UniformGAN: GAN in uniform probability spaces.

Marc Visser
Supervisor(s): Lydia Y. Chen, Zilong Zhao
EEMCS, Delft University of Technology, The Netherlands

June 19, 2022

A Dissertation Submitted to EEMCS faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering

UniformGAN: generative adversarial networks in uniform probability spaces.

Marc Visser

ABSTRACT

Sharing data is becoming increasingly difficult, due to the regulatory constraints imposed by the General Data Protection Regulation (GDPR). Businesses are not allowed to share data which contains privacy sensitive information. Synthetic data generation has emerged as a solution to this problem. State of the art generative adversarial networks (GAN) can generate synthetic data which statistically resembles the original data, while changing privacy sensitive information so that it cannot be related back to a person.

However, the process of generating synthetic data is still a very time consuming process for data scientists.

One of the challenges faced in synthetic data generation is aptly modeling the raw data; transforming it into numerical, and specifying the hyper-parameters such as which columns are categorical, mixed type, numerical or log distributed, is a non-trivial task. Another challenge is making estimations about the underlying distributions of the data and how these different distributions are correlated.

The proposed solution UniformGAN addresses these issues by adopting a transformer which can handle raw data and detect the data type and transforms it into a numerical equivalent. It uses the data type and estimated distribution to set the hyper-parameters for categorical columns, mixed columns, and log columns. Furthermore, it estimates the underlying distributions of the data and leverages a statistical transformation in order for the machine learning model to easier learn the dependence structure of variables.

The model proposed in this article extends the novel CTAB-GAN [21] model to add the flexibility of the probability integral transform idea from copulaGAN [6].

CTAB-GAN leverages a mixed-type encoder, training by sampling and treats long tails. copulaGAN makes use of a numerical encoder and uses a probabilistic transformation to make capture the dependence structure of the variables without any affect on the margins. UniformGAN aims to combine both these methods in order to remove the time-consuming hyper-parameter tuning of conditional tabular GAN and simultaneously improve the training time without sacrificing synthesizing quality.

The model works by transforming each non-categorical variable using a probabilistic transformer. The transformer applies the probability integral transform (cdf), to construct an equivalent set of variables, and a series of Kolmogorov–Smirnov tests is made of whether a uniform distribution is appropriate for the constructed data-set. CTAB-GAN is fitted with the transformed data and then sampled. On this sample, a reverse cdf is performed, inverting the cdf of the distribution that corresponds to each variable.

The evaluation with regard to machine learning utility, statistical similarity, and privacy preverabiliy has shown that UniformGAN improves accuracy with regard to decision tree classification utility, improving averaged machine learning utility by 2% compared to CTAB-GAN, and 19.21% compared to copulaGAN, while maintaining statistical similarity and privacy preservability compared to state of the art tabular data modeling techniques.

CCS CONCEPTS

• **Mathematics of computing** → **Distribution functions**; • **Computing methodologies** → **Neural networks**; • **Security and privacy** → *Privacy-preserving protocols.*

KEYWORDS

neural networks, GAN, distribution functions

1 INTRODUCTION

Many companies store an increasingly amount of data from various data sources. When the data is aggregated, data-scientists often try to enrich the datasets by merging various data in order to increase the amount of features, and thus information and insight available. However, the big knowledge behind big data often impedes personal privacy and leads to unjustified analysis[12].

State of the art generative adversarial networks (GAN), particularly conditional GAN, can generate synthetic data which statistically resembles the original data, while changing privacy sensitive information so that it cannot be related back to a person.

One of the challenges faced in synthetic data generation is aptly modeling the raw data; transforming it into numerical, and specifying the hyper-parameters such as which columns are categorical, mixed type, numerical or log distributed is a non-trivial task. Another difficult task is making estimations about the underlying distributions of the data and how these different distributions are correlated.

The **aim** of this research is to examine state-of-the-art data synthesising techniques in the statistical and machine learning realm, and further improve on conditional GAN tabular data generation. Specifically, the aim is to remove the time-consuming hyper-parameter tuning of conditional tabular GAN and improve the training time without sacrificing its synthesizing quality.

UniformGAN is based on CTAB-GAN[21] and copulaGAN [6]. Both copulaGAN and CTAB-GAN are based on CTGAN [19]; which efficiently treats minority classes, and adds classification, information and generator loss. CTAB-GAN introduces a mixed-type encoder to better represent mixed categorical-continuous variables as well as missing values. Additionally CTAB-GAN leverages a log-frequency sampler to overcome the mode collapse problem for imbalanced variables[21].

State-of-the-art GAN such as CTGAN[19], CTAB-GAN+ [20] suffer from an exponential increase in computational efforts when the dimensionality is increased. By transforming the data into uniform probability space, the model will try to learn the dependence structure of the copula without any affect on the marginals. This is why UniformGAN is proposed. To validate whether or not this

claim is true three questions are posed.
1) *Can statistical tabular data generators synthesize data with the same accuracy and quality as conditional tabular GAN?*
2) *Can we generate more accurate data with by leveraging the integral probability transform in UniformGAN?*
3) *Will the integral probability transform make it easier for the GAN to learn its dependence structure?*

The proposed solution takes the CTAB-GAN model and adds an additional probabilistic transformer. After the transformation made by the transformer all variables have uniform marginals, which means that the GAN will try to learn the copula function, this function describes the datasets' dependence structure of variables.

The aim is achieved by combining state of the art techniques in CTAB-GAN with the **integral probability transform**, by transforming the numerical variables into marginal univariates, we can use Sklar's theorem 2.2. The idea is that if we have random variables; say X and Y with uniform univariate margins $F_x(x)$ and $G_y(y)$ then there exists a copula function C such that $H(x, y) = C(F_x(x), G_y(y))$. The copula function allows to model the margins separately from the dependence structure. This is why we transform all variables to be uniform in order to capture the pure dependence structure between variables without any affect of the margins. This allows the machine learning model to easier learn the correlation. Furthermore, $u = F(x)$ and $F^{-1}(u) = x$. Which means we can easily transform back to the original data. This transformation into uniform probability space is the basis for UniformGAN.

In order to answer (1) we will examine utility metrics, making a comparison with the original data, and the transformed and inverted data.

Then in order to answer (2) UniformGAN is evaluated with regard to: utility of machine learning, statistical similarity to the real data, and privacy preservability. Specifically, the proposed UniformGAN is tested on five widely used machine learning datasets: Adult, Covertype, Intrusion and Loan. UniformGAN is tested against four state of the art tabular data generators: Copulas, CopulaGAN, CT-GAN and CTAB-GAN.

To conclude, we will answer (3) by analysing model performance across epochs giving us a insight into how many iteration are needed until we get reasonable results.

2 BACKGOUND AND RELATED STUDIES

The Copula and GAN are two building blocks for this thesis. We first describe the background of them and how the prior art builds on top of them.

2.1 conditional GAN

GAN are a popular method to generate synthetic data first applied with great success to images and later adapted to tabular data [1]. GAN leverage an adversarial game between a generator trying to synthesize realistic data and a discriminator trying to discern synthetic from real samples. Conditional GAN are trained via a zero-sum minimax game where the discriminator tries to maximize the objective, while the generator tries to minimize it. Furthermore, a conditional vector is introduced to leverage conditional sampling.

2.1.1 Copulas. Copulas are functions that enable us to separate the marginal distributions from the dependency structure of a given multivariate distribution [5].

Theorem 2.1. *(Copula). A copula is a multivariate distribution with CDF $C : [0, 1]^D \to [0, 1]$ that has standard uniform marginals, i.e. the marginals C_j of C satisfy $C_j \sim U[0, 1]$*

Theorem 2.2. *(Sklar's Theorem 1959). Consider a d-dimensional CDF, F with marginals $F_1, ..., F_d$ Then there exists a copula C, such that*

$$F(x_1, ..., x_d) = C(F_1(x_1), ..., F_d(x_d))$$

for all $x_i \in [-\infty, \infty]$ and $i = 1, ..., d$

The fundamental idea behind copula theory is that we can associate every multivariate distribution with a uniquely defined copula C.

2.2 GAN-based generator

There have been several studies that extend GAN to integrate categorical variables by changing the GAN Architecture.

One of the first instances is MedGAN [3]. MedGAN combines autoencoders and GAN. MedGAN is able to generate continuous and discrete variables and has been applied in order to facilitate Electronic Health Record data generation.

CrGAN-Cnet [11] integrates Cramer Distance [2] and a Cross-Net architecture [18] in order to generate Airline Passenger Name records. CrGAN-Cnet is also able to handle missing values by adding new variable for missing records.

TableGAN [14] introduces an auxiliary classifier and information loss the GAN. Specifically, it utilises Convolutional Neural Networks (CNN) for the generator, discriminator and classifier.

2.3 Conditional GAN-based generator

Due to the limitation of of controlling what data class is generated by the GAN, conditional GAN have emerged as a solution. In a conditional GAN a conditional vector can be used to generate data of a specific class. This is important when data is limited and/or highly skewed. By being able to generate data from a specific class we can re-balance the distribution as to reduce the discrepancy between the real distribution and synthesized distribution.

CWGAN [4] builds on the conditional GAN framwork by implementing wasserstein distance [1]. It uses the conditional vector to oversample minority classes.

CTGAN [19] integrates pacGAN [9] structure in its discriminator. Furthemore it implements Generator loss and WGAN loss and gradient penalty to train a conditional GAN.

CTAB-GAN [21] addresses the problem of mixed type variables, most existing GANs only treat variables as categorical or continuous. Additionally it log-transforms long tail variables in order to better capture the long tail. Furthermore training-by-sampling [19] is utilised, but extended to include the modes of continuous and mixed columns, in order counter imbalanced training data-sets.

An overview of the architecture of CT-GAN can be seen in figure 1

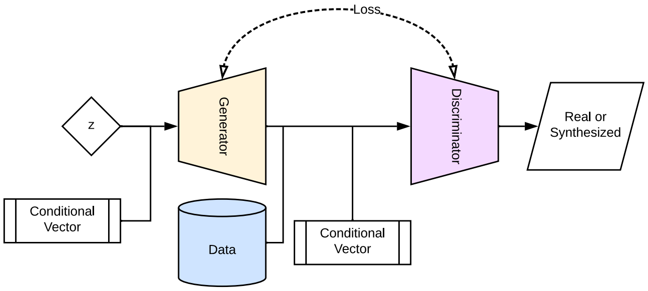

Figure 1: Architecture of CTGAN

2.4 Copula based generators

The Gaussian Copula model [15] (GCM) is a purely statistical generative model that relies on knowing the distribution shapes of each of its columns. In addition to the distributions, GCM must also calculate the co-variances between the columns. However, the shape of the distributions might unnecessarily influence the covariance estimates [16]. For this reason, the multivariate version of the Gaussian Copula is used. The Gaussian Copula removes any bias that the distribution shape may induce, by converting all column distributions to standard normal before finding the co-variances. Together, the parameters for each column distribution, and the covariance matrix Σ becomes the generative model.

SGN [8] is a segmented generative network where the generation process is split into two frames. One embedding the covariance or copula information in the uniform probability space, and the other embedding the marginal distribution information in the sample domain. This structure also provides an empirical method to sample directly from implicit copulas.

copulaGAN [7] is a variation of the CTGAN Model which takes advantage of the CDF based transformation that the Gaussian Copulas apply to make the underlying CTGAN model task of learning the data easier.

OBT & EBT [10] leverages a copula based generator to augment the data with generated synthetic data, which improves the mean absolute error in emulation-based training (EBT). This is not the case in observation-based training (OBT)

Vine copula AutoEncoders [17] in which a autoencoder (AE) compresses the data into a lower dimensional representation. Then the multivariate distribution of the encoded data is estimated with vine copulas. A generative model is obtained by combining the estimated distribution with the decoder part of the AE.

3 UNIFORMGAN

UniformGAN is a tabular data generator which is based on CTAB-GAN designed to improve modeling speed by transforming continuous variables into uniform probability space in order for the GAN to make learning the underlying distribution easier. UniformGAN adopts a reversible probability integral transform, which consists of bringing input data into a standard normal space by using a combination of cdf and $inversecdf$ transformations. UniformGAN base: CTAB-GAN, uses a mixed-type encoder to better represent mixed categorical-continuous variables as well as missing values. Furthermore it utilises training by sampling. Additionally CTAB-GAN leverages a log-frequency sampler to overcome the mode collapse problem for imbalanced variables[21].

3.1 Architecture of UniformGAN

The architecture of UniformGAN is shown in Figure 2. It comprises of four blocks: a Generator G, Discriminator D, and a classifier C [21]. Additionally we add a probabilistic Transformer T.

Instead of feeding the raw data directly into CTAB-GANs discriminator; the data is first transformed using T. T can handle raw data and detects the data type and transforms it into the desired input for the GAN while setting hyper-parameters in CTAB-GAN related to datatypes.

3.2 Data Transformation

The data transformation consists of two steps, (i) data inference, and (ii) data transformation and reversal.

The first step is inferring which transformation to use based on the object type in the column of the data-set. It maps the python object types to either a categorical, or numerical transformer. The raw data has to be transformed because CTAB-GAN only works on numerical data.

The second step is the data transformation. It utilises the categorical and numerical transformers to map objects into labels, and numerical data into uniform probabilistic space.

The data is transformed based on its column type, if it is an "object" type a categorical transformer is used, if it is an "integer" or "float" type the numerical transformer is used.

The categorical transformers transform the object into labels by utilizing clustering. An example of this is ["True", "False"] -> [0,1] or ["Alice", "Bob", "Bob"] -> [0,1,1]. This labeling step is performed because CTAB-GAN can only handle numerical data.

The numerical transformer performs a statistical transformation on numerical data. First it tries to fit a variable to a multitude of distributions, the difference D between the empirical cdf and the model cdf is calculated with the Kolmogorov–Smirnov test. The distribution with a D closest to 0 is chosen as this most accurately describes the variable.

Next, the probability integral transform is performed, the probability integral transform states that if X is a continuous random variable with cumulative distribution function F_x , then the random variable $Y = F_x(X)$ has a uniform distribution on $[0, 1]$. After these transformations the variable is normally distributed and the original variable can be recovered by applying the inverse cdf also known as percent point function. The goal here is that we are describing a copula, from 2.2. The idea being that if we have random variables; say X_i to X_n with uniform univariate margins $F_i(X_i)$ to $F_n(X_n)$ then there exists a copula function C such that $H(X_i, ..., X_n) = C(F_i(X_i), F_i(X_n))$. The copula function allows to model the margins separately from the dependence structure. This is why we transform all variables to be in standard probability space; in order to capture the pure dependence structure between variables without any affect of the margins.

After the variables are transformed by the transformer T, the CTAB-GAN model applies a second encoding. It converts categorical label columns into on-hot columns, and the continuous and

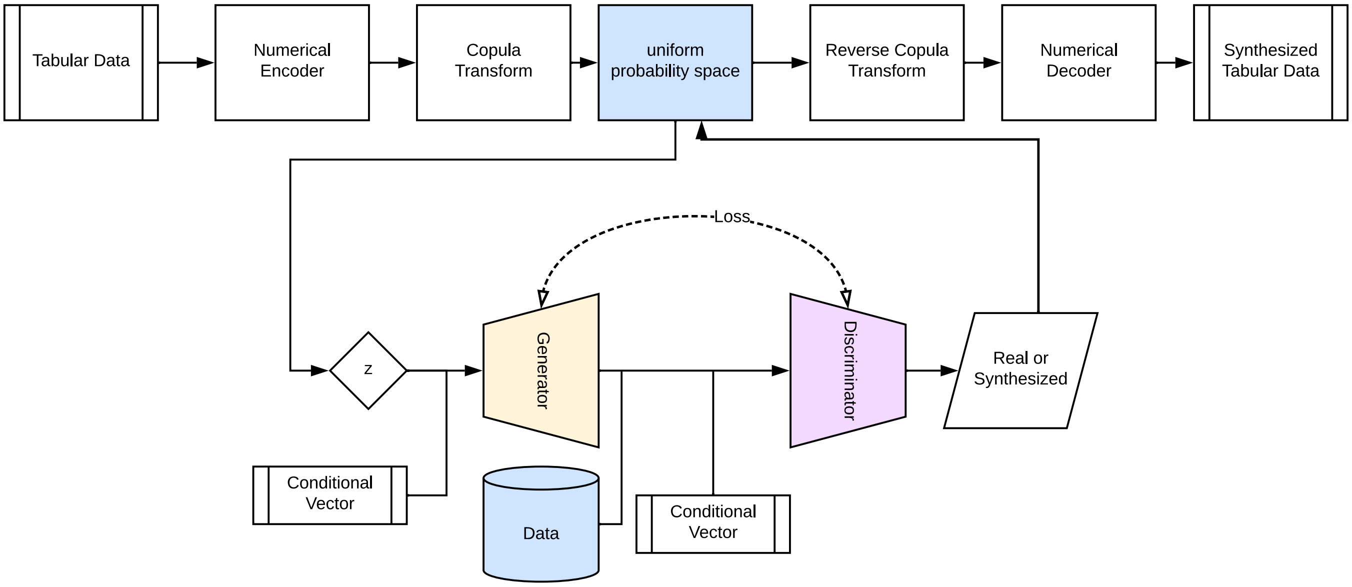

Figure 2: UniformGAN architecture.

mixed columns are treated with a variational Gaussian mixture to estimate the number of modes and to fit a Gaussian mixture. The transformer also passes information to the GAN specifying which columns are categorical, mixed, log or continuous. Removing the need to manually set these parameters.

3.3 Training

After transformation the GAN start the training process. GAN are trained via a zero-sum minimax game where the discriminator tries to maximize the objective, while the generator tries to minimize it. The training process is as follows: The conditional vector along with noise is fed into the generator, the result from the generator is then fed into the discriminator which compares the generated result with the encoded data. This process is displayed in 1. The train/test split is 4/1.

To enhance the generation quality, CTAB-GAN[21] incorporates three extra terms in the loss function of the generator: information [14], classification [13] and generator loss [19]. The information loss penalizes the discrepancy between statistics of the generated data and the real data. This helps to generate data which is statistically closer to the real one. The classification loss requires to add to the GAN architecture an auxiliary classifier in parallel to the discriminator.

3.4 Sampling

When training is finished, we can sample by simply feeding the generator with a conditional vector and a noise vector. The output generated will be in the uniform probability space, so we will need to apply the inverse cdf to transform the data back to the original format, giving us the synthetic data.

4 EVALUATION

In order to evaluate the model we first describe the setup and data, and then explain how the quality of synthetic data is assessed. We will then move on to the main results in terms of data similarity, and training time analysis, and model comparison. We will conclude with an ablation analysis of the integral probability transform as an independent component, and a comparison to CTAB-GAN.

To show the efficacy of the proposed UniformGAN, five commonly used machine learning datasets are selected, and compared with four state-of-the-art GAN based tabular data generators. We evaluate the effectiveness of copulaCTAB-GAN in terms of the resulting ML utility, statistical similarity to the real data, and privacy distance.

4.1 Experimental Setup and Data

4.1.1 Datasets. CopulaCTAB-GAN is tested on five commonly used machine learning datasets. Adult, Covertype, Intrusion, Credit and Loan. All five tabular datasets have a target variable, for which we use the rest of the variables to perform classification. The dataset description can be found in table 1.

4.1.2 Baselines. CopulaCTAB-GAN is compared with 4 state-of-the-art tabular data generators: CTAB-GAN, CopulaCTGAN, Copulas and CTGAN. For Gaussian mixture estimation of continuous variables, we use the same settings as the evaluation of CTGAN, i.e. 10 modes. All algorithms are trained for 50 epochs for Adult, Covertype, Credit and Intrusion datasets, whereas the algorithms are trained for 50 epochs on Insurance dataset. Lastly, each experiment is repeated 3 times.

4.1.3 Environment. Experiments are run under Windows 11 on a machine equipped with 8GB memory, a NVIDIA GeForce GTX 1650 with max-Q design and a 8 core Intel i5 CPU.

Data-set	Train/Test Split	Target Variable	Continuous	Binary	Multi-Class	Mixed-Type	Long-Tail
Adult	39k/9k	income	3	2	7	2	0
Covertype	40k/10k	Cover_Type	10	44	1	0	0
Credit	40k/10k	Class	30	1	0	0	1
Intrusion	40k/10k	Class	22	6	14	0	2
Insurance	1.12k/280	Charges	2	1	3	0	0

Table 1: Dataset Description

4.2 Results

The evaluation of the model is based on three aspects: (1) machine learning utility, (2) statistical similarity and (3) privacy preservability. The first two aspects measure if the synthetic tabular data can be used as a good replacement of the original data. The third aspect evaluates the nearest neighbour distances between the original and synthetic data. For this proposition no effort has been made to improve on privacy statistics. For this reason, we only display privacy preservability results and show that they are similar to state-of-the-art.

4.3 Utility Pipeline

In order to asses how well the synthesized data performs compared to the real data we analyse the machine learning utility. We consider three metrics with respect to machine learning utility; Accuracy difference, Area Under Curve (AUC) difference, and F1-score difference.

The pipeline is setup by comparing the real data utility and the synthesized data utility. This is done by feeding the generated data into a multitude of classifiers. Each classifier will make predictions which are expressed in terms of accuracy, area under curve (AUC) and F1 score metrics. The pipeline will take the difference of the metrics. The pipeline is shown in figure 3.

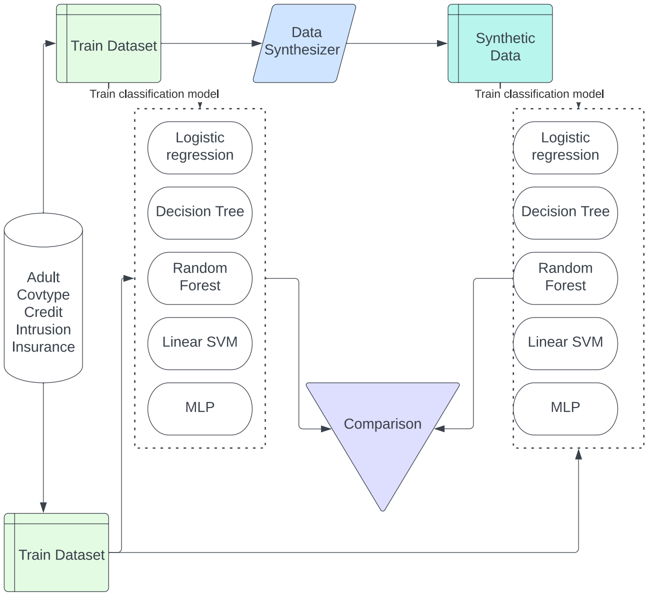

Figure 3: Utility Pipeline

4.3.1 Statistical Similarity. Another important property of the synthesized data is the statistical similarity to the real data. To assess the statistical similarity we consider the average Wasserstein distance, average Jensen–Shannon divergence, and correlation distance.

4.4 Overall results

4.4.1 Copulas. The first question in this thesis is related to generating data with purely statistical methods and its quality. Copulas is a purely statistical tabular data generator, this means it does not have to train like GAN, but it tries to fit every variable with a distribution, and uses a combination of 2.2 and inverse probability sampling, to sample data from these distributions. This method is highly performant when it comes to generation speed. This is why this method is investigated.

The model requires it to learn each variable's distribution, computationally it requires n * m * c computations, where n is the amount of variables, m is the amount of rows, and c is the amount of fit-able distributions.

Even though its synthesising speed is unmatched compared to GAN, the results were dissapointing. It was not able to synthesise enough target variables for the intrusion dataset, just like CTGAN. It did however generate adequate data for the adult dataset, having only a 8.384% difference in averaged accuracy compared to the original dataset.

4.4.2 Transform / Reverse Transform. When the data is transformed using the integral probability transform and transformed back, we want to be sure that the data is the same. To test this, the adult data-set is transformed and reverse transformed. We then run the analysis pipeline to see if the transformation had any effect on the data utility and statistical similarity. From table 2 we can see that the there is a negligible difference in utility between the original and transformed data-set. Furthermore, the statistical test resulted in zero difference between all for WD, JSD and correlation difference. From this analysis we know that we can transform and reverse transform without any significant changes of the data-set.

4.4.3 UniformGAN. To measure the impact of the probabalistic transformer for UniformGAN we compare it against CTAB-GAN, as UniformGAN is identical to CTAB-GAN only adding a probabalistic transformer.

To measure the difference in data generation quality we take the averaged utility statistics for UniformGAN and CTAB-GAN and calculate the difference. These results are shown in table 3.

Model	Acc	AUC	F1
logistic regression	-0.01023	-2.7628e-06	-0.0002
decision tree	-0.07165	-0.0004	-0.0007
random forest	-0.07165	-0.0001	-0.0008
multi layer perceptron	-0.20472	-0.0002	-0.0043

Table 2: Probability integral transform and reverse.

Model	Accuracy	AUC	F1
logistic regression	-0.795	-0.0204	0.0229
decision tree	**15.749**	0.114	0.154
random forest	-1.152	0.00833	0.0117
multi layer perceptron	-0.678	-0.0136	0.0487
support vector machine	-0.960	0.0864	-0.00754

Table 3: Utility difference: UniformGAN versus CTAB-GAN

We can see that UniformGAN performs similarly for most metrics, being roughly 1% worse for most classification models, but outperforming CTAB-GAN on the decision tree classifier (dt).

This difference in the dt utility, explains why UniformGAN outperforms CTAB-GAN on averaged machine learning utility. The majority of this difference comes from the intrusion dataset, where CTAB-GAN had a 83.57% difference in utility compared to the original datatset. While this was only at a 21% difference for UniformGAN.

With these positive results in regard to decision tree classification and a minimal difference in accuracy in other metrics we can conclude that UniformGAN can generate quality synthetic data.

5 RESPONSIBLE RESEARCH

This section is meant to highlight how the research is conducted as responsible as possible. We take a look at reproducability, scientific integrity and specific issues in research practice.

When we want to conduct research responsibly an important aspect is reproducability. If the results can't be replicated, or the replicated results differ from the original then we can never validate the results. This is why there is an extensive section on the experimental setup. To further help other researchers validate paper the code will be shared on GitHub as to increase transparency in the implementation.

Another important aspect of responsible research is integrity, there have been numerous cases in the scientific community where researchers cherry picked data, and/or trimmed data in order to push forward "good" results. In the case of this research there was an aim to pick datasets as to have a broad enough selection of datasets with a variety of datatypes and amount of features. Averaging the results over a multitude of different datasets will give a good indication of performance in real-life scenarios. Furthermore, all ideas borrowed from other authors should be properly cited and given credit.

With relation to GANs there are some specific issues regarding responsible research, an important part of this research is the privacy aspect of synthetic data. In order to assess this there is a section on privacy preservability in the experimental setup.

By addressing all these points I believe to have shown an effort to conduct this research responsibly and with integrity.

6 DISCUSSION

In this discussion we will highlight issues encountered, general advice and future improvements.

During the testing of copulaGAN and copulas some results were worse than expected, which lead me to discover a bug in the learning of the rounding scheme, all results produced in this are with the bug resolved. A pull request will be made to SDV to fix the bug.

Furthermore, during the testing of copulaGAN and copulas, one has to be very mindful of how the data is inferred by the model, one can easily make a mistake by using pre-processed datasets, which are already numeric. This will cause the issue inferring it as purely numerical, while there might be categorical variables. This is easily resolved by specifying the specific transformer to use. When running UniformGAN with more iterations it starts to under-perform compared to CTAB-GAN, this is most likely due to the inability of gaussian copulas to model tail dependence.

7 CONCLUSION

To conclude I want to reiterate on the main research questions of this paper, the first question is can statistical tabular data generators synthesize data with the same accuracy and quality as conditional tabular GAN?

Then we ask if we can generate more accurate synthetic data by leveraging the combination of CTAB-GAN and the integral probability transform. Lastly, we answer on the hypothesis of whether the integral probability transform makes it easier for the GAN to learn the dependence structure of its variables.

From running the utility pipeline we have validated that we can generate statistical tabular data with purely statistcal models, however the quality is subpar compared to state of the art GAN.

The integral probability transform allows UniformGAN to learn the underlying dependence structure with lowered training times, however given enough training-time, CTAB-GAN will converge later on more iterations and outperform UniformGAN, this is most likely due to the inability of gaussian copulas to model tail dependence.

The copula model, copulaGAN, and CTGAN all perform worse than UniformGAN, however copulas requires no additional training besides fitting the distributions and still gives relatively accurate results for datasets without long tail, or extreme outliers.

From running the utility pipeline we can say that it is possible to generate quality data using UniformGAN, while increasing the decision tree classifier utility. The increase in the decision tree classification utility shows us that the integral probability transform makes it easier for the GAN to learn its dependence structure.

ACKNOWLEDGMENTS

Zilong Zhao, Lydia Y. Chen

Model	ML Utility Difference			Statistical Similarity			Privacy Preservation					
							DCR			NNDR		
	Accuracy	AUC	F1-score	Avg JSD	Avg WD	Diff .Corr.	R&S	R	S	R&S	R	S
UniformGAN	**8.708**	**0.115**	**0.176**	**0.013**	**0.0761**	**3.210**	**1.373**	**0.308**	**0.958**	**0.782**	**0.421**	**0.623**
CTAB-GAN	11.205	0.134	0.205	0.331	0.070	1.900	1.260	0.3088	1.0840	0.751	0.4219	0.620
Copulas	18.998*	0.189	0.323	0.0172	0.126	3.703	1.759	0.308	1.584	0.826	0.421	0.745
CopulaGAN	29.97	0.21	0.371	0.082	0.294	5.814	1.424	0.201	0.535	0.815	0.337	0.538
CTGAN	35.442*	0.232	0.356	0.047	0.221	4.57	1.304	0.232	0.831	0.749	0.347	0.61

Table 4: Results 50 epochs: Average over Adult, Covtype, Intrusion and Insurance

1

REFERENCES

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein GAN. (2017). arXiv:1701.07875 http://arxiv.org/abs/1701.07875

[2] Marc G. Bellemare, Ivo Danihelka, Will Dabney, Shakir Mohamed, Balaji Lakshminarayanan, Stephan Hoyer, and Rémi Munos. 2017. The Cramer Distance as a Solution to Biased Wasserstein Gradients. (2017), 1–20. arXiv:1705.10743 http://arxiv.org/abs/1705.10743

[3] Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F. Stewart, and Jimeng Sun. 2017. Generating Multi-label Discrete Patient Records using Generative Adversarial Networks. 68 (2017), 1–20. arXiv:1703.06490 http://arxiv.org/abs/1703.06490

[4] Justin Engelmann and Stefan Lessmann. 2021. Conditional Wasserstein GAN-based oversampling of tabular data for imbalanced learning. *Expert Systems with Applications* 174, Ml (2021). https://doi.org/10.1016/j.eswa.2021.114582 arXiv:2008.09202

[5] Martin Haugh. 2016. IEOR E4602: Quantitative Risk Management & IEOR E4703: Monte-Carlo Simulation. *Quantitative Risk Management* 1 (2016). http://www.columbia.edu/{~}mh2078/FoundationsFE/DeterministicCashFlows.pdf{%}0Ahttp://www.columbia.edu/{~}mh2078/FoundationsFE/for{_}swap{_}fut-options.pdf{%}0Ahttp://www.columbia.edu/{~}mh2078/MachineLearningORFE/MCMC{_}Bayes.pdf{%}0Ahttp://www.columbia.edu/{~}mh2078/MonteC

[6] MIT Data To AI Lab. 2018. *CopulaGAN Model Model Description.* https://sdv.dev/SDV/user_guides/single_table/copulagan.html

[7] MIT Data To AI Lab. 2018. *GaussianCopula Model Kernel Description.* https://sdv.dev/SDV/user_guides/single_table/gaussian_copula.html

[8] Nunzio A. Letizia and Andrea M. Tonello. 2022. Segmented Generative Networks: Data Generation in the Uniform Probability Space. *IEEE Transactions on Neural Networks and Learning Systems* 33, 3 (2022), 1338–1347. https://doi.org/10.1109/TNNLS.2020.3042380

[9] Zinan Lin, Giulia Fanti, Ashish Khetan, and Sewoong Oh. 2018. PacGan: The power of two samples in generative adversarial networks. *Advances in Neural Information Processing Systems* 2018-December (2018), 1498–1507. https://doi.org/10.1109/jsait.2020.2983071 arXiv:1712.04086

[10] David Meyer, Thomas Nagler, and Robin Hogan. 2020. Copula-based synthetic data generation for machine learning emulators in weather and climate: application to a simple radiation model. *Geoscientific Model Development Discussions* January (2020), 1–21.

[11] Alejandro Mottini, Alix Lheritier, and Rodrigo Acuna-Agost. 2018. Airline Passenger Name Record Generation using Generative Adversarial Networks. (2018). arXiv:1807.06657 http://arxiv.org/abs/1807.06657

[12] Arvind Narayanan and Vitaly Shmatikov. 2008. Robust de-anonymization of large sparse datasets. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*. IEEE, 111–125.

[13] Augustus Odena, Christopher Olah, and Jonathon Shlens. 2017. Conditional Image Synthesis with Auxiliary Classifier GANs. (2017). arXiv:arXiv:1610.09585v4

[14] Noseong Park, Mahmoud Mohammadi, Hongkyu Park, and Youngmin Kim. 2018. Data Synthesis based on Generative Adversarial Networks. 11, 10 (2018). arXiv:arXiv:1806.03384v5

[15] Neha Patki, Roy Wedge, and Kalyan Veeramachaneni. 2016. GaussianCopula - The synthetic data vault SDV. *Proceedings - 3rd IEEE International Conference on Data Science and Advanced Analytics, DSAA 2016* (2016), 399–410. https://dai.lids.mit.edu/wp-content/uploads/2018/03/SDV.pdf

[16] Ludger Rüschendorf. 2013. Mathematical risk analysis. *Springer Ser. Oper. Res. Financ. Eng. Springer, Heidelberg* (2013).

[17] Natasa Tagasovska, Damien Ackerer, and Thibault Vatter. 2019. Copulas as high-dimensional generative models: Vine copula autoencoders. *Advances in Neural Information Processing Systems* 32, NeurIPS (2019), 1–23. arXiv:1906.05423

[18] Ruoxi Wang, Gang Fu, Bin Fu, and Mingliang Wang. 2017. Deep & cross network for ad click predictions. *2017 AdKDD and TargetAd - In conjunction with the 23rd ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2017* (2017). https://doi.org/10.1145/3124749.3124754 arXiv:1708.05123

[19] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. 2019. Modeling tabular data using conditional GAN. *Advances in Neural Information Processing Systems* 32, NeurIPS (2019). arXiv:1907.00503

[20] Zilong Zhao, Aditya Kunar, Robert Birke, and Lydia Y. Chen. 2022. CTAB-GAN+: Enhancing Tabular Data Synthesis. 1 (2022), 1–13. arXiv:2204.00401 http://arxiv.org/abs/2204.00401

[21] Zilong Zhao, Aditya Kunar, Hiek Van der Scheer, Robert Birke, and Lydia Y. Chen. 2021. CTAB-GAN: Effective Table Data Synthesizing. 1 (2021). arXiv:2102.08369 http://arxiv.org/abs/2102.08369