowledged from the European Commission (projects H2020-871042 "SoBigData++" and H2020-101006879 "MobiDataLab"), the Government of Catalonia (ICREA Acadèmia Prize to J. Domingo-Ferrer and grant 2017 SGR 705), and the Spanish Government (project RTI2018-095094-B-C21 "Consent"). The authors are with the UNESCO Chair in Data Privacy, but the views in this paper are their own and are not necessarily shared by UNESCO.

References

1. S. Bond, M. Brandt and P.-P. de Wolf. *Guidelines for the Checking of Output Based on Microdata Research*. Deliverable D11.8, project FP7-262608 "DwB: Data without Boundaries", 2015. `https://ec.europa.eu/eurostat/cros/system/files/dwb_standalone-document_output-checking-guidelines.pdf`
2. J. Domingo-Ferrer, D. Sánchez and J. Soria-Comas. *Database Anonymization: Privacy Models, Data Utility, and Microaggregation-Based Inter-Model Connections*. Morgan & Claypool, 2016.
3. *General Data Protection Regulation*. Regulation (EU) 2016/679. `https://gdpr-info.eu`
4. E. Griffiths, C. Greci, Y. Kotrotsios, S. Parker, J. Scott, R. Welpton, A. Wolters and C. Woods. *Handbook on Statistical Disclosure Control for Outputs (version 1.0)*. Safe Data Access Professionals Working Group, 2019. `https://ukdataservice.ac.uk/media/622521/thf_datareport_aw_web.pdf`
5. A. Hundepool, J. Domingo-Ferrer, L. Franconi, S. Gießing, E. Schulte-Nordholt, K. Spicer and P.-P. de Wolf. *Statistical Disclosure Control*. Wiley, 2012.
6. Ocean Protocol, accessed May 21, 2021. `https://oceanprotocol.com`