CD-MAKE 2022, LNCS 13480
This is a self-archived pre-print version of this article.
The final publication is available at Springer via
https://doi.org/10.1007/978-3-031-14463-9_20.

An Empirical Analysis of Synthetic-Data-based Anomaly Detection

Majlinda Llugiqi[1,2] and Rudolf Mayer[1,2]

[1] Vienna University of Technology, Vienna, Austria
[2] SBA Research, Vienna, Austria rmayer@sba-research.org

Abstract. Data is increasingly collected on practically every area of human life, e.g. from health care to financial or work aspects, and from many different sources. As the amount of data gathered grows, efforts to leverage it have intensified. Many organizations are interested to analyse or share the data they collect, as it may be used to provide critical services and support much-needed research. However, this often conflicts with data protection regulations. Thus sharing, analyzing and working with those sensitive data while preserving the privacy of the individuals represented by the data is needed. Synthetic data generation is one method increasingly used for achieving this goal. Using synthetic data would useful also for anomaly detection tasks, which often contains highly sensitive data.
While synthetic data generation aims at capturing the most relevant statistical properties of a dataset to create a dataset with similar characteristics, it is less explored if this method is capable of capturing also the properties of anomalous data, which is generally a minority class with potentially very few samples, and can thus reproduce meaningful anomaly instances. In this paper, we perform an extensive study on several anomaly detection techniques (supervised, unsupervised and semi-supervised) on credit card fraud and medical (annthyroid) data, and evaluate the utility of corresponding, synthetically generated datasets, obtained by various different synthetisation methods. Moreover, for supervised methods, we have also investigated various sampling methods; sampling in average improves the results, and we show that this transfers also to detectors learned on synthetic data. Overall, our evaluation shows that models trained on synthetic data can achieve a performance that renders them a viable alternative to real data, sometimes even outperforming them. Based on the evaluation, we provide guidelines on which synthesizer method to use for which anomaly detection setting.

Keywords: Anomaly Detection · Synthetic Data · Privacy Preserving · Machine Learning

1 Introduction

With increased data collection, also data analysis becomes more wide-spread. One important data analysis task is anomaly detection [5], which aims at finding

unusual behavior in datasets. Anomalies may be caused by variations in machine behavior, fraudulent behavior, mechanical defects, human error, instrument error and natural deviations in populations [15]. Anomalies in data lead to important actionable information in a broad range of application domains, including cyber-security intrusion detection, defect detection of safety-critical devices, credit card fraud and health-care [5].

From the above mentioned examples it becomes clear that many detection tasks will operate on sensitive and personal data, and therefore techniques to use those data while preserving privacy are needed. Thus, data privacy has become a concern also among the anomaly detection research community. Different approaches for data privacy were proposed. One of those is K-anonymity which protects against a single record linking threat [34]. Different releases, however, might be connected together to compromise k-anonymity. Extension of the original concept, such as l-diversity, protect against further risks, such as attribute disclosure. However, these have shown to destroy the data utility too much [3].

Another approach is generating synthetic data. The fundamental concept behind synthetic data is to sample from suitable probability distributions to replace some or all of the original data, while preserving their important statistical features [26]. In this paper, we explore in depth the problem of anomaly detection models learned from synthetic data. We analyse different synthetic data generation methods and different anomaly detection methods, to answer whether it can be used as a suitable surrogate for utilising the original data – which might not be a viable option due to data sharing or usage limitations.

We use three different data synthesizers, and multiple different supervised, semi-supervised and unsupervised techniques to detect anomalies, on two frequently used, benchmark data sets with sensitive data (credit card fraud and annthyroid) in the anomaly detection domain. Moreover, since the imbalanced data problem is a core issue in anomaly detection, for the supervised setting, we balance the data using three separate sampling approaches: oversampling, undersampling, and the Synthetic Minority Oversampling Technique (SMOTE) [6], and compare that effect also on the synthetic data.

The remained of this paper is organised as follows. Section 2 discusses related work and state of the art results. We describe our evaluation setting in Section 3, and discuss results in Section 4. We then conclude in Section 5 and discuss directions for future work.

2 Related Work

Privacy preserving of sensitive data has been the subject of extensive research, and several different approaches have been considered. K-anonymity [34] has been a traditional solution to privacy concerns. However, concerns over residual risks, and lack of utility [3] have given rise to other techniques. One recently widely studied approach is synthetic data generation. Synthetic data is created by building a model based on real-world source data, from which samples are drawn to form a surrogate dataset. While the data is (close to) statistically

indistinguishable from the real dataset, it no longer has a link to real individuals, and thus can be used, exchanged and transfered with less restrictions.

Rubin et al. [32] were among the first to propose synthetic data generation for disclosure control, namely repeated perturbation of the original data as a replacement for the original data. Ping et al. use a Bayesian network-based data synthesis approach, the *Data Synthesizer* [28]. They further also provide an independent attribute mode, which generates data for each attribute independently of the others. The *Synthetic Data Vault* (SDV) offers among other approaches one based on a multivariate version of the Gaussian Copula to model the covariances between the columns in addition to the distributions [27]. Nowok et al. propose a technique based on classification and regression trees (CART) in their *Synthpop* tool [26]. The synthetic values for the attributes are created progressively from their conditional distributions. Acs et al. [2] utilise generative neural networks. They first cluster the initial datasets into k clusters, and build synthesizer models for each cluster.

A high utility of the generated synthetic data is vital to successfully substitute original data. Several earlier works have specifically analysed and measured the utility of synthetic data for certain data analysis tasks, termed the application fidelity [7], e.g. for classification [12] or regression [13], or more generically supervised learning tasks [31], or for specific data types, e.g. microbiome data[14]. An earlier work addressing specific anomaly detection in synthetic data is provided by [23], focusing on a single dataset. We extend their work by considering more datasets, more detection techniques, and the incorporation of sampling methods, which are shows improvements for several of the supervised setting. Further, we re-create state-of-the-art results on the original dataset to find a more viable baseline for adequately assessing the comparative performance of the synthetic data based detectors, and thus achieve higher scores than [23] in the baseline.

Anomaly detection is a form of unbalanced data problems [20]. In anomaly detection, the majority of samples are "normal" data, whereas the minority samples are anomaly data. Fraud detection [39], disease detection [18], intrusion detection [19], identification systems [16], and fault diagnostics [29] are some example application domains. There are several ways to categorize anomaly detection methods. Goldstein et al. [11] distinguished three settings based on the availability of the data as illustrated in Figure 1: **Supervised** anomaly detection refers to a setting in which the data consists of training and test data sets labeled with normal and anomaly instances. **Semi-supervised** anomaly detection also employ training and test datasets, with training data consisting only of normal data, but no anomalies. A model is learnt on the normal class, and then anomalies may be found if they are deviating from that model. **Unsupervised** anomaly detection does not require any labels and no differentiation is made between a training and a test dataset.

A further distinction can be on the learning approach. **Classification-based** techniques learn a machine learning model from a set of labeled examples, and then classify a test instance into one of the classes using that model [5]. These techniques are used in supervised settings.

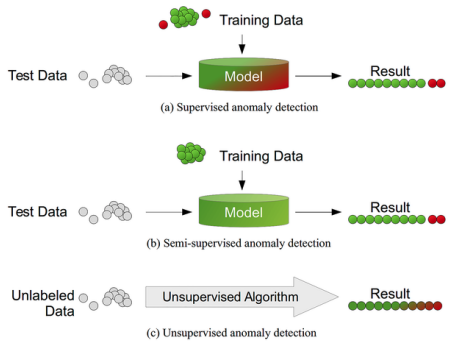

Fig. 1. Different anomaly detection modes depending on the availability of labels [11]

Statistical anomaly detection methods fit a statistical model (typically for normal behavior) to the available data, and then use a statistical inference test to check if an unseen instance fits the distribution [5]. Anomalies are assumed to occur in the low probability areas of the model. Statistical techniques can be used in an unsupervised context, without the necessity for labeled training data, if the distribution estimate phase is robust to data anomalies.

Clustering-based can be one of three categories. The first category is based on the premise that normal behavior data are grouped into clusters whereas anomalies are those samples not belonging to any of the clusters. The second category is based on the idea that anomalies are far away from their nearest cluster centroid, whereas normal data instances are close. One issue is that if data anomalies create clusters on their own, these methods will fail to detect them. To address this problem, a third category is based on the premise that anomalies belong to small or sparse clusters.Clustering is primarily an unsupervised technique. However, approaches from the second category can work in a semi-supervised mode.

Information theoretic approaches uses different information theoretic measures, such as Kolmogorov or entropy, to examine the information content of a data collection [5]. Anomalies are detected based on the assumption that anomalies in data cause inconsistencies in the data set's information content. Information theory techniques can be used in an unsupervised setting.

The specific techniques and algorithms that we have used for anomaly detection are described in Section 3.4.

3 Experiment Setting

In our experiment we used two datasets, *credit card fraud*[3] and *annthyroid*[4]. An overview of the experiment process can be seen in Figure 2. It starts with

[3] https://www.kaggle.com/mlg-ulb/creditcardfraud
[4] https://archive.ics.uci.edu/ml/datasets/Thyroid+Disease

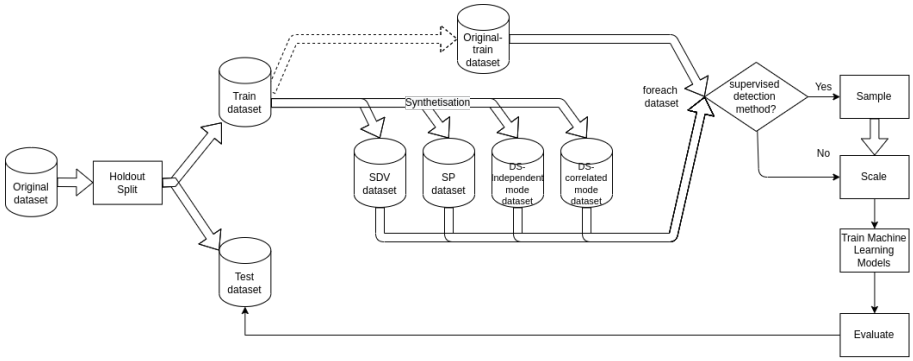

Fig. 2. Experiment flow diagram

the original datasets; to be able to evaluate the synthesizers, both datasets are split into train-validation splits (holdout method). From the train splits of the datasets we generate synthetic data with each synthesizer. After the synthetization step, depending if the target variable is available, the original and synthesized train splits are sampled using one of the sampling techniques mentioned in Section 3.3. As part of the training process, to improve the model performance, the hyperparameters are tuned using a grid search, for which the values of the hyperparameters are shown in 1. In the end the performance (effectiveness) is evaluated on the validation set with the best performing model from the train phase using the F2 score, accuracy and precision, which are described below.

In a two-class problem, we try to tell the difference between anomalies and normal behaviour data. If the anomaly class is considered as "positive" , and the non-anomalies as "negative" class, we distinguish four outcomes: “true positives” for correctly predicted anomalies, “false positive” for normal samples incorrectly predicted anomalies, “true negative” for correctly predicted normal samples as such, and “false negative” for incorrectly predicting an anomaly as normal sample. Based on this, we use the following evaluation metrics: **Precision**, which is the ability of a classification model to identify only the relevant data points, and can be calculated as the number of true positives divided by the number of true positives plus the number of false positives.

Recall, which is the ability of a model to find all the relevant cases within a dataset and can be calculated as the number of true positives divided by the number of true positives plus the number of false negatives.

Precision and recall are not representative on their own, as it is trivial to improve one at the cost of the other, but difficult to have both with high values at the same time. Therefore, we also use the **F1 and F2 scores**, which combines precision and recall. In contrast to the balanced F1, F2 puts an emphasis on recall, which is suitable for anomaly detection, where it is more critical to identify the majority of anomalies, and a certain amount of false positives may be allowed.

3.1 Datasets

The credit card datasetFootnote 3 includes credit card transactions made by European cardholders with two days in September 2013. It contains 492 frauds out of 284,807 transactions, and is thus highly unbalanced, with the positive class (frauds) accounting for just 0.172% of all transactions.

The dataset has 31 features, all of them are numerical. Features V1,...,V28 are the principal components obtained with a Principal Component Analysis (PCA). Further, one attribute holds the amount of the transaction, and the attribute 'Time' represents the time since the first transaction in the dataset.

The Annthyroid dataset Footnote 4 includes patients records from Garavan Institute. This dataset comprises of 7,200 records and has 22 features, all numerical. The target class contains "hyperfunction", "subnormal functioning" and "normal" (not hypothyroid), respectively. As common in literature ([21,10], we grouped hyperfunction and subnormal functioning as one group that represents anomalies in this dataset. Thus, 534 entries are anomalies, i.e. 7.4%.

3.2 Dataset Synthetization

For generating synthetic data we used SDV, Synthpop and two modes of DataSynthesizer, independent attribute and correlated attribute mode. We thus generate four synthetic datasets for each datasets.

3.3 Dataset pre-procesing

As the datasets are highly imbalanced, we employ three well known sampling techniques, namely Random Undersampling, Random Oversampling and Synthetic Minority Oversampling Technique (SMOTE)[6], to potentially improve the performance of supervised techniques (sampling can only be applied if we have labels for both anomalies and normal data). We used the implementation from the python package "imblearn"[5].

3.4 Anomaly Detection Methods

We use different supervised, semi-supervised and unsupervised machine learning techniques from the Python machine-learning library sk-learn [6].

The supervised methods include: (i) Ada Boost , (ii) XGB (extrem gradient boosting), (iii) Gaussian Naive Bayes, (iv) Linear SVC, (v) k-nearest Neighbors, (vi) Random Forest, and (vii) Logistic Regression. This selection of supervised machine learning techniques includes a wide range of different approaches, including probabilistic, linear, and rule-based classifiers, and ensemble techniques.

For semi-supervised techniques, we use: (i) AutoEncoder, and (ii) Gaussian Mixture

The selected unsupervised approaches are: (i) Isolation Forest, (ii) Local Outlier Factor, (iii) One Class SVM,

[5] https://imbalanced-learn.org/
[6] https://scikit-learn.org/stable/modules/outlier_detection.html

Table 1. Parameter grid for supervised methods

Method	Parameter	(Grid) values
GaussianNB	var_smoothing	[$1.0e^{-02}$,$1.0e^{-04}$, $1.0e^{-09}$]
k-NN	n_neighbors	[5, 10, 15]
Random Forest	n_estimators	[100, 200, 300]
Logistic Regression	C	np.logspace(-4, 4, 3)
Linear SVC	C	[1, 100, 1000]
AdaBoost	n_estimators	[100, 200, 300]
XGB	n_estimators	[100, 200, 300]
IsolationForest	n_estimators	[100, 200, 300]
LocalOutlierFactor	n_neighbors	[5, 10, 15]
OneClassSVM	gamma	[$1.0e^{-03}$,$1.0e^{-05}$, $1.0e^{-08}$]
GaussianMixture	n_components; n_init	1; 5
AutoEncoder	epochs; batch-size	10; 128

To improve the results for the supervised methods, we executed a grid search on the training set through a number of parameters and values, shown in Table 1. Each of the parameter values are taken through the full pipeline and evaluated on a five fold cross-validation inside the training set. For evaluation we used an unseen validation set, consisting of 20% of the original data.

4 Results

In this section, we present the results from our evaluation, based on the setup described in Section 3. We primarily discuss the F2 scores, but provide further results details, namely precision and recall for supervised, semi-supervised and unsupervised methods, for both datasets, in section A.

4.1 Credit Card dataset

From Table 2, we can observe that for the credit card dataset, in most cases, sampling methods can not improve the scores, with a few exceptions (such as random oversampling or SMOTE for the already very successful XGB on real data). In average over all classifiers, sampling decreases the performance. For better visual comparison, Figure 3 shows F2 scores for the different synthesizers and supervised methods when no sampling method was used.

We can further see that the performance drops when learning anomaly detection on real versus the synthetic data, implying that in general, it is slightly more difficult to learn an anomaly representation after synthesizing the data. However, in many settings, the drop is relatively small (within a few percent), and there are even a few cases where the best performance on real data is increased after synthetization, e.g. for the Gaussian Naive Bayes (from an albeit low base) and Logistic Regression without sampling.

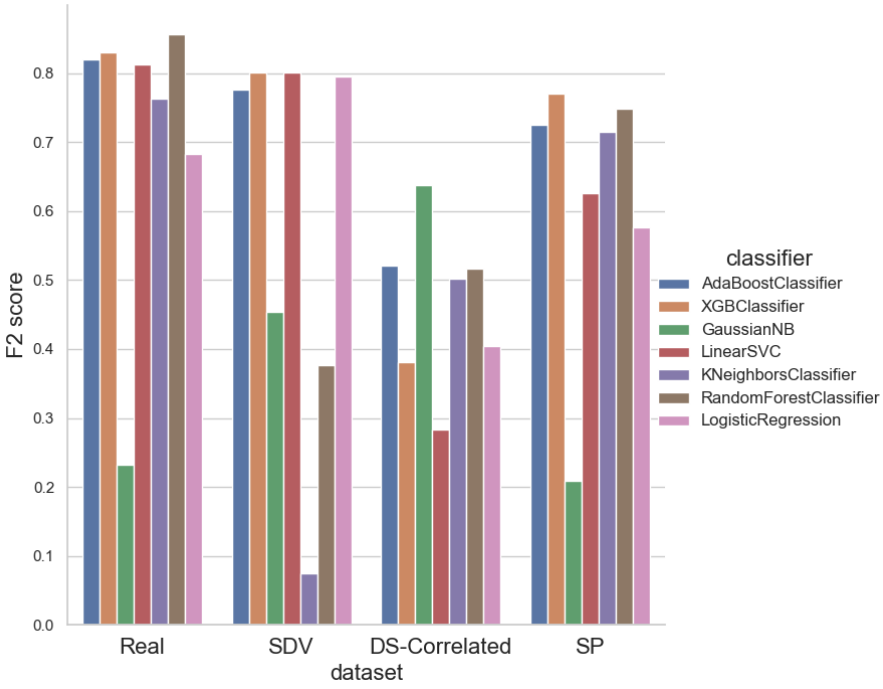

Fig. 3. Credit card: F2 scores for supervised techniques on datasets generated using no sampling

Moreover from Tables 2, 12 and 13 when analyzing the synthetic datasets, we can see that the best performing synthesizers are the Synthetic Data Vault (SDV) and Synthpop (SP); the latter is the best choice for k-NN and Random Forest. Overall, it is difficult to generally recommend which of these synthesizers is the better choice.

Regarding individual classifier performance, from the average columns in Tables 2, 12 and 13 we can observe that XGB achieves a relatively low recall of 77.5% compared to the other supervised methods, however, it has the highest F2 score and precision of 58.3% and 57.5% respectively, making it the most suitable for outlier detection tasks on the credit card dataset. On the other hand, we can see that Gaussian Naive Bayes is not useful for this task, achieving a good recall of 80%, but a very low precision and F2 score, 8.4% and 25.1% respectively, which is 49.1 and 33.2 percentage points less than the best performing classifier on the table, respectively.

When looking only at the sampling methods, it can be observed that on average none of the sampling methods provide a performance increase. There are some exceptions for specific cases where the sampling methods helped such as when synthesizing with DataSynthesizer in correlated mode, using XGB classifier and SMOTE sampling method, where the F2 score is increased by relative 65.1%.

Table 2. Credit Card: supervised results, F2 score (ROS: random oversampling, RUS: random undersampling, SM: SMOTE sampling)

dataset	Real				SDV				DS-Corr.				SP				
sampl.	No	ROS	RUS	SM	No	ROS	RUS	SM	No	ROS	RUS	SM	No	ROS	RUS	SM	Avg
AB	82.0	65.5	12.3	59.0	77.6	59.9	17.3	74.7	52.0	46.8	17.4	16.8	72.6	37.1	10.2	29.2	45.7
GNB	23.2	21.1	16.0	22.8	45.5	1.2	4.5	0.1	63.7	43.9	44.7	42.6	21.0	19.0	13.3	18.6	25.1
KNN	76.4	79.4	24.8	72.2	7.5	18.5	3.5	3.1	50.2	18.8	5.9	10.2	71.4	56.3	21.1	50.8	35.6
LSVC	81.3	26.7	7.5	30.9	80.2	63.7	53.3	64.8	28.3	27.5	31.4	29.7	62.6	15.9	14.4	15.9	39.6
LR	68.2	24.2	14.6	42.3	79.5	62.2	53.5	64.1	40.4	24.7	28.4	27.6	57.7	14.1	12.7	22.6	39.8
RF	85.6	81.4	18.7	82.7	37.7	0.0	6.5	1.0	51.7	3.8	33.7	28.3	74.8	75.7	11.4	76.3	41.8
XGB	83.0	83.9	14.3	86.4	80.2	78.6	9.0	78.9	38.1	57.9	16.8	62.9	77.1	78.3	12.1	76.1	58.3
Avg	71.4	54.6	15.4	56.6	58.3	40.6	21.1	41.0	46.4	31.9	25.5	31.2	62.5	42.3	13.6	41.4	

Table 3. Credit Card: semi- & unsupervised results, F1 and F2 score

dataset	Real		SDV		DS-Corr.		DS-Ind.		SP		Avg	
method	F1	F2	F1	F2	F1	F2	F1	F2	F1	F2	F1	F2
AutoEncoder	30.4	43.2	24.4	37.9	37.2	48.6	46.3	55.2	30.6	44.1	33.8	45.8
GMM	23.4	34.1	21.2	34.4	27.6	42.5	43.1	52.4	26.5	36.6	28.4	40.0
Isol.Forest	7.4	16.2	8.0	17.3	15.3	29.9	34.3	31.3	6.7	15.0	14.3	22.0
LOF	0.8	1.8	4.7	10.8	17.5	26.9	20.5	37.1	1.5	3.2	9.0	16.0
1-ClassSVM	0.6	1.6	0.8	1.9	1.8	4.2	2.4	5.6	0.6	1.7	1.23	3.0
Avg	12.5	19.4	11.8	20.5	19.9	30.4	29.3	36.3	13.2	20.1		

Figure 4 and table 3 show the results for semi- and unsupervised settings. Mind again that for these technique, we can not apply sampling strategies, as there is no knowledge about the different types of samples (anomaly or not) for the unsupervised, and no information about relative sizes of the two classes for the semi-supervised case, as in this case, we only have samples from the "normal" class. Therefore, the corresponding tables and figures only show results without any sampling.

The first observation is that unsupervised and semi-supervised methods on average perform worse than the supervised counterparts, the major impact is on precision, where many methods struggle to obtain high values. However, this is expected, as this setting is more difficult, due to less information that can be exploited. In line with this, the semi-supervised methods (AutoEncoder and GMM) work better than the unsupervised ones.

Another noteworthy finding is that, in contrast to the supervised techniques, for unsupervised and semi-supervised techniques synthesizing the data increases the precision, recall and F2 score on average. This improvement in F2 score is 27.6% for semi-supervised and unsupervised approaches.

When comparing synthetic datasets, from Tables 3, 8 and 9 we can see that in semi-supervised and unsupervised setting the best performing synthesizer is DataSynthesizer in independent mode with an average F2 score of 36.3 %. This is especially intriguing when compared to the supervised techniques, where there was a significant loss of quality (not depicted) when the data was synthesized with DataSynthesizer in this mode. On the other hand, the synthetic dataset generated by SDV is the worst performing synthesizer for semi-supervised meth-

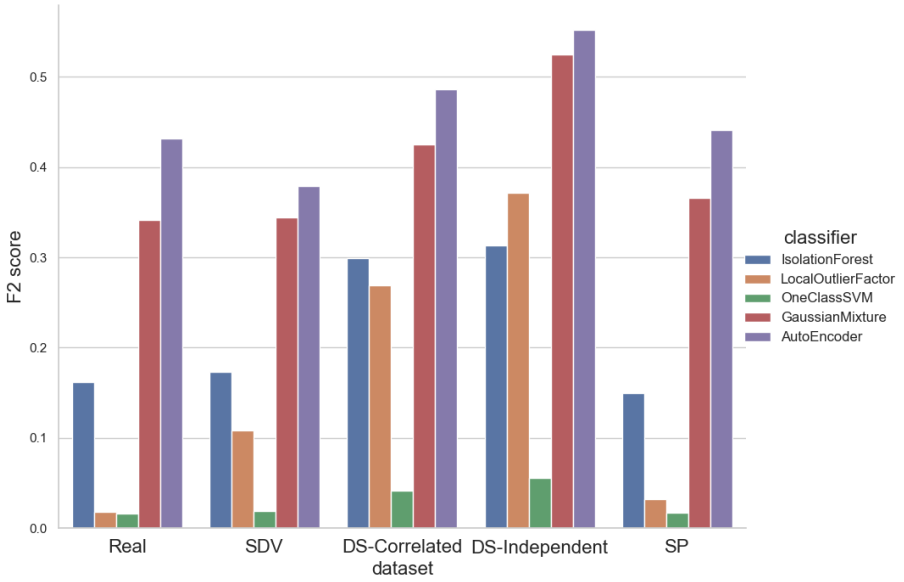

Fig. 4. Credit card: F2 scores for semi-supervised and unsupervised techniques

ods with an average F2 of 36.15%, whereas for unsupervised methods the worst performing synthesizer is Synthpop with an average F2 score of 6.3%, due to a very low precision of only 1.5%.

For the unsupervised methods, Isolation Forest obtains the best precision values, as well as a high recall value, resulting in the highest F2 score of 22%, whereas One Class SVM obtains the lowest F2 score of 3%, due to 0.6% precision. A significantly higher performance is achieved by semi-supervised methods. On average across the original and synthetic datasets, Auto Encoder outperforms Gaussian Mixture Model with an average precision of 23.8%, which is 4.3 higher than the Gaussian Mixture Model's precision and with an average recall of 61.4%, which is 4.5 higher than Gaussian Mixture Model's recall. Thus, also in regards to the F2 score, Auto Encoder outperforms the Gaussian Mixture Model, with an average F2 score of 45.8%, which is 5.8 higher than the Gaussian Mixture Model's average F2 score.

In Table 4 we show the results that are achieved in the literature using several different machine learning methods. If we compare these results with our results for the credit card dataset from Tables 2 and 13, we can observe that we achieved better results for Logistic Regression, Random Forest and XGB compared to [22], with 20.8, 26.8 and 21.6 higher precision. Mittal et al.[25] has the best precision for these three classifiers, achieving a precision of 99% for all three classifiers, which is 15.9, 4.7 and 5 higher than our precision for these three methods. On the other hand, Dornadula et al. [9] have a significantly better performance for Random Forest and Logistic Regression when using a

Table 4. Credit Card: benchmark algorithms and results (SM: SMOTE sampling, US: undersampling; '*' indicates scores we calculated from other scores)

Authors	Methods	Recall	Precision	Accuracy	F2-Score
Yann-Ael Le Borgne[1]	LR,RF, XGB	N\A	62.3, 67.8, 69.4	N\A	N\A
Mittal et al.[25]	NB,RF, KNN,LR, XGB,SVM, IF,LOF,K-Means	82,99,0,99, 99,93,100, 100,0	6,99,0, 99,99,0, 99,99,99	N\A	23.2*,99*,0*, 99*,99*,0*, 99.8*,99.8*, 0*
Dornadula et al.[9]	LOF,IF, SVM,LR,DT, RF,LOF-SM,IF-SM, LR-SM,DT-SM,RF-SM	N\A	0.38,1.47, 76.81,87.5, 88.54,93.10, 29.41,94.47, 98.31,98.14, 99.96	89.90,90.11, 99.87,99.90, 99.94,99.94, 45.82,58.83, 97.18,97.08, 99.98	N\A
Trivedi et al.[37]	RF,NB,LR, SVM,KNN, DT, GBM	95.12,91.98, 93.11,93,92, 91.99,93	95.98,91.20, 92.89,93.23, 94.59,90.99, 94	95,91.89, 90.45,93.96, 95,91,94	95.29*,91.8* 93.1*,93.05*, 92.5*,91.8*, 93.2*
Dhankhad et al.[8]	SC,RF, XGB,KNN, LR,GB, MLP,SVM, DT,NB	95,95,95,91, 94,94,93,93, 91,91	95,95,95,91, 94,94,93,93, 91,91	95.27,94.59, 94.59,94.25, 93.92,93.58, 93.24,93.24, 90.88,90.54	95*,95*,95*, 91*,94*,94*, 93*,93*,91*, 91*
Bachmann [2]	LR,KNN, SVM,SVC, LR-US, LR-SM	94,93,93,93, N\A,N\A	94,93,94,93, N\A,N\A	94,93,93,93, 94.21,98.70	94*,93*,93.2*, 93*,N\A, N\A
Mayer et al. [23]	NB,SVM, KNN,RF, LR,IF,LOF, 1CSVM, GMM,AE	76.8,70.5, 77.7,71.4, 76.8,76.8, 31.3,83, 71.4,55.4	6.6,88.8, 94.6,97.6, 86,5.4, 0.6,2.8, 87,19.3	N\A	24.5,73.6, 80.6,75.5, 78.5,21, 2.7,12.4, 74.1,40.3

[1] https://github.com/Fraud-Detection-Handbook/fraud-detection-handbook
[2] https://www.kaggle.com/janiobachmann/credit-fraud-dealing-with-imbalanced-datasets

sampling method to balance the dataset, with around 17 and 84 higher precision, respectively compared to our results. Overall, our results are however well in line with what is achievable in literature.

When comparing our unsupervised approaches to those in the literature, from Tables 4, 8 and 9 we can observe that our results for Isolation Forest and LOF slightly outperform the results from [9], achieving 2.43 and 0.02 better precision, respectively. On the other hand, for these two methods Mittal et al. [25] achieved an almost perfect results, with 100% recall and 99% precision for both methods, which is highly suspect due to the lack of similar results on any state-of-the-art paper; also, the paper is not clear on which data they evaluate on, i.e., whether they use an independent validation set, or not – hence, these results should be taken with a grain of salt

4.2 Annthyroid dataset

When comparing synthesizers with each other, we can observe from Tables 5, 14 and 15 that SDV achieves by far the lowest precision, recall and F2 scores. Data generated by Synthpop, on the other hand, achieves the best performance,

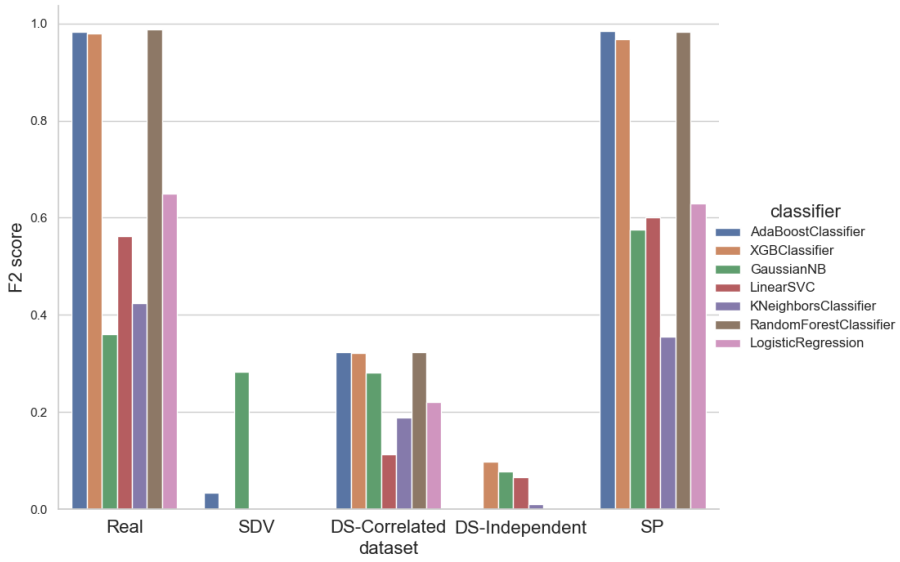

Fig. 5. Annthyroid: F2 scores for supervised techniques on datasets generated using no sampling

Table 5. Annthyroid: supervised results, F2 score (ROS: random oversampling, RUS: random undersampling, SM: SMOTE sampling)

dataset	Real				SDV				DS-Corr.				SP				
sampl.	No	ROS	RUS	SM	No	ROS	RUS	SM	No	ROS	RUS	SM	No	ROS	RUS	SM	Avg
AB	98.3	98.3	98.2	97.2	3.5	30.7	46.7	57.0	32.3	44.2	39.7	48.4	98.3	98.0	96.1	97.6	67.8
GNB	36.1	33.8	32.8	34.3	28.4	28.5	40.2	28.8	28.1	31.8	34.6	32.1	57.6	35.8	35.8	32.5	34.4
KNN	42.5	55.2	51.1	55.7	0.0	9.7	36.2	25.3	18.9	29.1	37.2	34.1	35.5	45.8	45.3	50.8	35.8
LSVC	56.2	80.3	64.9	68.2	0.0	41.2	54.4	45.6	11.4	55.7	52.5	53.8	60.0	88.5	80.8	74.6	55.5
LR	65.0	94.7	87.5	95.0	0.0	44.6	53.5	44.5	22.1	55.6	52.9	53.7	63.0	93.1	92.7	93.9	63.2
RF	98.7	98.5	97.3	98.5	0.0	0.0	62.2	23.5	32.4	36.2	44.4	47.8	98.3	98.3	95.9	99.1	64.4
XGB	97.9	98.5	97.1	98.5	0.0	5.8	55.5	23.6	32.1	34.9	40.3	46.9	96.7	97.4	96.7	99.1	63.8
Avg	70.7	79.9	75.5	78.2	4.5	22.9	49.8	35.4	25.3	41.1	43.1	45.3	72.8	79.6	77.6	78.2	

Table 6. Annthyroid: semi- & unsupervised results, F1 and F2 score

dataset	Real		SDV		DS-Corr.		DS-Ind.		SP		Avg	
method	F1	F2	F1	F2	F1	F2	F1	F2	F1	F2	F1	F2
AutoEncoder	23.1	37.7	18.6	33.5	18.0	31.0	18.6	34.5	23.0	35.3	20.3	34.4
GMM	13.8	28.6	13.8	28.6	13.8	28.6	13.8	28.6	13.8	28.6	13.8	28.6
Isol.Forest	10.9	9.3	16.2	18.5	6.5	5.3	5.4	4.3	10.2	8.4	9.8	9.1
LOF	18.0	17.8	16.8	22.1	7.5	6.9	11.1	8.6	21.0	20.7	14.9	15.2
1-ClassSVM	15.8	28.1	13.4	19.6	15.4	27.5	14.4	21.6	15.6	28.1	14.9	25.0
Avg	16.3	24.3	15.8	24.5	12.2	19.9	12.7	19.5	16.7	24.2		

Table 7. Algorithms used in literature and their results for annthyroid dataset ('*' indicates scores we calculated from other scores in the table)

Authors	Methods	Recall	Precision	Accuracy	F2-score
Salman et al. [33]	DT,SVM, RF,NB, LR,KNN	N\A	N\A	90.13,92.53, 91.2,90.67, 91.73,91.47	N\A
Sidiq et al.[35]	KNN,SVM, DT,NB	N\A	N\A	91.82,96.52, 98.89,91.57	N\A
Chandel et al. [4]	KNN, NB	N\A	N\A	93.44,22.56	N\A
Sinhya et al. [36]	NB, RF	N\A	N\A	95, 99.3,	N\A
Ionita et al. [17]	NB,DT, MLP	N\A	N\A	91.63,96.91, 95.15	N\A
Maysanjaya et al. [24]	RBF,LVQ, MLP,BPA, AIRS	95.3,93.5, 96.7,69.8, 93.5	95.3,94,96.8, 48.7,93.5	95.35,93.5, 96.74,69.77, 93.5	95.3*,93.6*, 96.72*, 65.23*,93.5*
Tyagi et al. [38]	KNN,SVM, DT	N\A	N\A	98.62,99.63, 75.76	N\A
Raisinghani et al.[30]	SVM,DT, LR,RF	96,99,97,99	96,99,97,99	96.25,99.46, 97.5,99.3	96*,99*,97*, 99*
Rehman et al. [1]	KNN,DT, NB,SVM, LR	90,67,100, 70,88	N\A	91.39,74.19, 100,80.46, 90.32	N\A

with achieving even better precision and F2 score than the original dataset. On average over all sampling methods, the dataset generated by Synthpop achieves the best F2 score of 77.05%, due to the highest precision and recall of 68.5% and 86.95% respectively.

Another interesting observation can be seen in the 'avg' columns, where we can see that Gaussian Naive Bayes, even though it achieves the best recall on average that ranges around 81.9%, has a very low precision of 13.8%, making it not suitable for our task. The other algorithms have a significantly higher precision, with Adaboost being the best with an average F2 score of 67.8%, due to good precision and recall of 68.7% and 74% respectively. Random Forest, XGB and Logistic Regression are the next best algorithms, and achieve an average F2 score of 64.4%, 63.8% and 63.2% respectively.

When comparing the sampling methods used to balance the datasets, we can see that, contrary to the credit card dataset's results, in almost all of the cases balancing helped to achieve better results. This is especially true for SDV and DataSynthesizer in correlated mode, though from a rather low base. When comparing the sampling methods between each other, we see that the best performing sampling method is Random Undersampling, followed by SMOTE with a slightly lower performance, whereas supervised methods performed worst on the datasets sampled with Random Oversampling.

Figure 5 depicts an average comparison of F2 scores for the different synthesizers and supervised methods when no sampling method was used

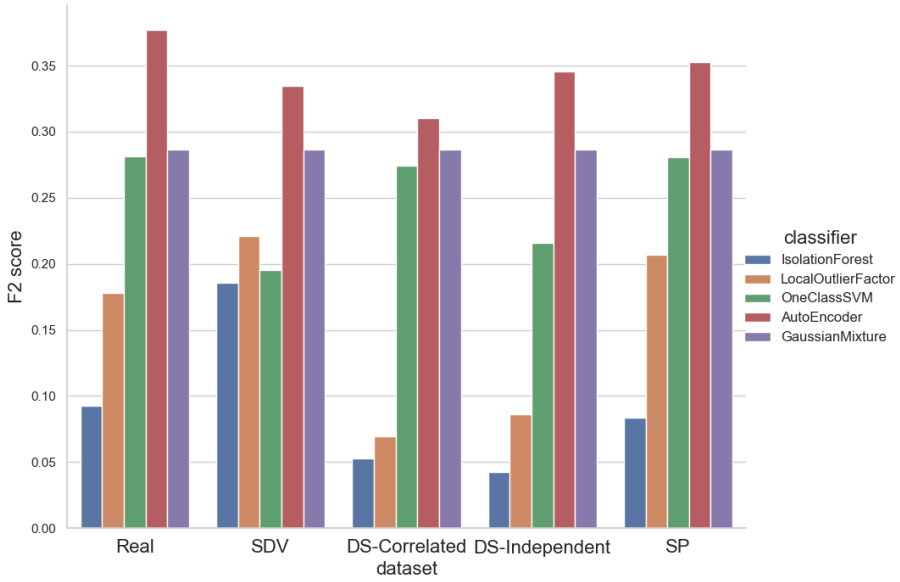

Fig. 6. Annthyroid: F2 scores for semi-supervised and unsupervised techniques

Table 7 shows the results reported in literature. When we compare these with our results in Table 15, we can see that for Random Forest we achieve slightly better recall compared to [30], but 1.8 lower precision. For Logistic Regression, Raisinghani et al. [30] and Rehman et al. [1] achieved better recall than we do, with 36.3 and 27.3 higher recall. Overall, our results are well embedded into the state-of-the-art results.

Tables 6, 10 and 11 show F2 scores, precisions and recalls for semi-supervised and unsupervised methods on original and synthetic datasets. Another visual representation of F2 scores is shown in Figure 6 for semi-supervised and unsupervised methods. As with the credit card dataset, and as expected, supervised approaches outperform unsupervised and semi-supervised ones. Moreover, similar to the supervised techniques, synthesizing the data decreases the F2 score on average by 9.5%, from 24.3% to 22%, this decrease on F2 score is due to precision and recall, with 11.7% and 5.8% decrease respectively. Thus, for the annthyroid dataset, synthesizing the dataset marginally decreases the performance of semi-supervised and unsupervised methods. An exception is the Gaussian Mixture Model, which achieves same F2 score, precision and recall, before and after synthesizing the dataset with any synthesizer that we have considered in our experiment.

Moreover, from Tables 6, 10 and 11 we can observe that datasets generated with Synthpop and Synthetic Data Vault outperform the datasets generated with DataSynthesizer in both correlated and independent modes. For semi-supervised methods dataset generated with Synthpop outperforms the other synthesizers,

with a precision of 11%, which is on average around 17.8% better than other synthesizers, and with a F2 score of 31.95%, which is 3.6% better than other synthesizers. On the other hand, semi-supervised methods perform worst on data generated by DataSynthesizer in correlated mode. On the other hand, when we compare the unsupervised methods to each other, data generated by Synthetic Data Vault outperforms other synthesizers, whereas data generated by DataSynthesizer in independent mode results in the worst F2 scores.

As expected, in all cases semi-supervised methods perform better than unsupervised ones. We note that Auto Encoder outperforms Gaussian Mixture Model in terms of F2 score and precision achieving on average 34.4% F2 score, and 12.0% precision over original and synthetic datasets. On the other hand GMM achieves the best recall compared to other semi-supervised and unsupervised methods, with a value of 100%.

As for unsupervised methods, even though One Class SVM shows a low precision of only 9%, having the highest recall of 47.3% compared to other unsupervised methods makes it the best performing method with a F2 score of 25%, which is around 10 and 16 percentage points higher than Local Outlier Factors and Isolation Forest, respectively. Isolation Forest with an F2 score of only 9.1% is not well suited for this task.

4.3 Observations

Based on the results shown in the Sections 4.1 and 4.2 we can conclude that the quality of anomaly detection in both credit card and annthyroid datasets for supervised methods is slightly reduced when using synthetic data, but they achieve results that are very competitive to the real data. As expected, unsupervised and semi-supervised approaches perform worse than supervised ones, which is expected, considering that having an actual label for the data provides more exploitable information than the other approaches have available.

In both datasets, XGB performed very well in the supervised techniques, while Gaussian Naive Bayes yields the worst results. When looking at semi-supervised methods for both datasets AutoEncoder outperforms Gaussian Mixture Model. Furthermore, the best synthesizer for the credit card dataset and annthyroid dataset in almost all of the cases is Synthpop. SDV performed well on the credit card dataset, but not so well on annthyroid. DataSynthesizer was overall the worst of the synthesizers.

One of the most noteworthy distinctions is the effect of sampling methods on the results in the two cases for supervised methods. For the annthyroid dataset, sampling significantly improved the performance of supervised methods in most cases. On the other hand, in most of the cases it has bad effect on the quality of anomaly detection in credit card dataset. We see that both effects also transferred to synthetic data, that is, for annthyroid, also synthetic data profits from choosing a fitting sampling methods.

When looking at non-sampled datasets for supervised techniques, another significant difference is the influence synthetization has on the data for the two datasets. After synthesizing the credit card dataset using the Synthetic Data

Vault, the performance dropped only marginally, whereas the same synthesizer on the annthyroid dataset incurs a substantial performance drop. This indicates that the type of data and its distribution has a significant impact on the synthetisation success. An interesting observation is how unsupervised methods performed differently on annthyorid and credit card datasets when the datasets are synthesized with DataSynthesizer in the independent attribute mode. Whereas on the annthyroid dataset this mode shows the worst results among the other synthetic datasets, it achieves the best F2 score of all synthesisers on the credit card dataset. This might be caused by the nature of the features in this dataset, which are obtained via PCA, a dimensionality reduction technique.

5 Conclusion

Data privacy tries to balance non-disclosing management of sensitive data, while also preserving data utility. Synthetic data generation is one method that recently gained a lot of attention, and has shown to exhibit high utility for several tasks. In this paper, we used multiple metrics to to assess the application fidelity of existing synthetic data creation strategies on datasets from the financial and health domain. We used state-of-the-art synthetic data generators for synthesizing the dataset; due to highly unbalanced datasets we applied several sampling approaches. Moreover, supervised, unsupervised and semi-supervised anomaly detection methods were used.

The results reveal that Synthpop overall outperforms the other synthetic data generators. As a guideline, it is thus a good overall choice to employ. In a few cases, especially on some of the supervised classifiers on the credit card data, the SDV outperformed Synthpop, though. Further, in the semi- and unsupervised settings on the same dataset, the DataSynthesizer yielded substantially better results. These might both correlate with the specific nature of most of the variables in this dataset. Thus, the best choice without any prior knowledge still remains Synthpop; however, if possible, a wider range of synthesizers should be tested before deciding on a specific approach, to account for particular attributes, distributions and correlations within a dataset.

Moreover, we have seen that XGB, Adaboost, and RandomForest outperform the other supervised approaches in our datasets, whereas Isolation Forest outperformed the other unsupervised techniques we used. For semi-supervised settings, the AutoEncoder was the best choice, and should be utilised first. While these two are thus recommended as guidelines for the semi- and unsupervised settings, for the supervised methods, we can generally recommend ensemble-based techniques as the ones mention above, of which in particular XGB was most successful in our evaluation.

Future work will focus on generalising our results beyond our current evaluation, in particular by addressing further datasets. Also other synthetisation methods, e.g. based on Generative Adversarial Networks (GANs), will be considered.

Acknowledgements This work was partially funded by the Austrian Research Promotion Agency FFG under grants 877173 (GASTRIC) and 871267 (WellFort). SBA Research (SBA-K1) is a COMET center within the COMET – Competence Centers for Excellent Technologies program, funded by BMK, BMDW, and the federal state of Vienna. The COMET program is managed by FFG.

References

1. Abbad Ur Rehman, H., Lin, C.Y., Mushtaq, Z., Su, S.F.: Performance Analysis of Machine Learning Algorithms for Thyroid Disease. Arabian Journal for Science and Engineering **46**(10) (Oct 2021). https://doi.org/10.1007/s13369-020-05206-x
2. Acs, G., Melis, L., Castelluccia, C., De Cristofaro, E.: Differentially Private Mixture of Generative Neural Networks. IEEE Transactions on Knowledge and Data Engineering **31**(6) (Jun 2019). https://doi.org/10.1109/TKDE.2018.2855136
3. Brickell, J., Shmatikov, V.: The cost of privacy: destruction of data-mining utility in anonymized data publishing. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD, ACM Press, Las Vegas, Nevada, USA (2008). https://doi.org/10.1145/1401890.1401904
4. Chandel, K., Kunwar, V., Sabitha, S., Choudhury, T., Mukherjee, S.: A comparative study on thyroid disease detection using K-nearest neighbor and Naive Bayes classification techniques. CSI Transactions on ICT **4**(2-4) (Dec 2016). https://doi.org/10.1007/s40012-016-0100-5
5. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM Computing Surveys **41**(3) (Jul 2009). https://doi.org/10.1145/1541880.1541882
6. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research **16** (Jun 2002). https://doi.org/10.1613/jair.953
7. Dankar, F.K., Ibrahim, M.K., Ismail, L.: A Multi-Dimensional Evaluation of Synthetic Data Generators. IEEE Access **10** (2022). https://doi.org/10.1109/ACCESS.2022.3144765
8. Dhankhad, S., Mohammed, E., Far, B.: Supervised Machine Learning Algorithms for Credit Card Fraudulent Transaction Detection: A Comparative Study. In: IEEE International Conference on Information Reuse and Integration. IRI, IEEE, Salt Lake City, UT (Jul 2018). https://doi.org/10.1109/IRI.2018.00025
9. Dornadula, V.N., Geetha, S.: Credit Card Fraud Detection using Machine Learning Algorithms. Procedia Computer Science **165** (2019). https://doi.org/10.1016/j.procs.2020.01.057
10. Goix, N.: How to Evaluate the Quality of Unsupervised Anomaly Detection Algorithms? In: ICML Anomaly Detection Workshop. New York, NY, USA, (Jul 2016)
11. Goldstein, M., Uchida, S.: A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data. PLOS ONE **11**(4) (Apr 2016). https://doi.org/10.1371/journal.pone.0152173
12. Hittmeir, M., Ekelhart, A., Mayer, R.: On the Utility of Synthetic Data: An Empirical Evaluation on Machine Learning Tasks. In: International Conference on Availability, Reliability and Security. ARES, ACM, Canterbury CA United Kingdom (Aug 2019). https://doi.org/10.1145/3339252.3339281
13. Hittmeir, M., Ekelhart, A., Mayer, R.: Utility and Privacy Assessments of Synthetic Data for Regression Tasks. In: 2019 IEEE International Conference on Big Data

(Big Data). IEEE, Los Angeles, CA, USA (Dec 2019). https://doi.org/10.1109/BigData47090.2019.9005476
14. Hittmeir, M., Mayer, R., Ekelhart, A.: Utility and Privacy Assessment of Synthetic Microbiome Data. In: Data and Applications Security and Privacy XXXVI. DBSec, Springer International Publishing, Cham (Jul 2022)
15. Hodge, V., Austin, J.: A Survey of Outlier Detection Methodologies. Artificial Intelligence Review **22**(2) (Oct 2004). https://doi.org/10.1023/B:AIRE.0000045502.10941.a9
16. Ibidunmoye, O., Hernández-Rodriguez, F., Elmroth, E.: Performance Anomaly Detection and Bottleneck Identification. ACM Computing Surveys **48**(1) (Sep 2015). https://doi.org/10.1145/2791120
17. Ioniţă, I., Ioniţă, L.: Prediction of Thyroid Disease Using Data Mining Techniques. BRAIN. Broad Research in Artificial Intelligence and Neuroscience **7**(3) (Aug 2016)
18. Jansson, D., Medvedev, A., Axelson, H., Nyholm, D.: Stochastic anomaly detection in eye-tracking data for quantification of motor symptoms in Parkinson's disease. In: International Symposium on Computational Models for Life Sciences. Sydney, Australia (2013). https://doi.org/10.1063/1.4825001
19. Kim, G., Lee, S., Kim, S.: A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. Expert Systems with Applications **41**(4) (Mar 2014). https://doi.org/10.1016/j.eswa.2013.08.066
20. Kong, J., Kowalczyk, W., Menzel, S., Bäck, T.: Improving Imbalanced Classification by Anomaly Detection. In: International Conference on Parallel Problem Solving from Nature. PPSN, vol. 12269. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-58112-1_35
21. Lazarevic, A., Kumar, V.: Feature bagging for outlier detection. In: ACM SIGKDD International Conference on Knowledge Discovery in Data Mining. KDD, ACM Press, Chicago, Illinois, USA (2005). https://doi.org/10.1145/1081870.1081891
22. Le Borgne, Y.A., Siblini, W., Lebichot, B., Bontempi, G.: Reproducible Machine Learning for Credit Card Fraud Detection - Practical Handbook. Université Libre de Bruxelles (2022), https://github.com/Fraud-Detection-Handbook/fraud-detection-handbook
23. Mayer, R., Hittmeir, M., Ekelhart, A.: Privacy-Preserving Anomaly Detection Using Synthetic Data. In: Data and Applications Security and Privacy XXXIV. DBSec, Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-49669-2_11
24. Maysanjaya, I.M.D., Nugroho, H.A., Setiawan, N.A.: A comparison of classification methods on diagnosis of thyroid diseases. In: International Seminar on Intelligent Technology and Its Applications. ISITIA, IEEE, Surabaya (May 2015). https://doi.org/10.1109/ISITIA.2015.7219959
25. Mittal, S., Tyagi, S.: Performance Evaluation of Machine Learning Algorithms for Credit Card Fraud Detection. In: International Conference on Cloud Computing, Data Science & Engineering. Confluence, IEEE, Noida, India (Jan 2019). https://doi.org/10.1109/CONFLUENCE.2019.8776925
26. Nowok, B., Raab, G.M., Dibben, C.: synthpop: Bespoke Creation of Synthetic Data in R. Journal of Statistical Software **74**(11) (Oct 2016). https://doi.org/10.18637/jss.v074.i11
27. Patki, N., Wedge, R., Veeramachaneni, K.: The Synthetic Data Vault. In: IEEE International Conference on Data Science and Advanced Analytics. DSAA, IEEE, Montreal, QC, Canada (Oct 2016). https://doi.org/10.1109/DSAA.2016.49

28. Ping, H., Stoyanovich, J., Howe, B.: DataSynthesizer: Privacy-Preserving Synthetic Datasets. In: International Conference on Scientific and Statistical Database Management. SSDBM, ACM, Chicago IL USA (Jun 2017). https://doi.org/10.1145/3085504.3091117
29. Purarjomandlangrudi, A., Ghapanchi, A.H., Esmalifalak, M.: A data mining approach for fault diagnosis: An application of anomaly detection algorithm. Measurement **55** (Sep 2014). https://doi.org/10.1016/j.measurement.2014.05.029
30. Raisinghani, S., Shamdasani, R., Motwani, M., Bahreja, A., Raghavan Nair Lalitha, P.: Thyroid Prediction Using Machine Learning Techniques. In: International Conference on Advances in Computing and Data Sciences. Springer Singapore, Singapore (2019). https://doi.org/10.1007/978-981-13-9939-8_13
31. Rankin, D., Black, M., Bond, R., Wallace, J., Mulvenna, M., Epelde, G.: Reliability of Supervised Machine Learning Using Synthetic Data in Health Care: Model to Preserve Privacy for Data Sharing. JMIR Medical Informatics **8**(7) (Jul 2020). https://doi.org/10.2196/18910
32. Rubin, D., Reiter, J., Rubin, D.: Statistical Disclosure Limitation. Journal of Official Statistics **9**(2) (1993)
33. salman, K., Sonuç, E.: Thyroid Disease Classification Using Machine Learning Algorithms. Journal of Physics: Conference Series **1963**(1) (Jul 2021). https://doi.org/10.1088/1742-6596/1963/1/012140
34. Samarati, P.: Protecting respondents identities in microdata release. IEEE Transactions on Knowledge and Data Engineering **13**(6) (Dec 2001). https://doi.org/10.1109/69.971193
35. Sidiq, U., Mutahar Aaqib, S., Khan, R.A.: Diagnosis of Various Thyroid Ailments using Data Mining Classification Techniques. International Journal of Scientific Research in Computer Science, Engineering and Information Technology (Jan 2019). https://doi.org/10.32628/CSEIT195119
36. Sindhya, K.: Effective Prediction of Hypothyroid using various data mining techniques. International Journal of Research and Development **5**(2), 311–317 (Feb 2020)
37. Trivedi, N.K., Simaiya, S., Lilhore, U.K., Sharma, S.K.: An Efficient Credit Card Fraud Detection Model Based on Machine Learning Methods. International Journal of Advanced Science and Technology **29**(5) (Apr 2020)
38. Tyagi, A., Mehra, R., Saxena, A.: Interactive Thyroid Disease Prediction System Using Machine Learning Technique. In: International Conference on Parallel, Distributed and Grid Computing. PDGC, IEEE, Solan Himachal Pradesh, India (Dec 2018). https://doi.org/10.1109/PDGC.2018.8745910
39. Zhang, W., He, X.: An Anomaly Detection Method for Medicare Fraud Detection. In: IEEE International Conference on Big Knowledge. ICBK, IEEE, Hefei, China (Aug 2017). https://doi.org/10.1109/ICBK.2017.47

A Appendix: Additional Results

A.1 Credit Card dataset results

Tables 8 and 9 show the precision respectively recall for semi-supervised and unsupervised techniques on original and synthetic datasets, whereas Tables 12 and 13 show the precision respectively recall for supervised methods.

A.2 Annthyroid dataset results

Tables 10 and 11 show the precision respectively recall for semi-supervised and unsupervised techniques on original and synthetic datasets, whereas Tables 14 and 15 show the precision respectively recall for supervised methods.

Table 8. Credit Card: semi- & unsupervised results, precision

dataset method	Real	SDV	DS-Corr.	DS-Ind.	SP	Avg
AutoEncoder	20.3	15.3	26.7	36.5	20.3	23.8
GMM	15.4	12.9	17.4	33.3	18.2	19.5
Isol.Forest	3.9	4.2	8.4	40.8	3.5	12.2
LOF	0.4	2.4	11.1	11.7	0.8	5.3
1-ClassSVM	0.3	0.4	0.9	1.2	0.3	0.6
Avg	8.1	7.0	12.9	24.7	8.6	

Table 9. Credit Card: semi- & unsupervised results, recall

dataset method	Real	SDV	DS-Corr.	DS-Ind.	SP	Avg
AutoEncoder	60.2	60.2	61.2	63.3	62.2	61.4
GMM	49.0	59.2	66.3	61.2	49.0	56.9
Isol.Forest	81.6	79.6	83.7	29.6	84.7	71.8
LOF	14.3	85.7	41.8	81.6	13.3	47.3
1-ClassSVM	96.9	95.9	93.9	91.8	96.9	95.1
Avg	60.4	76.1	69.4	65.5	61.2	

Table 10. Annthyroid: semi- & unsupervised results, precision

dataset method	Real	SDV	DS-Corr.	DS-Ind.	SP	Avg
AutoEncoder	14.0	10.7	10.6	10.5	14.5	12.0
GMM	7.4	7.4	7.4	7.4	7.4	7.4
Isol.Forest	15.5	13.3	10.4	10.3	16.0	13.1
LOF	18.1	12.0	9.0	21.6	21.4	16.4
1-ClassSVM	9.1	8.8	8.9	9.2	9.0	9.0
Avg	12.8	10.4	9.3	11.8	13.7	

Table 11. Annthyroid: semi- & unsupervised results, recall

dataset method	Real	SDV	DS-Corr.	DS-Ind.	SP	Avg
AutoEncoder	65.4	72.0	59.8	80.4	55.1	66.5
GMM	100	100	100	100	100	100
Isol.Forest	8.4	20.6	4.7	3.7	7.5	9.0
LOF	17.8	28.0	6.5	7.5	20.6	16.1
1-ClassSVM	58.9	28.0	57.0	32.7	59.8	47.3
Avg	50.1	49.7	45.6	44.9	48.6	

Table 12. Credit Card: supervised results, precision (ROS: random oversampling, RUS: random undersampling, SM: SMOTE sampling)

dataset	Real				SDV				DS-Corr.				SP				
sampl.	No	ROS	RUS	SM	No	ROS	RUS	SM	No	ROS	RUS	SM	No	ROS	RUS	SM	Avg
AB	87.8	32.6	2.8	25.9	67.5	29.6	4.1	57.7	52.0	17.1	4.1	4.2	69.2	11.0	2.2	7.9	29.7
GNB	5.9	5.2	3.7	5.8	18.5	0.2	0.9	0.0	32.0	14.8	15.2	14.7	5.2	4.6	3.1	4.5	8.4
KNN	84.9	68.6	6.4	46.6	75.0	10.0	0.7	0.7	80.4	28.8	1.3	2.6	75.8	40.7	5.3	21.7	34.3
LSVC	84.0	7.0	1.6	8.5	74.8	33.9	22.3	35.6	75.0	7.3	8.8	8.1	81.7	3.7	3.3	3.7	28.7
LR	83.1	6.1	3.3	13.6	72.1	31.9	22.5	34.5	56.1	6.3	7.6	7.3	71.1	3.2	2.9	5.7	26.7
RF	94.3	95.1	4.5	82.7	71.7	0.0	1.4	0.9	66.7	100.0	9.6	75.0	80.9	81.1	2.5	60.6	51.7
XGB	94.0	89.0	3.3	85.0	74.8	78.6	2.0	76.5	63.0	66.3	4.0	61.4	84.1	77.0	2.7	57.9	57.5
Avg	76.3	43.4	3.7	38.3	64.9	26.3	7.7	29.4	60.7	34.4	7.2	24.8	66.9	31.6	3.1	23.1	

Table 13. Credit Card: supervised results, recall (ROS: random oversampling, RUS: random undersampling, SM: SMOTE sampling)

dataset	Real				SDV				DS-Corr.				SP				
sampl.	No	ROS	RUS	SM	No	ROS	RUS	SM	No	ROS	RUS	SM	No	ROS	RUS	SM	Avg
AB	80.6	87.8	90.8	86.7	80.6	80.6	86.7	80.6	52.0	82.7	87.8	68.4	73.5	91.8	90.8	87.8	81.8
GNB	84.7	87.8	87.8	87.8	71.4	90.8	82.7	4.1	84.7	86.7	86.7	80.6	84.7	87.8	84.7	87.8	80.0
KNN	74.5	82.7	86.7	83.7	6.1	23.5	84.7	31.6	45.9	17.3	66.3	39.8	70.4	62.2	85.7	76.5	58.6
LSVC	80.6	91.8	94.9	90.8	81.6	81.6	81.6	81.6	24.5	89.8	87.8	89.8	59.2	90.8	92.9	91.8	82.0
LR	65.3	91.8	91.8	89.8	81.6	81.6	81.6	81.6	37.8	89.8	88.8	89.8	55.1	92.9	92.9	89.8	81.4
RF	83.7	78.6	90.8	82.7	33.7	0.0	86.7	1.0	49.0	3.1	89.8	24.5	73.5	74.5	91.8	81.6	59.1
XGB	80.6	82.7	91.8	86.7	81.6	78.6	85.7	79.6	34.7	56.1	88.8	63.3	75.5	78.6	92.9	82.7	77.5
Avg	78.6	86.2	90.7	86.9	62.4	62.4	84.3	51.5	46.9	60.8	85.1	65.2	70.3	82.7	90.2	85.4	

Table 14. Annthyroid: supervised results, precision(ROS: random oversampling, RUS: random undersampling, SM: SMOTE sampling)

dataset	Real				SDV				DS-Corr.				SP				
sampl.	No	ROS	RUS	SM	No	ROS	RUS	SM	No	ROS	RUS	SM	No	ROS	RUS	SM	Avg
AB	95.5	95.5	91.5	93.8	100.0	64.4	18.0	37.6	83.3	22.1	14.8	27.5	92.2	90.7	82.9	89.2	68.7
GNB	10.4	9.4	9.1	9.6	7.5	7.6	37.0	7.7	25.0	10.3	10.6	9.2	37.8	10.2	10.2	9.0	13.8
KNN	93.0	35.0	21.9	28.7	0.0	26.5	12.0	18.1	77.3	17.9	14.2	15.6	89.2	32.0	20.0	29.0	33.1
LSVC	90.2	45.7	38.4	31.9	0.0	16.4	24.1	21.8	90.9	37.4	27.9	35.7	83.3	73.0	61.0	47.7	45.3
LR	90.3	78.1	67.8	79.3	0.0	21.6	25.1	21.5	83.3	35.8	28.6	35.3	87.5	77.2	73.6	82.5	55.5
RF	97.2	96.4	87.7	96.4	0.0	0.0	27.6	53.7	85.7	81.0	18.0	48.6	95.5	95.5	82.3	95.5	66.3
XGB	97.2	96.4	87.0	96.4	0.0	83.3	23.2	38.3	76.9	73.3	15.3	41.3	94.5	94.6	85.6	95.5	68.7
Avg	82.0	65.2	57.6	62.3	15.4	31.4	23.8	28.4	74.6	39.7	18.5	30.4	82.9	67.6	59.4	64.1	

Table 15. Annthyroid: supervised results, recall (ROS: random oversampling, RUS: random undersampling, SM: SMOTE sampling)

dataset	Real				SDV				DS-Corr.				SP				
sampl.	No	ROS	RUS	SM	No	ROS	RUS	SM	No	ROS	RUS	SM	No	ROS	RUS	SM	Avg
AB	99.1	99.1	100.0	98.1	2.8	27.1	77.6	65.4	28.0	58.9	68.2	59.8	100.0	100.0	100.0	100.0	74.0
GNB	94.4	96.3	94.4	96.3	91.6	91.6	41.1	91.6	29.0	66.4	80.4	85.0	66.4	96.3	95.3	94.4	81.9
KNN	37.4	64.5	76.6	72.9	0.0	8.4	72.9	28.0	15.9	34.6	62.6	48.6	30.8	51.4	66.4	62.6	45.9
LSVC	51.4	99.1	78.5	95.3	0.0	66.4	79.4	62.6	9.3	63.6	67.3	61.7	56.1	93.5	87.9	86.9	66.2
LR	60.7	100.0	94.4	100.0	0.0	60.7	74.8	60.7	18.7	64.5	67.3	61.7	58.9	98.1	99.1	97.2	69.8
RF	99.1	99.1	100.0	99.1	0.0	0.0	90.7	20.6	28.0	31.8	70.1	47.7	99.1	99.1	100.0	100.0	67.8
XGB	98.1	99.1	100.0	99.1	0.0	4.7	85.0	21.5	28.0	30.8	68.2	48.6	97.2	98.1	100.0	100.0	67.4
Avg	77.2	93.9	92.0	94.4	13.5	37.0	74.5	50.1	22.4	50.1	69.2	59.0	72.6	90.9	92.7	91.6	