Privacy-preserving Anomaly Detection using Synthetic Data

Rudolf Mayer[1 [0000−0003−0424−5999]], Markus Hittmeir[1 [0000−0002−3363−6270]], and Andreas Ekelhart[1 [0000−0003−3682−1364]]

SBA Research gGmbH, Floragasse 7, Vienna, Austria
{rmayer, mhittmeir, aekelhart}@sba-research.org

Abstract. With ever increasing capacity for collecting, storing, and processing of data, there is also a high demand for intelligent knowledge discovery and data analysis methods. While there have been impressive advances in machine learning and similar domains in recent years, this also gives rise to concerns regarding the protection of personal and otherwise sensitive data, especially if it is to be analysed by third parties, e.g. in collaborative settings, where it shall be exchanged for the benefit of training more powerful models. One scenario is anomaly detection, which aims at identifying rare items, events or observations, differing from the majority of the data. Such anomalous items, also referred to as outliers, often correspond to problematic cases, e.g. bank fraud, rare medical diseases, or intrusions , e.g. attacks on IT systems.
Besides anonymisation, which becomes difficult to achieve especially with high dimensional data, one approach for privacy-preserving data mining lies in the usage of synthetic data. Synthetic data comes with the promise of protecting the users' data and producing analysis results close to those achieved by using real data. However, since most synthetisation methods aim at preserving rather global properties and not characteristics of individual records to protect sensitive data, this form of data might be inadequate due to a lack of realistic outliers.
In this paper, we therefore analyse a number of different approaches for creating synthetic data. We study the utility of the created datasets for anomaly detection in supervised, semi-supervised and unsupervised settings, and compare it to the baseline of the original data.

Keywords: Synthetic Data · Anomaly Detection · Machine Learning

1 Introduction

The demand for and practice of data sharing and exchange between different data collecting parties is increasing, often because different data sets complement each other, or because the processing and analysis of data is outsourced. Many interesting knowledge discovery tasks are dependent on large, high quality amounts of data being available. However, when data is sensitive, e.g. when it concerns individuals or is business related, there are certain regulatory and other barriers for data sharing. Still, collaborative analysis of data can be very

beneficial, e.g. when learning from misuse patterns that other parties have been exposed to, such as network intrusions or fraud. Thus, means to enable such data exchange are required. In some cases, anonymisation techniques such as k-anonymity [20] or Differential Privacy [6] can be successfully employed. For a detailed overview on privacy-preserving data publishing methods, see [5]. However, k-anonymity has been shown to be still prone to linkage attacks when adversaries have background knowledge and access to other data sources. Differential privacy, when applied to the model or the output of the model, on the other hand is not applicable for all types of analysis techniques. Both approaches distort the data records to some extend, which, due to the information loss, has potentially negative effects on the utility of the data and the models subsequently trained upon.

Synthetic data is generally considered as data obtained not from direct measurement. In the context of data analysis efforts, it is often considered to be data generated (or synthesised) from a real dataset that, e.g. for privacy considerations, can not be shared. Its aim is to provide a dataset containing records that are similar to the original ones, and that preserve the high-level relationships within the data, without actually disclosing real, single data points. While preserving global properties is often possible, there is generally still a certain loss in data utility, albeit for many settings, this might be acceptable – and potentially be of higher utility than anonymisation techniques.

Approaches can be distinguished on how the model for generating data is obtained. If the original data is not directly used, synthetic data can be generated based on rules and constraints describing the characteristics, requiring an in-depth knowledge on the original data and expert knowledge on the domain. Approaches that can access the original data can learn models from that data, and use these models to generate new data. These methods have the advantage of being applicable to virtually any kind of domain and being scalable to large amounts of attributes. They can also capture correlations that are not easily understood by human experts and for which rule-based methods would fail. The complexity of the employed models for representing the original data is of varying degree, which has implications on how well the synthetic data resembles the original. In this paper, we focus on approaches that utilise the original data.

Another differentiation is whether the dataset is **partially** or **fully** synthetic. The former means that only a subset of the available attributes (or sometimes a subset of the samples), especially those identified to be sensitive, are replaced by synthetically generated ones. For partially synthetic data, data utility is expected to be higher, even though disclosure risks may be higher as well.

Anomaly detection methods typically utilise general outlier detection techniques, but differ when there is a need to distinguish harmless noise from malicious intentions, actions or attacks. Many sub-types of anomaly detection can be distinguished, depending on the type of data, the required output, and whether labels that distinguish between normal and abnormal cases are available [4]. The latter decides whether the detection is to be performed in a supervised, unsupervised or semi-supervised setting. The general settings is that the *anomaly* (or

outlier, minority) type of data points are very few compared to the "normal" data points – sometimes well below 1% of the whole dataset.

Anomaly detection can be considered a rather peculiar use case for employing synthetic data, as this approach tries to specifically preserve the global characteristics, but not of single data points – which could result in information leakage. Anomaly detection techniques, however, try to specifically find the unusual data points among a large amount, and the synthetically created dataset is not expected to produce outlier data points similar to the original ones, as they do not represent global properties. In this paper, we thus evaluate the utility of synthetic data for this task. We consider multiple scenarios, utilising three different approaches for generating synthetic data, and address supervised, semi-supervised and unsupervised methods for anomaly detection based on the generated data. We compare the results to a baseline of methods run on the original dataset.

Section 2 describes related work in synthetic data and anomaly detection. Section 3 describes the setup for our experiments, before we present our evaluation in Section 4. Finally, we present conclusions and future work in Section 5.

2 Related Work

One of the earliest applications of generic synthetic data generation in a data mining context is described by Rubin in [17], where multiple imputation is used to synthetically generate certain columns of datasets. This is thus a setting for partially synthetic data. Specific applications include e.g. the generation of time-series or log data, as they would be encountered for intrusion-detection scenarios [11,3]. More recently, several efforts have been made to generate synthetic images, e.g. in the medical domain for MR images [8].

In general, most approaches to synthetic data generation that are based on existing data, and not (only) on rules, consist of the following steps:

- Learning a representation (**model**) of the original data, with a certain statistical approach (such as estimating a probability density function)
- **Synthesising** the data, i.e. generating new data based on the model
- Optionally, a module to **ensure privacy** of the generated, synthetic data samples, e.g. by means of applying Differential Privacy

A major difference in existing approaches lies in the complexity of the learned model from the data. This can range from relative simple models learning (independent) probability density functions for each attribute, and more sophisticated models that preserve the correlation between attributes e.g. via co-variances, to models that capture very complex correlations e.g. via auto-encoders, or approaches utilising generative adversarial networks (GANs)[7]. In the synthetisation step, some approaches apply further methods for data protection, e.g. by applying Differential Privacy before publishing the synthetic dataset.

While synthetic data has shown to be of utility for multiple data analysis tasks including regression or classification (see e.g. [9]), another major aspect is whether it is actually able to fulfil the promises towards preserving the privacy

of the individuals in the original dataset. Regarding disclosure risks for synthetic data, usually a distinction is made between fully and partially synthetic data. Reiter and Mitra [16] proposed identification disclosure risk estimations for partially synthetic data. They also note that the notion of identification disclosure is *not meaningful* for **fully** synthetic data, since fully synthetic records do not relate to original records in terms of a 1-to-1 correspondence. From an intruder's perspective, the approach to gain information by linking certain synthetic records to individuals is thus not promising, as such links generally do not exist, and matches that have been established with real individuals are only spurious, and not real. Other disclosure risks exist, e.g. attribute disclosure, where it is assumed that the intruder knows the values of certain attributes of their victim and wants to learn the value of some sensitive attribute (called the *target* variable). For a detailed discussion of the latter, see e.g. [10].

For our evaluation in this paper, we employ the following three data synthesizers: The *Synthetic Data Vault* (DV) [14] builds a model based on estimates for the distributions of each column. In order to preserve the correlation between attributes, the synthesizer applies a multivariate version of the Gaussian copula and computes the covariance matrix. The *DataSynthesizer* (DS) [15] provides, among simpler techniques, the 'correlated attribute mode', in which dependencies between attributes are represented by a Bayesian network, learned on the original data. The DataSynthesizer further uses the framework of Differential Privacy, and offers the possibility to inject noise in the model and thus the generated data, by a parameter controlling the magnitude. Finally, *synthpop* (SP) [13] uses as the default method a CART (Classification and Regression Trees) algorithm. The user is able to specify a large number of parameters and may apply a built-in function for disclosure control to the resulting synthetic dataset.

When preparing a dataset for publishing via anonymisation, synthetisation or similar approaches, some information at the level of individual records is invariably removed [5] in order to achieve the desired level of protecting the sensitive information that cannot be shared. Utility evaluation of such datasets can generally be done by two methods. One is to measure certain properties on the sanitised dataset and compare it to the original dataset, such as mean or standard deviation or the data distributions. This evaluation has the advantage of being independent of the final task being carried out on the dataset, but is also generally more difficult to put into an application context. Another approach is to measure the utility on a task, e.g. a supervised classification task, or outlier detection. In this approach, the metric measures the differences in effectiveness of the models on the original dataset compared to the sanitised one. In this paper, we focus on the second approach, as it is more suited for anomaly detection.

Anomaly detection typically utilises general outlier detection techniques [2,21], but differ when there is a need to distinguish harmless noise, fluctuations or various forms of novelty, from malicious intentions, actions or attacks. There are many forms of anomaly detection depending on the nature of the input data, the form of output and the presence or absence of data labels [4]. Point anomalies denote individual data instances that are different from the normal data

points. In a contextual anomaly, data instances are anomalous within a specific context, such as temporal or spatial. Collective anomalies identify a collection of instances as anomalous with respect to the entire dataset. In this paper, we focus primarily on point anomalies. Techniques for this type of anomalies are often the building blocks for contextual and collective anomalies, thus some of our conclusions transfer also to these scenarios.

Anomaly detection is used in a wide variety of contexts, such as fraud detection [1], intrusion detection [22], video surveillance systems, or forensic investigations in general. Specific techniques include unsupervised learning to find structures or patterns in data in the absence of any labels. Recent techniques include Generative Adversarial Networks (GANs), such as *f-AnoGAN* [18]. When labels are present only for the "normal" data points, a semi-supervised methods such as a one-class support vector machine (SVM)[19] or autoencoders are employed. Supervised methods require labels for both the normal and the anomaly cases, and can be addressed by utilising standard classification techniques, such as Logistic Regression, Random Forests, or Support Vector Machines.

3 Experiment Setup

In our experiments, we utilise the *"credit card fraud"* dataset by the ML group at ULB, which was provided publicly on Kaggle[1]. It contains data based on real samples from transactions made by credit cards in September 2013 by European cardholders, gathered over two days. Out of the total of 284,807 transactions, only 492 are identified as fraudulent. The dataset is thus highly unbalanced, as frauds account for only 0.172% of all transactions. The dataset contains 30 input variables. Due to confidentiality reasons, most of the original features have been transformed with Principal Component Analysis (PCA). A projection of the dataset to two dimensions can be seen in Figure 1, created using t-Distributed Stochastic Neighbour Embedding (t-SNE) [12]. It can be observed that some of the anomaly data points are separable, especially the ones seen towards the the top-right of the visualisation. However, also a large number of points seem to be mixed with normal (legit) records, and thus likely difficult to separate.

For the generation of synthetic data, we used the Synthetic Data Vault, the DataSynthesizer and the synthpop package (cf. Section 2). Our primary goal is an unbiased evaluation, and not an optimisation towards a specific synthesizer or target evaluation, but we performed a limited parameter search starting with the standard settings of each synthesizer. We performed the following procedure in order to synthesise and prepare the data for the utility evaluation.

1. We deleted columns in the context of standard feature cleaning, e.g. purely identifying attributes like the 'Time' column in the original dataset.
2. We performed a holdout method, i.e. we randomly split the data into training and test data, such that the size of the latter is 20% of the original table.

[1] `https://www.kaggle.com/mlg-ulb/creditcardfraud`

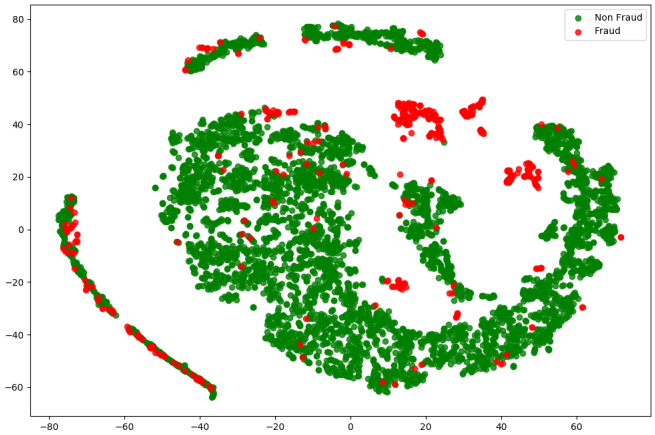

Fig. 1: Projection of the original dataset, generated using t-SNE

3. On the training data, we applied all three data synthesis methods. As output, we generate new, synthetic training data of equal length.

To investigate different configurations regarding its Differential Privacy settings, the DataSynthesizer is applied twice in Step 3. For each of the splits generated in Step 2, we therefore obtain five data files: (i) the original training data, (ii) the training data synthesised by the Synthetic Data Vault, (iii) the training data synthesised by synthpop, (iv) the training data synthesised by the DataSynthesizer without applying Differential Privacy, and (v) the training data synthesised by the DataSynthesizer applying Differential Privacy with the parameter ε.

With this procedure, we obtain datasets that we utilise for training a supervised anomaly detection system (e.g. a classification algorithm). Moreover, we want to separately investigate the behaviour of semi-supervised and unsupervised methods as well. We simulate semi-supervised data by repeating the above outlined procedure for just the data samples that are present in the training set (i.e. after splitting the data), and are labelled as "normal" cases. Finally, for the unsupervised approach, we simply remove the class label before synthesising the data. We thus obtain another five datasets for each of these cases, as for the supervised case, and therefore utilise in total 15 different synthetic training sets for each of the splits generated in Step 2. In addition, we also utilise the test dataset, which is used to estimate the results of the machine learning models on all the training sets. This dataset is not modified in any way, i.e. the synthetisation is performed only to generate the synthetic training sets. This setup will be used for our experiments for anomaly detection.

In Figure 2, we show the same projection as in Figure 1, but for the synthetic data generated by the DataSynthesizer and synthpop. We observe that DataSynthesizer is generating the fraudulent records much closer to the legit ones, while for synthpop separate clusters of fraudulent data appear. This can be an indication for the utility of the datasets for the anomaly detection task.

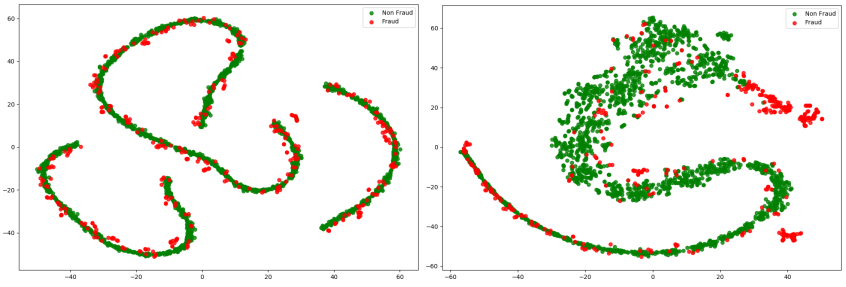

Fig. 2: Projection of the synthetic datasets generated by DataSynthesizer (left) and synthpop (right), generated using t-SNE

For the anomaly detection, we utilise the implementation provided in the Python *scikit-learn* framework[2]. We specifically use these methods, a detailed description of which can be found at the scikit-learn documentation[3]:

- Supervised: Logistic Regression, k-nearest Neighbours, Random Forests, Support Vector Machines (SVMs), Naïve Bayes
- Semi-supervised: One-Class SVMs, Gaussian Mixture Models, Auto Encoder
- Unsupervised: Isolation Forests, Local Outlier Factors

4 Evaluation

Following the generation of synthetic training data, the final step of our experiment is to train machine learning models on the real and the synthesised training datasets and to evaluate these models by comparing their prediction scores on the test data. For a binary task like anomaly detection, which either detects an anomaly or normal behaviour, there can be four different outcomes, depending on the label associated to the data point and the prediction of the machine learning model. Assuming that the fraudulent class is considered the "positive" class, we can distinguish two successful outcomes: *true positives* are data points that are frauds and detected as such, while *true negatives* are data points that are normal (legit) behaviour and identified as such. Two types of errors can occur: *false positives* are legit cases that the model predicts as frauds, while *false negatives* are fraud cases that have not been detected as such, but are considered legit. These are depicted in so-called confusion matrices, which for a binary classification task are 2×2 matrices, where the columns represent the two classes 0 ('legit') and 1 ('fraud'). The first row shows the number of elements in the class that have been predicted to be legit, the second row shows the number of elements that have been predicted to be fraudulent.

Based on this representation, we report the following measures. **Precision** is defined as the ratio of true positives (correctly identified frauds) to all data

[2] Specifically, version 0.22, available at `https://scikit-learn.org/0.22`
[3] https://scikit-learn.org/stable/modules/outlier_detection.html

Table 1: Supervised anomaly detection: confusion matrix

	Naïve Bayes	SVM	K-NN	Random Forest	Log Regression
Real	55626 1224	56840 10	56845 5	56848 2	56836 14
	26 86	33 79	25 87	32 80	26 86
SDV	56838 12	56850 0	56850 0	56850 0	56845 5
	100 12	112 0	112 0	112 0	102 10
DS	55243 1607	56848 2	56848 2	56843 7	56776 74
	17 95	62 50	58 54	49 63	28 84
DSP	56498 352	56850 0	56850 0	56849 1	56745 105
	41 71	112 0	112 0	110 2	70 42
SP	55424 1426	56848 2	56845 5	56849 1	56824 26
	24 88	51 61	39 73	41 71	26 86

Table 2: Supervised anomaly detection: scores

	Naïve Bayes			SVM			K-NN			Random Forest			Log Regression		
	Pr	Re	F2	Pr	Re	F2	Pr	Re	F2	Pr	Re	F2	Pr	Re	F2
Real	6.6	76.8	24.5	88.8	70.5	73.6	94.6	77.7	80.6	97.6	71.4	75.5	86.0	76.8	78.5
SDV	50.0	10.7	12.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	66.7	8.9	10.8
DS	5.6	84.8	22.1	96.2	44.6	50.0	96.4	48.2	53.6	90.0	56.3	60.8	53.2	75.0	69.3
DSP	16.8	63.4	40.8	0.0	0.0	0.0	0.0	0.0	0.0	66.7	1.8	2.2	28.6	37.5	35.3
SP	5.8	78.6	22.4	96.8	54.5	59.7	93.6	65.2	69.4	98.6	63.4	68.3	76.8	76.8	76.8

points predicted as frauds, i.e. how many of the cases the model has *predicted* to be frauds are actually such. **Recall** indicates how many of the frauds have been identified by the model, given as the ratio of the true positives to all cases that have been *labelled* as fraud. Each of these measures alone is not representative, as it is rather easy to optimise one of them, but difficult to have both take high values in conjunction. The **F1 score** provides a unified score, by computing the harmonic mean. The **F2 score** weighs recall higher than precision, which makes it suitable in our application, where it's likely more important to identify most of the anomalies, and a certain amount of false positives can be tolerated. In the following tables, we report these scores in *percent*. It is important to note that classification **accuracy** is less meaningful in our setting. It is defined as the ratio of true positives and true negatives to the size of the dataset. With highly imbalanced data, even models which trivially predict all instances as legit (normal) cases, and do not detect frauds at all, achieve high accuracy scores.

For each of the anomaly detection methods, we performed a random grid search over a number of parameters to optimise the results on the original training set. We then applied these settings, without any further optimisation, also to the models learned on the synthetically created datasets.

Table 1 shows the confusion matrices for the supervised anomaly detection algorithms, the metrics derived thereof are shown in Table 2. We can observe that the anomaly detection is a difficult task already on the original dataset, where Naïve Bayes is achieving a good recall, but has very low precision, and

Table 3: Unsupervised & semi-supervised anomaly detection: confusion matrix

	Isol. Forest		LOF		1-Class SVM		GMM		AutoEncoder	
Real	55335	1515	50756	6094	53655	3195	56838	12	56591	259
	26	86	77	35	19	93	32	80	50	62
SDV	55268	1582	51190	5660	51718	5132	56759	91	56582	268
	34	78	13	99	13	99	31	81	52	60
DS	55185	1665	51279	5571	52465	4385	56309	541	56533	317
	28	84	25	87	14	98	49	63	53	59
DSP	55445	1405	50755	6095	52126	4724	56643	207	56531	319
	33	79	13	99	14	98	53	59	53	59
SP	55128	1722	54283	2567	54110	2740	56719	131	56607	243
	29	83	20	92	21	91	32	80	52	60

seems thus not usable for that task. The other algorithms perform significantly better, with k-Nearest Neighbours scoring best on precision, recall and F2 score.

On the synthetic datasets, we can see that the Synthetic Data Vault is not able to create datasets that can be learned by the supervised methods. The confusion matrices show that SVMs, k-NN and Random Forest cannot identify any of the anomaly data points, but classify them all as the "non-fraud" class, therefore achieving zero precision, recall, and F2 score. Only a very small number of frauds are correctly identified by Naïve Bayes and Logistic Regression, and while there are relatively few non-fraud cases wrongly predicted as fraud, the overall F2 score stays low – it is in the range of approximately 10%, and thus not useful. We can observe similar patterns for the DataSynthesizer when using Differential Privacy – even though larger values of recall are achieved, the more relevant F2 scores are still rather low, being in the range of 35% to 40%.

The models trained on the data synthesised by the DataSynthesizer without Differential Privacy do a better job at detecting frauds than the models trained on the data synthesised by the Synthetic Data Vault. Precision stays relatively high for most settings, but recall drops compared to the original data. Thus, the overall F2 scores drop by 15% to more than 20%, except for Naïve Bayes, which drops only marginally, but from a very low baseline of 24%. This trend of lower recall is slightly inverted for Logistic Regression, where it drops only marginally, by 1.8%, and thus achieves the best results for this synthetic dataset.

The best results on synthetic data are achieved with synthpop. For Logistic Regression, recall stays the same, and precision drops only marginally, and thus the overall F2 score is very close to the original dataset, lower just by 1.8%. For the other classifiers, the degradation is a bit larger, mostly with a lower recall, which results in an F2 score lower by approximately 10%. In overall, however, the scores are still relatively close to the original dataset.

In Table 3, we can see the confusion matrix for the unsupervised (Isolation Forest, Local Outlier Factor) and semi-supervised (One-Class SVM, Gaussian Mixture Model, Auto Encoder) methods. The scores are given in Table 4. We can observe that the unsupervised task, and to some extent also the semi-supervised

Table 4: Unsupervised & semi-supervised anomaly detection: scores

	Isol. Forest			LOF			1-Class SVM			GMM			AutoEncoder		
	Pr	Re	F2	Pr	Re	F2	Pr	Re	F2	Pr	Re	F2	Pr	Re	F2
Real	5.4	76.8	21.0	0.6	31.3	2.7	2.8	83.0	12.4	87.0	71.4	74.1	19.3	55.4	40.3
SDV	4.7	69.6	18.5	1.7	88.4	8.0	1.9	88.4	8.7	47.1	72.3	65.3	18.3	53.6	38.7
DS	4.8	75.0	19.1	1.5	77.7	7.1	2.2	87.5	9.9	10.4	56.3	29.9	15.7	52.7	35.8
DSP	5.3	70.5	20.4	1.6	88.4	7.5	2.0	87.5	9.3	22.2	52.7	41.3	15.6	52.7	35.7
SP	4.6	74.1	18.4	3.5	82.1	14.8	3.2	81.3	13.9	37.9	71.4	60.7	19.8	53.6	39.9

task, is a much harder one. The major impact is on precision, where many methods struggle to achieve high values without reducing recall too much.

For the unsupervised methods, the Isolation Forest obtains the best precision values, and also high recall values. Thus, also in regards to the F2 score, Isolation Forest achieves by far the highest scores in the range of around 20%. It is interesting to note that the synthetic datasets are almost on par with the original data for Isolation Forests. Contrary to before, in this setting, the Synthetic Data Vault and DataSynthesizer with Differential Privacy are not much different than the other approaches – the latter is even the best of the synthetic datasets, achieving an F2 score of 20.4%. Results for the Local Outlier Factor are generally not satisfying. An interesting observation, however, is that all the synthetic datasets outperform the model on the original dataset – albeit still at a low overall score. Only synthpop is closer to the results of the Isolation Forest.

For the One-Class SVM as a semi-supervised method, the results are in between the two unsupervised other approaches. Interestingly, also for this setting, the synthetic dataset generated by synthpop performs better than the original one. Overall, however, the One-Class SVM shows a low precision. We obtain much better results with the Auto Encoder, which is mostly due to a much lower false-positive rate. The number of actual frauds correctly detected is 62 (55%) on the real data, and only marginally lower on most synthesizers, which achieve 59 or 60 correctly identified fraud cases. Precision and thus F2 score are the highest on the dataset created with synthpop, which is overall very close to the real dataset, with a just 0.4% lower score. Similarly to the unsupervised methods, also on this method the Synthetic Data Vault achieves better results than the DataSynthesizer. Finally, the best results on the semi-supervised task are achieved by the Gaussian Mixture Model. On the original dataset, 71.4% of the fraud cases are identified, with a low false-positive rate of only 12 records. The results on synthpop match the recall, but have a lower precision, thus resulting in an overall rather large drop of F2 score by approximately 14%. Again, the Synthetic Data Vault is the best of the synthesising methods, having actually a marginally higher recall, and a still acceptable precision; thus, the overall F2 score drops only around 9% from 74.1% on the original dataset.

While these are still below the results of the supervised approach, the semi-supervised setting is a more difficult task, as it can rely only on labels for the

"normal" cases. It is however a setting of practical application value, as it is generally easier to obtain these labels, and more difficult to label anomaly cases.

5 Conclusions and Future Work

In this paper, we evaluated the utility of synthetic data for the task of anomaly detection, on the example of fraud detection. We considered the setting where the data required to build the models can not easily be shared with the people responsible for the training due to its sensitive nature. In settings where multiple parties would like to collaborate to obtain a more powerful model, facilitating such exchange can be crucial. Synthetic data can be used to create a counterpart of the data that does not represent individual records, but still preserves important characteristics. Thus, the disclosure of sensitive data can be reduced.

In our evaluation, we considered the cases of supervised, semi-supervised and unsupervised anomaly detection. We have assumed that synthetic data might not be usable for such a task, as anomaly detection deals with outliers, and synthetic data generally preserves global characteristics – thus rather the ones of the legit, normal cases. However, the evaluation showed that while anomaly detection is generally a hard task, in specific settings, synthetic data can reach similar effectiveness as the models trained on the original data. It can thus be a viable alternative when the original data cannot be shared, and other forms of data sanitisation, such as anonymisation via k-anonymity, are not feasible.

Future work will extending our evaluation to additional anomaly detection data sets, e.g. on network intrusion, and will include additional detection methods, especially recent approaches like generative adversarial networks (GANs). Further, we will investigate how well synthetic data compares in terms of utility to datasets that need to be anonymised or otherwise treated before they could be shared. We will investigate further inference attacks on synthetic data, especially in the context of trying to infer information on the outliers.

Acknowledgement

This work was partially funded by the BRIDGE 1 programme (No 871267, "WellFort") of the Austrian Research Promotion Agency (FFG), and the EU Horizon 2020 programme under grant agreement No 732907 ("MyHealth-MyData"). SBA Research (SBA-K1) is funded within the framework of COMET — Competence Centers for Excellent Technologies by BMVIT, BMDW, and the federal state of Vienna, managed by the FFG.

References

1. Abdallah, A., Maarof, M.A., Zainal, A.: Fraud detection system: A survey. Journal of Network and Computer Applications **68** (2016)
2. Agyemang, M., Barker, K., Alhajj, R.: A Comprehensive Survey of Numeric and Symbolic Outlier Mining Techniques. Intell. Data Anal. **10**(6) (2006)

3. Barse, E., Kvarnstrom, H., Johnson, E.: Synthesizing test data for fraud detection systems. In: 19th Annual Computer Security Applications Conference. IEEE, Las Vegas, Nevada, USA (2003)
4. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM Computing Surveys **41**(3) (2009)
5. Chen, B.C., Kifer, D., LeFevre, K., Machanavajjhala, A.: Privacy-preserving data publishing. oundations and Trends in Databases **2**(1;2) (2009)
6. Dwork, C.: Differential privacy. In: 33rd International Colloquium on Automata, Languages and Programming (ICALP). Springer, Venice, Italy (2006)
7. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative Adversarial Nets. In: Advances in Neural Information Processing Systems 27. Curran Associates, Inc. (2014)
8. Han, C., Hayashi, H., Rundo, L., Araki, R., Shimoda, W., Muramatsu, S., Furukawa, Y., Mauri, G., Nakayama, H.: GAN-based synthetic brain MR image generation. In: 15th International Symposium on Biomedical Imaging (ISBI). IEEE, Washington, DC (2018)
9. Hittmeir, M., Ekelhart, A., Mayer, R.: On the Utility of Synthetic Data: An Empirical Evaluation on Machine Learning Tasks. In: 14th International Conference on Availability, Reliability and Security (ARES). ACM Press, Canterbury, UK (2019)
10. Hittmeir, M., Ekelhart, A., Mayer, R.: A Baseline for Attribute Disclosure Risk in Synthetic Data. In: 10th ACM Conference on Data and Application Security and Privacy (CODASPY). New Orleans, LA, United States (2020)
11. Lundin, E., Kvarnström, H., Jonsson, E.: A Synthetic Fraud Data Generation Methodology. In: Information and Communications Security. Springer (2002)
12. Maaten, L.v.d., Hinton, G.: Visualizing data using t-SNE. Journal of machine learning research **9**, 2579–2605 (2008)
13. Nowok, B., Raab, G., Dibben, C.: synthpop: Bespoke creation of synthetic data in r. Journal of Statistical Software, Articles **74**(11) (2016)
14. Patki, N., Wedge, R., Veeramachaneni, K.: The synthetic data vault. In: IEEE International Conference on Data Science and Advanced Analytics (DSAA). Montreal, QC, Canada (2016)
15. Ping, H., Stoyanovich, J., Howe, B.: Datasynthesizer: Privacy-preserving synthetic datasets. In: 29th International Conference on Scientific and Statistical Database Management. Chicago, IL, USA (2017)
16. Reiter, J.P., Mitra, R.: Estimating risks of identification disclosure in partially synthetic data. Journal of Privacy and Confidentiality **1**(1) (2009)
17. Rubin, D.B.: Multiple Imputation for Nonresponse in Surveys. Wiley-Interscience (2004)
18. Schlegl, T., Seeböck, P., Waldstein, S.M., Langs, G., Schmidt-Erfurth, U.: f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks. Medical Image Analysis **54** (2019)
19. Smola, A.J., Schölkopf, B.: A tutorial on support vector regression. Statistics and Computing **14**(3) (2004)
20. Sweeney, L.: K-anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **10**(5) (2002)
21. Zimek, A., Campello, R.J., Sander, J.: Ensembles for unsupervised outlier detection: challenges and research questions a position paper. ACM SIGKDD Explorations Newsletter **15**(1) (2014)
22. Zuech, R., Khoshgoftaar, T.M., Wald, R.: Intrusion detection and Big Heterogeneous Data: a Survey. Journal of Big Data **2**(1) (2015)