KTH ROYAL INSTITUTE
OF TECHNOLOGY

Degree Project in Computer Science and Engineering

Second cycle, 30 credits

Variational AutoEncoders and Differential Privacy : balancing data synthesis and privacy constraints

BAPTISTE BRÉMOND

Stockholm, Sweden, 2024

Variational AutoEncoders and Differential Privacy : balancing data synthesis and privacy constraints

BAPTISTE BRÉMOND

Degree Programme in Computer Science and Engineering
Date: January 15, 2024

Supervisors: Björn Thuresson, Gabriel Prevot
Examiner: Pawel Herman
School of Electrical Engineering and Computer Science
Host organization: Inetum, Department of Innovation
Swedish title: Variational AutoEncoders och Differential Privacy: balans mellan datasyntes och integritetsbegränsningar

Abstract

This thesis investigates the effectiveness of Tabular Variational Auto Encoders (TVAEs) in generating high-quality synthetic tabular data and assesses their compliance with differential privacy principles. The study shows that while TVAEs are better than VAEs at generating synthetic data that faithfully reproduces the distribution of real data as measured by the Synthetic Data Vault (SDV) metrics, the latter does not guarantee that the synthetic data is up to the task in practical industrial applications. In particular, models trained on TVAE-generated data from the Creditcards dataset are ineffective. The author also explores various optimisation methods on TVAE, such as Gumbel Max Trick, Drop Out (DO) and Batch Normalization, while pointing out that techniques frequently used to improve two-dimensional TVAE, such as Kullback–Leibler Warm-Up and B Disentanglement, are not directly transferable to the one-dimensional context. However, differential privacy to TVAE was not implemented due to time constraints and inconclusive results. The study nevertheless highlights the benefits of stabilising training with the Differential Privacy - Stochastic Gradient Descent (DP-SGD), as with a drop-out, and the existence of an optimal equilibrium point between the constraints of differential privacy and the number of training epochs in the model.

Keywords

TVAE, Differential privacy, Tabular data, Synthetic data, DP-SGD

Sammanfattning

Denna avhandling undersöker hur effektiva Tabular Variational AutoEncoders (TVAE) är när det gäller att generera högkvalitativa syntetiska tabelldata och utvärderar deras överensstämmelse med differentierade integritetsprinciper. Studien visar att även om TVAE är bättre än VAE på att generera syntetiska data som troget återger fördelningen av verkliga data mätt med Synthetic Data Vault (SDV), garanterar det senare inte att de syntetiska data är upp till uppgiften i praktiska industriella tillämpningar. I synnerhet är modeller som tränats på TVAE-genererade data från Creditcards-datasetet ineffektiva. Författaren undersöker också olika optimeringsmetoder för TVAE, såsom Gumbel Max Trick, DO och Batch Normalization, samtidigt som han påpekar att tekniker som ofta används för att förbättra tvådimensionell TVAE, såsom Kullback-Leibler Warm-Up och B Disentanglement, inte är direkt överförbara till det endimensionella sammanhanget. På grund av tidsbegränsningar och redan ofullständiga resultat implementerades dock inte differentierad integritet för TVAE. Studien belyser ändå fördelarna med att stabilisera träningen med Differential Privacy - Stochastic Gradient Descent (DP-SGD), som med en drop-out, och förekomsten av en optimal jämviktspunkt mellan begränsningarna för differential privacy och antalet träningsepoker i modellen.

Nyckelord

TVAE, differentiell integritet, tabelldata, syntetiska data, DP-SGD

Acknowledgments

I would like to express my gratitude to the team at the Innovation Department at Inetum, in particular the Data Science Division, for giving me the opportunity to complete this internship. Their patience and understanding in the face of delays and administrative complications on the part of KTH was invaluable.

Special thanks go to Gabriel Prevot, my internship manager, for the trust he placed in me, and to Amaury Lepicard for his invaluable advice and assistance with practical applications related to my research topic.

I would also like to thank Charles Eole Marichal for facilitating my contact with Gabriel Prevot. My gratitude also goes to Björn Thuresson, who generously agreed to supervise this internship and various projects at KTH, and to Pawel Herman, my examiner, without whom this internship would not have been possible.

Finally, I'd like to mention the people who brightened up my stay in Sweden: Thomas Couturou, Antoine Deparday, Aubane Espérance, Manon Lecart and Marine Czaplinski, as well as my family for their ongoing support both in Sweden and during my internship.

Orléans, France, January 2024
Baptiste Brémond

Contents

1 Introduction **1**
1.1 Background . . . 1
1.2 Problem . . . 2
1.3 Purpose . . . 3
1.4 Goals . . . 4
1.5 Delimitations . . . 4
1.6 Structure of the thesis . . . 5

2 Background **7**
2.1 VAE . . . 7
2.1.1 Shortcomings of the VAE . . . 8
2.1.1.1 Loss of variance and blurring of generated images . . . 8
2.1.1.2 Disentanglement . . . 10
2.1.1.3 Variational pruning and posterior law collapse 10
2.1.1.4 Reconstruction/Divergence imbalances . . . 11
2.1.1.5 The "gravity at origin" effect . . . 11
2.1.2 Solutions and Improvements . . . 12
2.1.2.1 A posteriori law regularisation . . . 12
2.1.2.2 Variants of the a priori / a posteriori law . . 13
2.1.2.3 Architectural changes . . . 13
2.1.2.4 Other changes . . . 14
2.2 Gumbel-Max trick . . . 14
2.3 Differential privacy . . . 15
2.3.1 Definition . . . 15
2.3.2 Approximate differential confidentiality . . . 17
2.3.3 Renji differential privacy . . . 18
2.3.4 Implementing differential privacy . . . 18
2.4 The Yeo-Johnson transform . . . 19

2.5 Related Work 20

3 Methods **23**
3.1 Data 23
3.1.1 Pre-processing 24
3.1.2 Post-processing 26
3.1.3 Data separation 27
3.2 Choice of method and evaluation metric 27
3.2.1 Choice of methods 27
3.2.2 Choice of evaluation metrics 28
3.2.2.1 VAE performance evaluation 28
3.2.2.2 Classification performance evaluation 31
3.3 Experimental protocol 31
3.3.1 Improving the VAE 31
3.3.2 Differential privacy 33
3.3.3 CVAE for MNIST 34

4 Results **37**
4.1 VAE improvements 37
4.1.1 Architecture 37
4.1.2 Schedulers 41
4.1.3 Variants 42
4.1.4 Pre-processing and post-processing 43
4.1.5 Synthetic data quality 44
4.2 Differential privacy applicability 45

5 Discussion **51**
5.1 Results 51
5.2 Limitations of the methods 51
5.3 Industrial impact 52
5.4 Scientific contribution 52
5.5 Ethical and sustainable aspects 53
5.6 Future work 53

6 Conclusion **55**

References **57**

List of Figures

2.1 Examples of data generated with a Conditional Variational Auto Encoder (CVAE) as part of this Master thesis. 9
2.2 Data for the date column in the Berka dataset, in continuous form (left) and in the Tabular Variational Auto Encoder (TVAE) form (right). Obtained as part of the results of this Master thesis. 9
2.3 Correlation matrix for the Berka Czech data set in the form required for TVAE operation. On the left is the real data matrix, and on the right is the synthetic data matrix. Obtained as part of the results of this Master thesis. 10
2.4 From *Ladder Variational Autoencoders* [32]: 6 examples of posterior collapse are shown on the top left; an empirical remedy for these deficiencies is to maintain a factor varying between 10 and 100 for the reconstruction loss ratio to the Kullback–Leibler (KL) divergence, as illustrated by the experiments carried out in this Master thesis. 11

3.1 Illustration of pre-processing for TVAE and Tabular Conditional Generative Adversarial Network (TCGAN). From Modeling Tabular Data using Conditional GAN [20] 25

4.1 Overall score on data formatted for TVAE on the Berka dataset. Activation kernel is the initialisation function for the dense layers of the encoder and decoder, activation log var for the initialisation function of the log variance of the latent space, and activation mean for the initialisation of the latent space. Obtained with Optuna during this master thesis. 46
4.2 Census real correlation matrix on the left, and data generated with the ECB on the right Obtained in Experiment 1 of this Master thesis. 47

4.3 Example of the distribution of the data generated for the Berka dataset obtained with the TVAE. Note the difficulty the model has in mimicking the distribution peaks on the numerical data, despite having only 100 training epochs, and the slight discrepancies between the number of real elements in a given category and the number of synthetic elements. Obtained in Experiment 1 of this Master thesis. 47
4.4 Training curve of the TVAE with no improvement, using the softmax activation for categorical data. From left to right: Berka Czech, Census, Creditcards. Obtained in Experiment 1 of this Master thesis. 48
4.5 Training curve of the TVAE with DO (0.05) and Batch Normalization (BN), using the Gumbel Max trick activation for categorical data. From left to right: Berka Czech, Census, Creditcards. Obtained in Experiment 1 of this Master thesis. . 48
4.6 TVAE training curves on Creditcards. Left Leaky ReLU, centre SELU, right ReLU. . Obtained in Experiment 1 of this Master thesis. 48
4.7 Examples of numbers generated by the CVAE.The real images are on the right, and the generated images are on the left. . . . 49
4.8 Instance of pre-processing success (left, column 73), and failure (right, column 0) on the dataset Census. . Obtained in pre-processing of this Master thesis. 50
4.9 Left: correlation matrix for real Creditcards data. On the right, its synthetic counterpart (obtained with a TVAE trained with LEaky ReLU, DO and BN). . Obtained in Experiment 6 of this Master thesis. 50

1 Results of experiment 1.1 68
2 Results of experiment 1.2 69
3 Results of experiment 1.3 69
4 Results of experiment 2 70
5 Results of experiment 3 71
6 Results of experiment 4 72
7 Results of experiment 5 72
8 Results of experiment 6. Reads the model name: Dataset, initial value then final value of the B-disentanglement, presence or absence of batch normalisation. 73
9 Results of experiment 7 74

10 Results of experiment 8 75
11 Results of experiment 9 76
12 Results of experiment 10.1 77
13 Results of experiment 10.2 77
14 Results of experiment 11.1 78
15 Results of experiment 11.2 78
16 Results of experiment 12: model trained on real data evaluated on real data 79
17 Results of experiment 12: model trained on real data evaluated on synthetic data 79
18 Results of experiment 12: model trained on synthetic data evaluated on synthetic data 80

List of Tables

3.1 Datasets breakdown 27

List of acronyms and abbreviations

AI	Artificial Intelligence
ANOVA	Analysis of Variance
BN	Batch Normalization
CIWAE	Combination importance-weighted auto encoder
CLR	Cyclical Learning Rate
CRF	Conditional Random Field
CVAE	Conditional Variational Auto Encoder
DFC-VAE	Deep Feature Consistent Variational Auto Encoder
DO	Drop Out
DP	Differential Privacy
DP-SGD	Differential Privacy Stochastic Gradient Descent
ELBO	Evidence Lower BOund
GAN	Generative Adversarial Network
GDPR	General Data Protection Regulation
GLM	Generalized Linear Model
GMM	Gaussian Mixture
GVAE	Gaussian Variational Auto Encoder
Info-VAE	Information Variational Auto Encoder
IWAE	Importance-Weighted Auto Encoder
KL	Kullback–Leibler
LLM	Large Language Model
LVAE	Ladder Variational Auto Encoder
MIWAE	Multiply Importance-Weighted Auto Encoder
MMD	Maximum Mean Discrepancy
MNIST	Modified National Institute of Standards and Technology
MSE	Mean Squared Error

PCA	Principal Component Analysis
PII	Personally Identifiable Information
PIWAE	Partially Importance-Weighted Auto Encoder
RDP	Renji Differential Privacy
ReLU	Rectified Linear Unit
SDV	Synthetic Data Vault
SELU	Scaled Exponential Linear
ST-VAE	Synthetic Tabular Variational Auto Encoder
SVAE	Hyperspherical Variational Auto Encoder
TCGAN	Tabular Conditional Generative Adversarial Network
TVAE	Tabular Variational Auto Encoder
UMAP	Uniform Manifold Approximation and Projection
VAE	Variational Auto Encoder
VAE-GAN	Variational Auto Encoder Generative Adversarial Network
VQ-VAE	Vector Quantized Variational Auto Encoder
WAE	Wasserstein Variational Auto Encoder
zCDP	zero Centered Differential Privacy

Chapter 1

Introduction

1.1 Background

Synthetic data generation has experimentd substantial growth, closely mirroring advancements in computational methodologies and machine learning paradigms since the early 2000s [1]. First coined in 1993 by Rubin and Little, the discipline has seen rapid evolution, mainly due to the infusion of transformative machine learning algorithms post-2010 [1]. These developments have enabled increasingly complex frameworks for generating synthetic data, elevating its applicability across various domains [2]. Synthetic data, utilised in diverse applications from healthcare research to financial modelling, serve not only to augment existing datasets but also to address class imbalances frequently observed in specialised tasks [3, 4].

Conventional methodologies for data generation, including probabilistic models such as Gaussian Mixtures (GMMs), Conditional Random Fields (CRFs), Generalized Linear Models (GLMs), as well as decision trees, face significant constraints when it comes to producing synthetic data that is both complex and high-dimensional [5, 6]. In contrast, generative models exploiting deep learning techniques can produce data of exceptional quality but at the cost of increased requirements in terms of data volume for training and computational resources [5, 7].

This ability to produce synthetic data also meets another fast-growing requirement: data anonymisation. Indeed, as international governance bodies introduce more legislation and regulations aimed at safeguarding the rights of their citizens, the creation of data avatars is emerging as a potential solution to the issues associated with database leaks or reverse engineering of the algorithms that manipulate them [5]. Although data avatarisation is

not a guarantee in itself, it can be judiciously coupled with formal, proven mathematical methods to strengthen the level of protection. Among these methods, Differential Privacy stands out by adding specific noise to the data before it is processed to prevent the identification of individuals within the database [8].

In this context, Variational Auto Encoders (VAEs) are emerging as a promising way of generating synthetic tabular data. VAEs are generative models based on neural networks that learn the latent distribution of input data [9]. They are then able to produce synthetic data by sampling this latent space while preserving the significant attributes of the original data [10, 5, 11].

Nevertheless, it should be noted that most commercial players operating in the synthetic data market favour another type of generative model: Generative Adversarial Networks (GANs). These are reputed to be superior to VAEs in terms of generation, although they are with their limitations [11, 12].

1.2 Problem

The democratisation of Artificial Intelligence (AI) and the increasing adoption of machine learning algorithms in industrial applications have created an insatiable demand for extensive, high-quality data [2, 3, 4]. In particular, the emergence of Large Language Models (LLMs) requires extensive crawling of the entire World Wide Web to collect training data that would yield statistically robust models. While this approach can deliver high performance, it presents legal and ethical challenges, especially when data collection involves unauthorised access to proprietary or sensitive information [13].

The need for high-quality datasets is universal across all sectors but more acute in areas such as healthcare, where acquiring labelled data can be challenging. Many medical conditions have sparse datasets due to limited numbers of patients consenting to data collection [14]. Even when such datasets are assembled, they often exhibit significant class imbalances. For example, anomaly detection in financial fraud prevention using the Creditcards dataset reveals an anomaly rate of less than 0.5%, severely compromising the predictive power of classifiers trained on such data [15].

Amidst escalating cases of data exploitation for targeted advertising and user profiling, legislative bodies such as the European Union have enacted strict data protection laws such as the General Data Protection Regulation (GDPR) [16]. Such regulations inhibit not only sharing data between companies but also cross-departmental sharing within organisations. Databases also require robust measures to protect against burgeoning cyber

threats, further complicating the landscape for data collection and use.

Synthetic data generation is emerging as a versatile solution to these multifaceted challenges. Synthetic data can mimic the statistical properties of real-world data, bypassing the hurdles of data scarcity, imbalance, and legal constraints. Furthermore, combining synthetic data with data anonymisation techniques, such as removing Personally Identifiable Information (PII) and implementing Differential Privacy, minimises the risk of unauthorised data exposure [17, 18, 19]. Against this complex background, the thesis addresses the following questions:

1. How do VAE-generated synthetic datasets compare to real-world tabular data in statistical metrics and ML model performance, and what VAE configurations best achieve this?

2. How do Differential Privacy (DP) constraints impact neural networks in terms of model performance (such as accuracy), and how do these vary with different privacy levels?

1.3 Purpose

The focus of this research is to investigate the efficacy of VAEs for synthetic data generation in the context of tabular data, an area conspicuously underrepresented in the existing literature. While VAEs are lauded for their stability and low computational overhead, the preponderance of research in this area is skewed towards GANs, especially for tabular data types [11, 12]. With only one specialised variant, Tabular Variational Auto Encoder (TVAE), there is a glaring gap in scientific efforts to refine or extend this architecture [20].

This research has a twofold academic benefit. First, it explores the transferability of methodologies developed for VAE in multidimensional domains to their one-dimensional counterpart, the TVAE. The findings could either reveal a robust data generation tool or highlight an entirely new avenue of specialised research.

From an Industry 4.0 perspective, searching for efficient, secure, and computationally inexpensive data generation algorithms is imperative. TVAE is a strong candidate that could meet these stringent criteria, offering promising benefits regarding data adequacy and operational efficiency.

By improving the security profile and data quality of synthetic data, this research also has significant societal implications. Improved data security

protocols could alleviate concerns about AI applications, fostering public trust and ensuring compliance with evolving data protection legislation.

1.4 Goals

This study aims to rigorously investigate the potential and limitations of TVAEs in generating synthetic data, thereby filling a significant gap in current academic discourse. This research is divided into three interrelated objectives:

1. Methodological Transfer: The first objective is to assess whether advances in VAEs for image and audio data directly apply to TVAEs. This step involves adapting and evaluating image or sound-specific VAE improvements to their tabular equivalents.

2. Generative performance metrics: The second objective revolves around a multifaceted evaluation of TVAEs based on three central criteria:

   (a) Fidelity and similarity of the generated synthetic data to the original dataset

   (b) Utility in practical application

   (c) Data privacy.

3. Differential Privacy: The third objective involves an in-depth analysis of the applicability of Differential Privacy principles to TVAEs, intending to identify the trade-offs between Privacy and Utility

1.5 Delimitations

The focus of this research is to elucidate the generative potential of VAEs for tabular data while ensuring the utility and Privacy of the synthesised data. Rather than developing new VAE derivatives or modifying TVAEs, this work aims to transpose existing techniques from multidimensional VAEs to their tabular counterparts, serving expediency and exploratory validation.

It is essential to clarify the methodological boundaries of this study. GANs, diffusion models, and LLMs are explicitly outside the scope of this investigation. Despite their proven effectiveness, the intention here is not to develop a commercial product but to fill an existing knowledge gap related to VAE in the context of tabular data.

1.6 Structure of the thesis

Chapter 2 provides an overview of VAEs and their derivatives and an introduction to Differential Privacy in the context of the research problem.

Chapter 3 outlines the research methodology, detailing data collection, pre-processing, and formatting specific to the use of TVAE. It also explains evaluation metrics for synthetic data and methods for assessing model performance under Differential Privacy.

Chapter 4 reports the research results and their subsequent analysis, further discussed in Chapter 5. Conclusions and implications are summarised in Chapter 6.

Chapter 2

Background

2.1 VAE

Variational Auto Encoder (VAE) are generative models incorporating Bayesian variational inference and deep learning. They were first conceptualised in 2013 by Kingma and Welling, simultaneously as Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra [9, 21]. Like AutoEncoders, VAEs are structured around encoder and decoder networks. Between these two components is a probabilistic latent space, which can be interpreted as a regularised, continuous, and probabilistic version of the latent space intrinsic to Auto Encoders. Unlike Auto Encoders, which aim to reproduce the input data as faithfully as possible, VAEs aim to learn the data distribution using damped variational inference and the reparametrisation trick [9, 21].

VAEs have many applications, the most notable being the generation of synthetic data, be it tabular data, images, sound, text, or video [9, 21, 20, 22, 23]. They are also used in data compression, representation learning, and reinforcement learning [9, 21, 24]. They can also perform tasks similar to autoencoders, such as anomaly detection or pre-training other models [25, 26].

The main advantages of VAEs lie in the flexibility of the latent space they generate, their ability to model complex, high-dimensional data distributions, and the stability of their training process. However, they have their drawbacks. The main drawback is that the data produced is lower quality and fuzzier than that produced by competing models such as GANs [27]. VAEs are also very sensitive to the probability distributions used to define the latent space and to generate the data. For the latent space, a multivariate Gaussian distribution is often preferred for its simplicity, but this leads to excessive regularisation and an inability to model a multimodal distribution [28]. In addition, calculating

the loss, which minimises the evidence lower bound (ELBO), often results in an unsatisfactory compromise for both reconstruction and inference [29].

Despite these limitations, VAE has been widely used in various areas of artificial intelligence research and remains an active research topic. Advances in the field can be grouped into four main categories:

- Regularise the a posteriori distribution $Q_w(Z|X)$ to reduce the Evidence Lower BOund (ELBO) or optimise its use.
- Variants of the a priori distribution $P(Z)$ and the a posteriori distribution $Q_w(Z|X)$ to obtain a more representative latent space.
- Architectural changes, either in the types of networks used for the encoder and decoder or in combination with other models.
- Other changes include a new way of training the model or performing variational inference.

2.1.1 Shortcomings of the VAE

2.1.1.1 Loss of variance and blurring of generated images

One of the main shortcomings of Variational Auto Encoder (VAE) is the blurring inherent in the reconstruction phase. This shortcoming is particularly evident in the context of images, which are the main application area for VAE testing (as seen in [30]).

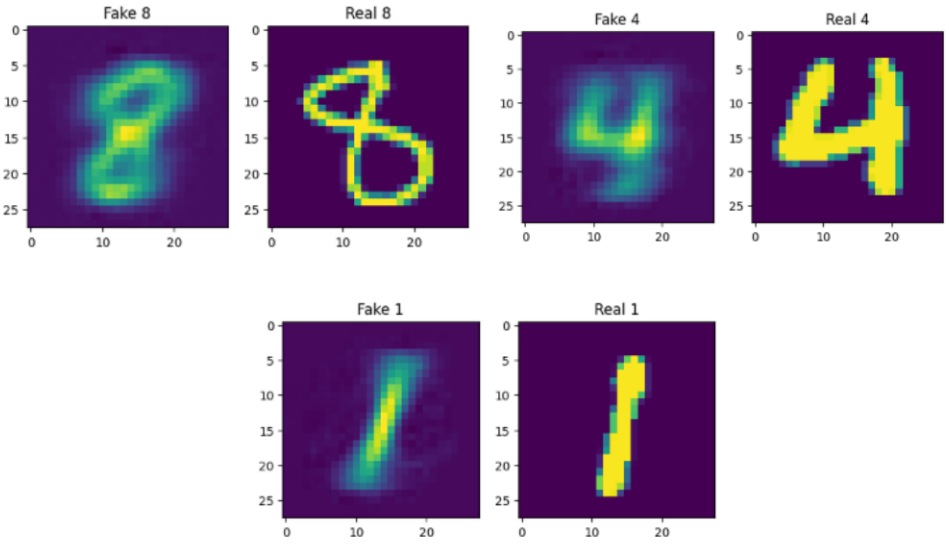


Figure 2.1: Examples of data generated with a Conditional Variational Auto Encoder (CVAE) as part of this Master thesis.

In the case of tabular data, this blurring is reflected in the continuity of the data distribution.

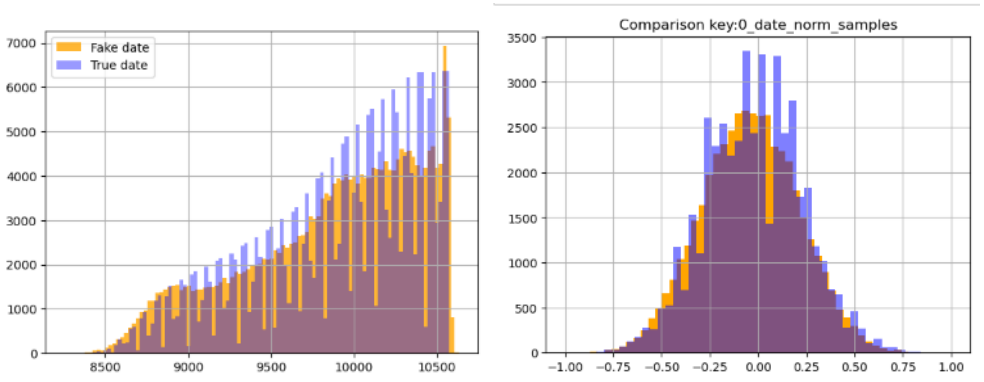


Figure 2.2: Data for the date column in the Berka dataset, in continuous form (left) and in the TVAE form (right). Obtained as part of the results of this Master thesis.

In addition, this imprecision is accompanied by a less faithful reconstruction of the covariance of the original data. The data reconstructed in this way show less interdependence, which may compromise their similarity to the original data. From this point of view, although the distributions of each

column taken individually may be close to the authentic distribution, the data generated as a whole may be inconsistent.

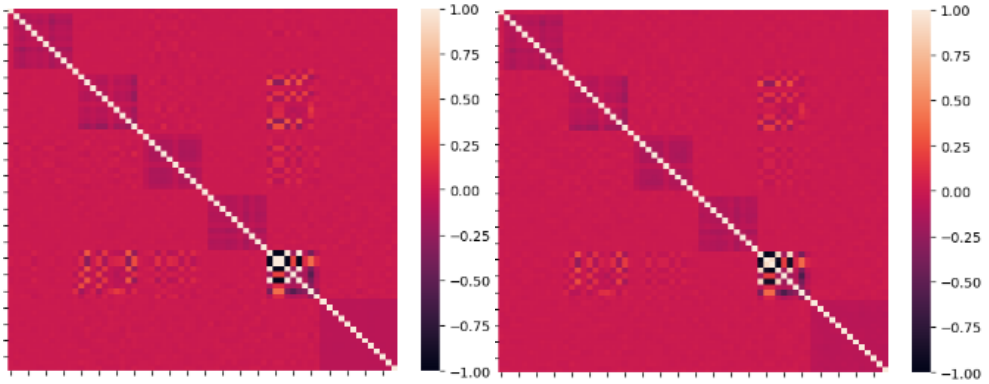


Figure 2.3: Correlation matrix for the Berka Czech data set in the form required for TVAE operation. On the left is the real data matrix, and on the right is the synthetic data matrix. Obtained as part of the results of this Master thesis.

2.1.1.2 Disentanglement

The concept of disentanglement, which refers to the independence of the components of the latent space, provides a better understanding of the impact of each variable on the generation of data and, consequently, better control over the latter. Historically, VAEs have encountered difficulties in this respect, and several advances have been made to improve their performance in this area [31].

2.1.1.3 Variational pruning and posterior law collapse

During the learning phase and the construction of the latent space, some components may fall to 0. These components are not used, forcing the model to work with a smaller latent space that may contain less information. If too many components become 0, the posterior law collapses. This phenomenon occurs in models that are too complex, in the case of poor initialisation or, in particular, in the case of an imbalance between Kullback–Leibler (KL) divergence and reconstruction loss [32, 33].

This imbalance between KL divergence and reconstruction loss is one of the most common difficulties in this course and is the subject of the next section.

2.1.1.4 Reconstruction/Divergence imbalances

The loss function associated with Variational Auto Encoder (VAE) is divided into two distinct terms, each serving different purposes that are not always congruent. During training, one of these objectives may predominate over the other. As a result, the term associated with this goal will decrease significantly, while the other may even increase. If the loss of reconstruction predominates, the model may have excellent reconstruction capabilities but fail to generate adequate data. Conversely, if the emphasis is on minimising the KL divergence while allowing the reconstruction loss to stagnate, I find ourselves in a "posterior collapse" scenario. In this case, the quality of the reconstruction is poor, and the generation of new data is also deficient.

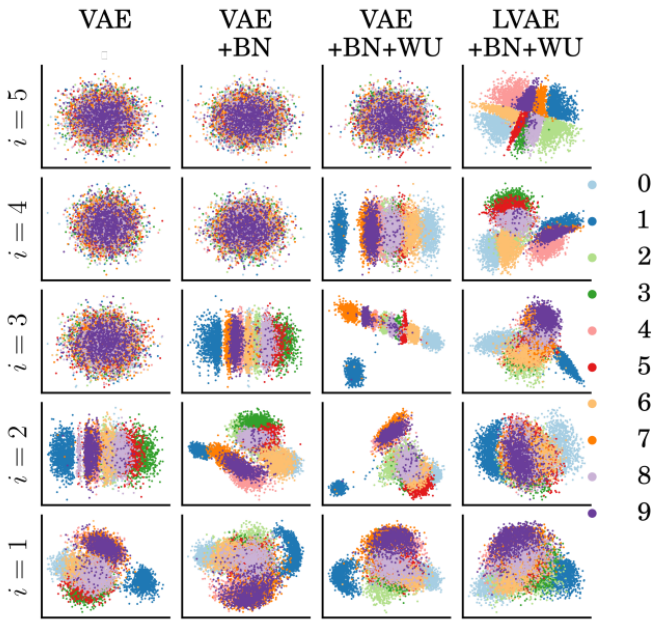


Figure 2.4: From *Ladder Variational Autoencoders* [32]: 6 examples of posterior collapse are shown on the top left; an empirical remedy for these deficiencies is to maintain a factor varying between 10 and 100 for the reconstruction loss ratio to the KL divergence, as illustrated by the experiments carried out in this Master thesis.

2.1.1.5 The "gravity at origin" effect

The prior and posterior distributions are often Gaussians, which are monomodal. This peculiarity of the distribution leads to a concentration of the latent space around a central point for each component, which can lead

to an under-exploitation of the latent space, significantly limiting the amount of information that can be encoded in the model, which not only affects the quality of the reconstruction and data generation but also complicates the process of disentangling latent features [32, 34].

2.1.2 Solutions and Improvements

There are many VAE variants, each seeking to solve one or more of the above problems. Here are a few examples.

2.1.2.1 A posteriori law regularisation

Some variants of VAE seek to weight the KL divergence or replace it with another divergence measure between two distributions.

β-VAE and Disentangled β-VAE

The β-VAE model was designed to optimise the ability of VAE to construct disentangled latent representations. The basic principle lies in weighting the KL divergence term within the loss function, using a hyperparameter called β. By increasing the value of β, it becomes possible to obtain more disentangled latent representations [31]. Furthermore, a variant called Disentangled β-VAE introduces an additional parameter, C. It replaces the KL divergence by the absolute difference between the latter and C while keeping the modified β-VAE weighting (parameter here called Gamma) [34]. This modification aims to relax the constraint imposed by the KL divergence, thus allowing the latent space to contain considerably more information and overcome what the authors call the "information bottleneck." In general, the C value increases as the training progresses, from an initial value (0 or 0.5) in the first epochs to its final value (a value of 25-50 is used in [34]).

Info-VAE

Information Variational Auto Encoder (Info-VAE) modifies the cost function by replacing the KL divergence with a measure of mutual divergence [29]. One of the most common measures of mutual divergence is Maximum Mean Discrepancy (MMD), used in InfoVAE: Information Maximizing Variational Autoencoders [41], which I have used. MMD compares the mean of the samples in a space of kernel functions (kernel space), often Gaussian [35].

KL warm-up and cyclic KL warm-up

The KL warm-up technique, introduced in [32], consists of gradually increasing the weighting of the KL divergence until a target value is reached. This approach allows the model to initially focus on the reconstruction task

before gradually adapting to the constraints imposed by the KL divergence. This sequence avoids the premature collapse of multiple dimensions of latent space during the training phase.

The "cyclical KL warm-up" presented in [33] extends the KL warm-up technique. In this variant, the KL divergence weighting increases and then reaches a plateau for a defined number of epochs before being reset to zero to start the cycle again. This method improves the disentanglement of latent representations and reduces posterior collapse.

2.1.2.2 Variants of the a priori / a posteriori law

Another part of the VAE modifies the latent space and its definition.

VQ-VAE (I and II)

Vector Quantized Variational Auto Encoder (VQ-VAE) uses a discrete latent space and introduces a vector quantisation mechanism to operate in this discrete latent space [36]. An autoregressive model then reproduces the discrete latent space from a normal distribution for data generation. This discretisation allows more complex data to be modelled and better-quality samples to be generated, particularly in word processing. VQ-VAE II also uses a hierarchical architecture, making it more powerful and able to tackle even more complex problems [25].

SVAE

Hyperspherical Variational Auto Encoder (SVAE) is a variant of VAE that replaces the latent space modelled by a multivariate Gaussian with a hyperspherical latent space generated by a Von Mises - Fisher law [37]. This new distribution improves the model's overall performance, especially in cases where the multivariate Gaussian model is too simple.

2.1.2.3 Architectural changes

Other variants combine VAE and other models to exploit the strengths of all these models and mitigate their weaknesses.

VAE-GAN and WAE

The Variational Auto Encoder Generative Adversarial Network (VAE-GAN) replaces the GAN model with a VAE [38]. In addition to entirely generated data, as in the case of a GAN, the VAE produces a reconstruction of the real data presented to the discriminator, which must then be classified as synthetic data. The Wasserstein Variational Auto Encoder (WAE) is a VAE-GAN that uses the Wasserstein measure as the divergence measure instead of the KL divergence [39].

2.1.2.4 Other changes

Finally, some variants fall into more than one of these categories or modify not the VAE itself but the form of the input data or the activation functions of the output layer.

TVAE

The best-performing model for tabular data generation, developed in parallel with TCGAN in Modeling Tabular Data using Conditional GAN [20], has a unique feature. It uses data formatted for Tabular Conditional Generative Adversarial Network (TCGAN) and batch variance (i.e., identical variance for all simultaneously generated or reconstructed data) for each output instead of the dense variance used in other VAEs. In addition, the Synthetic Data Vault (SDV) specifies that TVAE training is not based on a reconstruction loss measured by the Mean Squared Error (MSE) but rather on the least squares method weighted by the digital outputs' variances. The logarithms of these variances are also added to the loss function. Note that clipping limits these variances to an interval between 0.01 and 1.

IWAE, MIWAE, CIWAE, PIWAE

These variants, including Importance-Weighted Auto Encoder (IWAE), attempt to improve the estimation of the likelihood's lower bound using importance-weighting techniques. IWAE uses multiple samples to approximate the posterior distribution, which allows it to model more complex posterior distributions than the standard VAE [40]. Multiply Importance-Weighted Auto Encoder (MIWAE) is an extension of IWAE that handles missing data by maximising a potentially tight lower bound on the likelihood of the observed data [41], while Combination importance-weighted auto encoder (CIWAE) and Partially Importance-Weighted Auto Encoder (PIWAE) modify the way importance weights are used to ensure a better signal-to-noise ratio of gradient estimators [41].

2.2 Gumbel-Max trick

The use of Gumbel random variables facilitates the sampling of categorical data [42], but this sampling is not differentiable. The Gumbel-Max trick is similar to the reparametrisation trick and overcomes this difficulty. A Softmax function often follows it and performs better than the latter alone on a data set containing categorical data to be generated [43].

2.3 Differential privacy

As more and more data is collected daily from users of Internet technologies such as mobile applications and social networks, there has been a need to protect users and give them more control over their data. Several techniques for anonymising stored data were introduced in the 2000s, including k-anonymity, l-diversity, and t-closeness [44, 45]. Although they provide some protection, these two techniques remain vulnerable to certain attack types and are somewhat limited in their initial versions [46].

In 2006, C. Dwork et al. published "Differential Privacy," demonstrating the mathematical foundations of the technique of the same name [17]. Differential Privacy was introduced as a means to achieve privacy-preserving data analysis. The idea behind differential privacy is to add carefully designed noise, parameterised by a value called ϵ, to the data to ensure that the statistical properties of the data remain intact while preventing the identification of an individual. This technique is known as ϵ DP. DP provides a strong, mathematically proven confidentiality guarantee, independent of the adversary's knowledge or resources, at the cost of response quality that can be adjusted according to one's needs [17, 18].

Due to the inherent structure of DP, some queries, particularly those on groups of individuals or repeated queries on the same individual, compromise more information than a single query on an individual would [18]. An illustrative example is the naive implementation of Report Noisy Max, which identifies the maximum value after adding ϵ-DP-compliant noise to each element of a set R and then returns this noisy maximum. In this case, the loss of privacy is estimated as the product of the number of elements in R and ϵ, resulting in a linear degradation of ϵ-DP performance as a function of the number of elements. To overcome these challenges, more than 200 variants and improvements to DP have been developed since 2006 [19]. The four most well-known methods are the Renji DP, the ρ-zero Centered Differential Privacy (zCDP), the Exponential Mechanism and the Parsimonious Vector Technique [18]. I will focus on the Renji DP.

2.3.1 Definition

Formally, differential privacy is defined as follows:

"A function that respects differential privacy is often called a mechanism. I say that a mechanism F satisfies differential privacy if, for all neighbouring data sets x and x' and all possible result sets S,

$$\frac{\mathbf{Pr}[F(x) \in S]}{\mathbf{Pr}[F(x') \in S]} \leq e^{\epsilon} \tag{2.1}$$

" [18]

The ϵ value is more empirical and is used to determine the level of privacy: an ϵ of 1 is appropriate for a high level of privacy. In contrast, an ϵ of 10 compromises privacy and security.

One way of ensuring differential confidentiality is to add noise to the result of a query. One way to add noise while respecting the definition of differential confidentiality is the Laplace mechanism:

"According to the Laplace mechanism, for a function $f(x)$ that returns a number, the following definition of $F(x)$ satisfies ϵ-differential privacy:

$$F(x) = f(x) + \mathbf{Lap}\left(\frac{s}{\epsilon}\right) \tag{2.2}$$

Where S is the sensitivity of f, and Lap(S) denotes sampling from the Laplace distribution with centre 0 and scale S" [18].

Differential Privacy has three major properties that make it simple to use:

Sequential composition "- If $F_1(x)$ satisfies differential privacy ϵ_1

- and $F_2(x)$ satisfies differential confidentiality ϵ_2,

- The mechanism $G(x)$ = ($F_1(x)$, $F_2(x)$) that releases both results respects (ϵ_1 + ϵ_2) differential confidentiality." [18]

This property makes it possible to use several differential confidentiality mechanisms simultaneously and to know the total confidentiality cost.

Parallel composition "- If $F(x)$ satisfies differential privacy ϵ.

- And we divide a data set X into k disjoint pieces such that $x_1 \cup ... \cup x_k$ = X

- The mechanism that releases all results $F(x_1)$, ..., $F(x_k)$ satisfies differential privacy ϵ." [18]

This property proves particularly beneficial when all the data is distinct. For example, it allows a histogram with n intervals to be created by performing multiple counting requests while only incurring a cost of ϵ rather than nϵ.

Post-processing "If F satisfies differential privacy on x with ϵ, then, for any function G (deterministic or random), $G \circ F$ satisfies differential privacy on x with ϵ.

Once the differential privacy mechanism has been implemented, the subsequent process is guaranteed to remain under the umbrella of differential privacy, whatever operations are carried out." [18]

Finally, it should be noted that differential confidentiality only secures

data at the level of a single line in the database. If an attacker obtains several complete lines from the database before launching his attack, or if an individual is associated with several identical lines in the same database, the protection afforded by differential confidentiality is reduced accordingly.

2.3.2 Approximate differential confidentiality

The definition of differential confidentiality given so far is very restrictive, and the ϵ confidentiality cost increases very quickly, which limits the operations that can be carried out on a database for a given confidentiality budget. One solution is to relax the constraints imposed by formulating an approximate differential confidentiality [18]:

$$\mathbf{Pr}[F(x) = S] \leq e^{\epsilon}\mathbf{Pr}[F(x') = s] + \delta \tag{2.3}$$

The new delta parameter corresponds to the probability that the differential confidentiality will not be respected. A tiny factor is generally used (the dataset size squared, for example). In the case where differential confidentiality is not respected, there is no catastrophe, such as total access to the dataset by the entity that made the request, but rather a response that respects differential confidentiality for a higher cϵ parameter.

Approximate differential privacy satisfies the properties of sequential, parallel and post-processing composition.

Instead of the Laplace mechanism, it uses a Gaussian noise mechanism [18]:

$$F(x) = f(x) + \mathcal{N}(\sigma^2), \text{ where } \sigma^2 = \frac{2s^2 \log\left(\frac{1.25}{\delta}\right)}{\epsilon^2} \tag{2.4}$$

This Gaussian mechanism substantially reduces confidentiality costs when working with integer vectors rather than scalars. One can extend differential confidentiality to vectors by using the L1 norm with the Laplace mechanism and the Gaussian mechanism with the L2 norm. Whereas the differential confidentiality of a query of sensitivity 1 on a vector of dimension k will be worth k in the case of pure differential confidentiality, it will only be worth $\sqrt{k}$ in the case of approximate differential confidentiality.

A final important point to mention is the advanced composition theorem: It provides a more nuanced way of analysing the cumulative loss of confidentiality in sequential compositions of differentially private mechanisms. Specifically, the theorem states:

"- If each mechanism m_i in an adaptive composition k-fold m_1, ... m_k satisfies differential privacy ϵ.

- Then, for any $\delta' \geq 0$, the set of adaptive composition k satisfies (ϵ', δ')-differential privacy, where:

$$\epsilon' = 2\epsilon\sqrt{2k\log\left(\frac{1}{\delta'}\right)} \tag{2.5}$$

" [18]

This formula is particularly advantageous when the number of iterations k is high, as it often lowers the privacy loss bound compared to traditional sequential composition methods.

2.3.3 Renji differential privacy

Renji Differential Privacy (RDP) is an innovative extension to the conventional differential privacy framework, specifically designed to establish a tighter bound in the definition of differential privacy, particularly in the context of iterative algorithms. This pioneering approach was first introduced by Ilya Mironov in 2017 [47]. In RDP, a new parameter, symbolised by α, is introduced alongside ϵ to articulate the privacy guarantees provided. Again, according to [18] :

$$F(x) = f(x) + \mathcal{N}(\sigma^2), \text{ where } \sigma^2 = \frac{\Delta f^2 \alpha}{2\bar{\epsilon}} \tag{2.6}$$

A notable feature of RDP is its simplified sequential composition theorem. If a mechanism F1 adheres to (α, ϵ)-RDP and another F2 to (α, ϵ')-RDP, their sequential composition will naturally respect (α, ϵ +ϵ')-RDP. This feature makes RDP invaluable for algorithms that iteratively deploy privacy mechanisms, providing a strict upper bound on the entire disclosure of private information. In addition, RDP can be directly converted to (ϵ, δ)-differential confidentiality, making it versatile and compatible with existing differential confidentiality architectures.

2.3.4 Implementing differential privacy

Renji's differential privacy is used in practice for all its qualities and its low cost of implementation. One of the techniques for implementing Renji's differential privacy is Differential Privacy Stochastic Gradient Descent (DP-SGD), described in Deep Learning with Differential Privacy [48]. It is

sufficient to apply three additional operations when calculating the gradient when training the model:

1. Before starting training, choose the noise to be added and the clipping parameter C.
2. Calculate the gradient of a batch.
3. Apply clipping so that the L2 norm of the gradient is less than or equal to C.
4. Add Gaussian noise.
5. Update the model parameters.
6. Consider the confidentiality budget consumed at each stage.
7. Repeat steps 2 to 7 until the model converges.

The parameter C is equivalent to the square of the desired gradient sensitivity. As a general rule, leave $C = 1$. The noise parameter is equivalent to $\frac{\alpha}{2\epsilon'}$. Before training, I test several α values to find the one that gives the best differential privacy.

Google has implemented differential privacy twice through its subsidiaries: once through Tensorflow in the Tensorflow Privacy package and once in the Google Privacy module [49, 50]. Tensorflow Privacy can be used quickly on simple models, such as sequential ones. Google Privacy calculates the confidentiality cost of operations when given the α and ϵ values, for example. One of the conditions imposed in practice to further reduce the confidentiality cost is the confidentiality amplification by subsampling introduced in [51] using a Poisson process to select the elements that will make up the batch at each training stage. New parameters appear, such as the sampling rate of the Poisson process, and the confidentiality cost can easily be divided by 10.

2.4 The Yeo-Johnson transform

As part of transforming the data into the format expected by the TVAE, the Yeo-Johnson transform was used [52], and for the sake of completeness, here is a presentation of this formula.

$$\Psi(\lambda, y) = \begin{cases} ((y+1)^{\lambda} - 1)/\lambda & \text{if } \lambda \neq 0, y \geq 0 \\ \log(y+1) & \text{if } \lambda = 0, y \geq 0 \\ -[(-y+1)^{(2-\lambda)} - 1]/(2-\lambda) & \text{if } \lambda \neq 2, y < 0 \\ -\log(-y+1) & \text{if } \lambda = 2, y < 0 \end{cases}$$

The Yeo-Johnson transform is a data normalisation technique that generalises the Box-Cox transform by allowing negative values in the data set. It is defined by a family of parametric functions controlled by a parameter λ, which generally varies in an interval such as [-5,5]. The transformation is defined for a variable x. This transformation aims to make the data distribution closer to normal, which is often desirable for statistical methods that make assumptions about the data distribution. The parameter λ is usually estimated from the data using likelihood maximisation. The Yeo-Johnson transform is beneficial when working with data containing positive and negative values, in case a transform defined for both is needed, unlike the Box-Cox transform.

2.5 Related Work

In March 2023, almost the entire state of the art in VAE is focused on image and sound, and there are many variants of this model in addition to those implemented in this thesis. The Pythae initiative groups about twenty of them, compares them in terms of generation performance, and makes a library of these VAE variants freely and openly available [30]. My documentary research has found over forty VAE improvements and corrections [9, 21, 20], among others*. However, I found only two have been tested on anything other than images in their original publication: the Music VAE on sound and the TVAE on tabular data [20, 22]. With the Synthetic Tabular Variational Auto Encoder (ST-VAE) proposed by Unlocking the Potential for Synthetic Tabular Data Generation [75], these are the only two publications I have found that apply VAE to tabular data.

In Unlocking the Potential for Synthetic Tabular Data Generation [75], van Bree presents the ST-VAE, a classic VAE using the Drop Out (DO) and the Gumbel Max trick, and the Gaussian Variational Auto Encoder (GVAE), a VAE trained with a Gaussian likelihood used as the reconstruction loss. Their performance is compared with that of several GAN variants (TGAN, WGAN, MedGAN and TableGAN), and they find that ST-VAE and GVAE consistently outperform the GAN variants on Berka and Creditcards,

*[40, 41, 53, 30, 31, 32, 34, 29, 39, 33, 54, 55] [56, 57, 36, 37, 28, 58, 59, 60, 61, 62, 38] [63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74]

respectively, highlighting the weakness of VAE in reproducing categorical data.

In Modeling Tabular Data using Conditional GAN [20], Xu et al. propose the Tabular Conditional Generative Adversarial Network (TCGAN), a GAN trained on tabular data with a unique data shaping and data conditioning process during the generation that improves the performance of a GAN and allows it to control the categorical values of the generated data partially. They also introduce the TVAE, already presented in section 2.1.2.4, and compare the performance of the two models with the state of the art in synthetic data generation in 2021. While TVAE proves to be the best model in terms of the negative log-likelihood of the data generated, the authors question its ability to be applied industrially and highlight the flexibility of the new TCGAN. Many websites currently use the latter model of synthetic data [76, 77, 78, 79, 80].

In 2023, the implementation of the models from Modeling Tabular Data using Conditional GAN [20] resides in the Git repo of the Synthetic Data Vault (SDV), an open-source initiative that contains many other models, allowing it to generate not only tabular data but also complete databases. The site is still under development, with many new metrics in the testing phase [81]. In particular, the SDV is looking to develop metrics to measure the anonymisation of a synthetic data table. One of the metrics being developed is to compare the number of rows in common between the synthetic table and the real table.

Faced with this lack of tools to measure privacy, I have to turn to guarantees such as differential privacy and its implementation by Tensorflow via Tensorflow privacy [49] and Google via Google Differential Privacy [50]. These two libraries are still active in 2023, with changes being made every week, showing that Differential Privacy is still a development priority for companies. In the research world, as mentioned in section 2.3, more than 200 variants of this technique have been developed over the last 18 years [19]. However, the most efficient variant applicable to machine learning remains RDP with a Poisson sampling data loader [51].

My work therefore takes the TVAE, a model created in Modeling Tabular Data using Conditional GAN [20], and tests various improvements made to the VAE to compensate for known weaknesses in the VAE (see Section 2.1), as well as weaknesses specific to the TVAE discovered during this Master thesis. At the same time, I verify the applicability of a transfer of techniques between multidimensional VAE and unidimensional VAE. I verify their applicability on 3 very different datasets proposed in Unlocking the Potential for Synthetic Tabular Data Generation [75] for possible comparison.

Chapter 3

Methods

3.1 Data

In this Master thesis, my research involved five datasets: Census 1994, Creditcards, Berka Czech, Modified National Institute of Standards and Technology (MNIST), and Churn Modelling. The choice of Census 1994, Creditcards, and Berka Czech aligns with the rationale presented in Unlocking the Potential of Synthetic Data [75]. These datasets are large, accessible without specialist knowledge, and have a wide range of characteristics. This diversity strengthens the generalisability and robustness of my findings and avoids the pitfall of overfitting to a particular dataset type.

Specifically, the Census 1994 dataset is derived from the 1994 and 1995 US Census data, stripped of PII to prevent re-identification through cross-referencing [82]. It includes 40 attributes, predominantly categorical (32 out of 40). This dataset was selected to evaluate the performance of my synthetic data generation method on predominantly categorical data, a scenario where generative models often show reduced effectiveness [20]. After excluding entries with zero values or those that were sparsely represented, I retained 146,450 records from the initial 299,285. A value is considered sparsely represented if it represents less than one-tenth of its expected frequency in a balanced dataset. This filtering criterion was the only pre-processing step applied to the Census dataset.

The Czech Berka dataset is a publicly available collection of anonymised bank transactions [83]. My analysis used only the 'Trans' table, which - after removing primary and foreign keys - contains eight attributes: an even split of four numerical and four categorical. Due to its manageable size and balanced structure, Berka Czech served as the primary testbed for this work. After

eliminating records with null values, the dataset was reduced from over one million records to 230,465. Although alternative pre-processing approaches were conceivable, such as class merging or introducing a 'null' category, I opted for simplicity and the ability to compare my results with those found in Unlocking the Potential of Synthetic Data [75]. The only significant pre-processing change I made to Berka Czech was to convert the transaction data into a continuous numeric format, as per the data handling protocol used by the Synthetic Data Vault (SDV) [81].

The Creditcards dataset, a well-recognized fraud detection resource, plays a central role in my study [15]. Consisting of a time-step column, twenty-eight features derived from a Principal Component Analysis (PCA) of bank data characteristics, and a categorical column to indicate anomalies, it serves as my benchmark for digital data representation. Notably, the Creditcards dataset required no pre-processing, allowing its entirety - 284,807 rows - to be used, providing a comprehensive basis for my analysis.

I also included the dataset of the MNIST. This dataset is a collection of 70,000 greyscale images of handwritten digits, each 28x28 pixels in size. This dataset, divided into 60,000 training and 10,000 test images, is accompanied by labels indicating which digit each image represents (0 to 9). Known for its application in supervised learning, particularly in Deep Learning for image recognition [84], MNIST was chosen to validate the performance of my modified 1D VAE. Its simplicity and readiness for use without pre-processing allowed for the efficient use of resources at this study stage.

Finally, the Churn Modelling dataset was used as a further benchmark for anomaly detection [85]. This dataset contains 10,000 rows with 12 characteristics (10 excluding columns with personally identifiable information), consisting of eight numeric and two categorical columns. A class label indicates whether an entry is an anomaly. For this dataset, I used the pandas dummies version of these variables . As with MNIST, Churn_Modelling was selected for its unadulterated and straightforward nature, requiring no pre-processing. This selection criterion was instrumental in ensuring the reproducibility of my research findings.

3.1.1 Pre-processing

Implementing the TVAE requires a specific pre-processing scheme. As applied to the Census, Berka, and Creditcards datasets, this methodology follows the procedures described in Xu et al.'s Modelling Tabular Data Using Conditional GAN [20].

The pre-processing includes transforming categorical data into the One Hot Encoding format, similar to the method used for Pandas dummies, and transforming numerical data into a normal distribution in one column and columns using the One Hot Encoding format. The process is the following:

1. **Bayesian Gaussian Mixture Model:** Each numerical column is fitted with a Bayesian Gaussian mixture model with ten components, similar to the Synthetic Data Vault approach. Kernels with insufficient representation (less than 0.5% of a balanced distribution with ten kernels, for instance) are merged with the nearest kernel.

2. **Kernel mapping:** Each row is probabilistically assigned to a kernel, weighted by its likelihood of belonging to that kernel.

3. **Normalisation:** The data in each row are normalised using the mean and standard deviation of the corresponding kernel, followed by a division of 4.

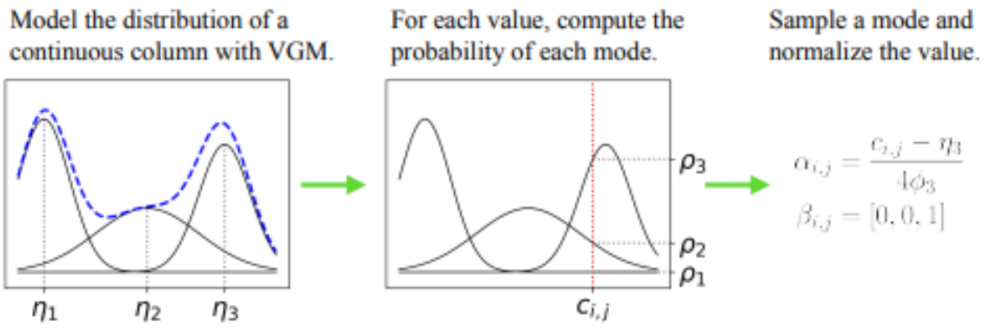


Figure 3.1: Illustration of pre-processing for TVAE and TCGAN. From Modeling Tabular Data using Conditional GAN [20]

Theoretically, the result of this process is a new normalised numerical column, while the One Hot Encoding columns represent the kernel memberships. This transformation improves the performance of the TVAE, along with its loss function (see section 2.1.2.4). It offers better results than the traditional VAE on tabular data (see section 4.1). However, this transformation is not always successful with numerical data. In case of a failure, the transformed numerical column does not resemble a normal distribution, which negatively affects the generative capabilities of the TVAE. This problem was particularly prevalent in the transformed Census and Berka datasets.

To mitigate these challenges, I included two Yeo-Johnson transformations: one before the data transformation process (as step 0) and another after (as step 4). The Yeo-Johnson transformation was chosen for its effectiveness in reducing data asymmetry and approximating a normal distribution, as observed in the Berka dataset (see Figure 7). Crucially, this transformation is bijective, ensuring no loss of information and allowing the data to be back-transformed - from the TVAE format to its pre-transformation state. This approach solved the transformation problems encountered with the Berka and Creditcards datasets, although challenges remain with the Census dataset (see Figure 7).

The operational framework of the TVAE necessitates working with data in a specific format. Consequently, the synthetic data it produces are also in this bespoke format. However, a crucial aspect of producing realistic synthetic data is the ability to reverse the TVAE's formatting process, ensuring that the synthetic data closely resemble the original datasets. This reversibility is a crucial feature of the data processing techniques employed in my methodology. As mentioned above, the Yeo-Johnson transformations are inherently reversible. This bijective nature extends to transforming categorical data into One Hot Encoding. In numerical data, random reassignment is not used, unlike pre-processing. Instead, a deterministic approach is used to reconstruct each numerical column. This reconstruction uses the mean and standard deviation corresponding to the One Hot Encoding associated with a particular numeric column. The effectiveness of this deterministic reconstruction ensures that the original data structure and statistical properties are retained as closely as possible in the synthetic data.

3.1.2 Post-processing

The operational framework of the TVAE necessitates working with data in a specific format. Consequently, the synthetic data it produces are also in this bespoke format. However, a crucial aspect of producing realistic synthetic data is the ability to reverse the TVAE's formatting process, ensuring that the synthetic data closely resemble the original datasets. This reversibility is a crucial feature of the data processing techniques employed in my methodology.

As mentioned above, the Yeo-Johnson transformations are inherently reversible. This bijective nature extends to transforming categorical data into One Hot Encoding. In numerical data, random reassignment is not used, unlike pre-processing. Instead, a deterministic approach is used to reconstruct each numerical column. This reconstruction uses the mean

and standard deviation corresponding to the One Hot Encoding associated with a particular numeric column. The effectiveness of this deterministic reconstruction ensures that the original data structure and statistical properties are retained as closely as possible in the synthetic data.

As a result, the synthetic data, although generated in a format optimised for the TVAE, can be converted back to a format that reflects the original data sets in both structure and statistical properties. This capability not only enhances the realism and utility of the synthetic data but also underlines the robustness and adaptability of the TVAE in dealing with different data types and formats.

3.1.3 Data separation

We have used the same values for separating data across all datasets. Using the sckikit learn function Train test split with a seed of 417. The first time, to separate 20% of the data to form the test set. The second time, on the remaining data to separate the 16% of the data used to form the validation set from the 64% used to form the training data.

Dataset	Total	Train	Validation	Test
Census	146450	93758	23432	29290
Berka	230465	147497	36875	46093
Creditcards	284807	182276	45569	56962
MNIST	70000	44800	11200	14000
Churn Modeling	10000	6400	1600	2000

Table 3.1: Datasets breakdown

3.2 Choice of method and evaluation metric

3.2.1 Choice of methods

The limitations of the thesis restricted the tools to VAE and its variants. As mentioned in section 2.5, most commercialised synthetic tabular data are generated using GANs [76, 77, 78], and diffusion models have achieved promising results [86]. Nevertheless, only VAE has been studied in this work.

My first objective was generating synthetic tabular data using VAE and their variants. The model is trained using an unsupervised approach where the input is reconstructed from a latent space projection. This approach eliminates the need for labels, as exemplified by data sets such as Berka Czech. An

advantage of this training method is its ability to utilise a larger volume of data, increasing its applicability in industrial settings.

This paradigm was challenged by the Conditional Variational Auto Encoder (CVAE) tests, as it uses the labels associated with the data to control the class of synthetic data generated. While I had the labels for the MNIST dataset to validate the architecture and schedulers, such as the KL warm-up, this was not the case for Berka, for example. I then used the training by sampling proposed in Modelling tabular data using conditional GAN [20], which works for TCGAN, using the same technique as CVAE but applied to a GAN. I did not have the expected success with this variant (see section 4.1.3).

My second objective was to investigate the impact of Differential Privacy on VAE capabilities. I planned a supervised approach on synthetic datasets from Creditcards and Census with anomaly detection and classification, respectively. Due to time and performance constraints on the synthetic data, I replaced these two synthetic datasets with the Churn and MNIST datasets, which also perform anomaly detection and classification, respectively. Churn was used to study the impact of DP-SGD on a model, and MNIST was used to study the training time/noise ratio required for DP-SGD.

3.2.2 Choice of evaluation metrics

3.2.2.1 VAE performance evaluation

In assessing generative performance, I recognised that this aspect does not necessarily correlate with the model's reconstruction performance, measured during training through an LSE-based loss. Therefore, it was imperative to select metrics that provided insight into the generative capabilities of the trained model. In the literature, negative or simple log-likelihood is often the preferred metric [25, 28, 29]. However, its interpretability is limited as it does not allow a nuanced understanding of the discrepancy between real and synthetic data. To overcome this problem and add interpretability to my results, I have chosen the four metrics used by the Synthetic Data Vault to evaluate single tables. These metrics include :

The quality of column representation

The quality of column representation measures the similarity of the generated columns to their original version. KS complement is used for numerical data, and TV complement for categorical data [80].

KS Complement

This test is the complement of the statistic derived from the Kolmogorov-Smirnov test. It is obtained by transforming the numerical distribution into a cumulative distribution function and then looking at the maximum deviation between the real and synthetic data curves.

TV Complement

"This test calculates the total variation distance (TVD) between the real and synthetic columns. To do this, it first calculates the frequency of each category value and expresses it as a probability. The TVD statistic compares the differences in probabilities, as shown in the formula below:

$$\delta(R, S) = \frac{1}{2} \sum_{\omega \in \Omega} |R_\omega - S_\omega| \tag{3.1}$$

Here, ω describes all the possible categories in a column, Ω. At the same time, R and S refer to the actual and synthetic frequencies of these categories. TVComplement returns 1 - TVD, so a higher score means better resemblance.

$$score = 1 - \delta(R, S) \tag{3.2}$$

The quality of covariances

To check the quality of the reproduction of the links between the columns, I measure the similarity of the covariance matrices. To do this, I measure the Correlation Similarity when working between two numerical columns and the Contingency Similarity between two categorical columns. Between a numerical column and a categorical column, I discretise the numerical column into two columns and then perform a Contingency Similarity.

Contingency Similarity

For a pair of columns, A and B, the test calculates a standardised contingency table for the real and synthetic data. This table describes the proportion of rows with each combination of categories in A and B.

It then calculates the difference between the contingency tables using the total variation distance. Finally, I subtract the distance from 1 to ensure that a high score means high similarity. The process is summarised by the formula below.

$$score = 1 - \frac{1}{2} \sum_{\alpha \in A} \sum_{\beta \in B} |S_{\alpha,\beta} - R_{\alpha,\beta}| \tag{3.3}$$

In the formula, α describes all possible categories in column A, and β describes all possible categories in column B. Furthermore, R and S refer to these categories' actual and synthetic frequencies.

Correlation Similarity

For a pair of columns, A and B, this test calculates a correlation coefficient on the real and synthetic data, R and S. The result is two distinct correlation values. The test normalises and returns a similarity score using the formula below.

$$score = 1 - \frac{|S_{A,B} - R_{A,B}|}{2} \tag{3.4}$$

Note that there are several ways of calculating the correlation coefficient. The test supports Pearson's correlation coefficient [1][2] and Spearman's rank correlation coefficient [3][4]. Both are between -1 and +1.

The quality of recovery

I measure whether the percentage of the values in the original data which are also present in the synthetic data.

"If r and s represent the real and synthetic columns, this measure calculates how close the min and max s values are to the real min and max r values using the formula below.

$$score = 1 - \left[\max\left(\frac{\min(s) - \min(r)}{\max(r) - \min(r)}, 0 \right) + \max\left(\frac{\max(r) - \max(s)}{\max(r) - \min(r)}, 0 \right) \right] \tag{3.5}$$

The above equation may become negative if the synthetic data has deficient coverage. In this case, I indicate a score of 0 as this is the lowest possible value.

Note that the score is not penalised if the synthetic data falls outside the limits. Suppose the synthetic data exceeds the actual min and max values. In that case, the range is fully covered, and the score will be 1."

Respecting borders

It measures the extent to which the synthetic data exceeds the extremes of the real data values. The score is the percentage of data between the minimum and maximum value of the range of real data being imitated.

Overall score

Each metric provides a score between 0 and 1, culminating in an overall score between 0 and 1. These metrics, more interpretable and informative than negative log-likelihood, provide a clearer understanding of the VAE's strengths and weaknesses in synthetic data generation.

3.2.2.2 Classification performance evaluation

To assess the capabilities of a classification model, we use its precision, recall and F1-score, which are the metrics most commonly used in this type of task. They can be applied to a binary classification task, such as the Churn Modelling dataset, or multi-class classification tasks, such as MNIST.

Precision

Precision is defined as follows:

$Precision = \frac{true\ positives}{true\ positives + false\ positives}$

It represents the ability of a model to find all positive cases in the evaluated dataset without classification error.

Recall

Recall is defined as follows:

$Recall = \frac{true\ positives}{true\ positives + false\ negatives}$

It represents the ability of a model to find all positive cases in the evaluated dataset.

F1-score

F1-score is defined as follows:

$F1-score = \frac{2}{\frac{1}{precision} + \frac{1}{recall}}$

It is the harmonic mean between the two other scores. Maximising the F1-score means finding the optimal balance between precision and recall.

3.3 Experimental protocol

3.3.1 Improving the VAE

In my exploration of the generative capabilities of the VAE, I prioritised understanding the impact of each enhancement under controlled conditions.

Except where specifically mentioned, each experiment changed only a single parameter to ensure clarity in assessing the influence of that parameter.

I used the following standardised experimental setup:

- **Optimiser:** Adam, with a learning rate of 1e-3, was used throughout. A lower learning rate resulted in insufficient model learning, while a higher rate resulted in training instability.
- **Regularisation:** I used a small weight-decay of 1e-5, a technique that effectively mitigated overfitting, which was particularly evident when training the TVAE and contributed to training stability. The weight decay is also used by the SDV [81].
- **Training duration:** Each model was trained for 100 epochs, balancing computational constraints with sufficient duration to observe minimal effects on the training process.

I also kept a consistent model architecture:

- **Gumbel-Max trick:** I used the Gumbel Max trick for most experiments except the first two with a temperature of 0.35 for output neurons generating categorical columns. This choice of temperature in the experiments was in contrast to the optimal temperature of 3.85 observed in the Berka dataset. This discrepancy results from human error in the automation of the experimental process. However, this discrepancy did not affect the experimental conclusions.
- **Model architecture:** With few exceptions, the architecture of Xu et al. [20] was retained, with an encoder and decoder, each with two dense layers of 128 neurons and a latent space of size 128. Unless specified otherwise, I used he_uniform initialisation and ReLu activation for all dense layers and output layers except the latent space, where I used a random uniform.

I ensured the validity and robustness of my results through the use of multiple data sets and a rigorous evaluation protocol:

- **Datasets:** Experiments were conducted using the Berka, Creditcards, and Census datasets. Initial testing was conducted on Berka, followed by validation on the other datasets, except for the VAE variants experiment (see section 9 of the Appendices), which was limited to Berka due to time constraints.

- **Generative performance evaluation:** I evaluated the generative performance of the TVAE at two points in time - in the format required by the TVAE and in the original data format. This dual evaluation allowed us to assess the effectiveness of the TVAE in reproducing data and the impact of data transformation. The latter was prioritised for results and conclusion, as the synthetic data would ultimately be used in its original form. For each assessment, a sample was generated ten times and compared to the test data, with the results representing the average of these tests. The standard deviation observed was typically low, around 1e-4, in scenarios where the variation in results was around 1e-3.

- **Statistical analysis:** I supplemented my analysis of results with a statistical investigation, performing an Analysis of Variance (ANOVA) for each experiment on each dataset and further substantiating this with a Student's T-test or if we have more than two distributions to compare, Tukey's test to identify significant results.

3.3.2 Differential privacy

In the case of DP, I have reimplemented DP-SGD because TensorFlow privacy is still in transition between Tensorflow 1 and Tensorflow 2 [49]. As a result, it is not yet compatible with custom models like my TVAE implementation. To simplify the calculation of DP parameter values, I used the functions available in Google Differential Privacy to calculate the associated ϵ privacy cost value for a given noise [50]. However, instead of training the model on a given noise and calculating a privacy cost that differs from the desired cost, I interpolated the noise value to be added from different ϵ values. With this technique, I could choose a confidentiality cost and the constraint associated before training the model.

I used a dense model with three dense layers of 12, 8 and 8 neurons for anomaly detection on the churn dataset, with a LeakyRectified Linear Unit (ReLU) and Batch Normalization (BN). The final activation function is sigmoid. The optimiser is Adam in the standard version of Tensorflow, and the loss used is binary_crossentropy.

For classification on the MNIST dataset, I used a convolutional model with the following architecture

- Conv2D: 32 neurons, kernel (5.5), padding = same and activation=ReLU

- Conv2D: 32 neurons, kernel (5.5), padding = equal and activation=ReLU
- Max pooling 2D (if used in the experimental protocol: Dropout (0.1))
- Conv2D: 64 neurons, kernel (3.3), padding = equal and activation=ReLU
- Conv2D: 64 neurons, kernel (3.3), padding = equal and activation=ReLU
- Max pooling 2D (If used in the experimental protocol: Dropout (0.1))
- Flatten
- Dense: 128 neurons and activation=ReLU (If used in the experimental protocol: DO (0.1))
- Dense output: 10 neurons, softmax

The optimiser is Adam in the standard version of Tensorflow, and the loss used is categorical_crossentropy.

Results were verified using a 10-fold cross-validation for Churn Modelling and MNIST.

3.3.3 CVAE for MNIST

An experiment was made to check that the CVAE, the various KL warm-up techniques, the B disentanglement and the Deep Feature Consistent Variational Auto Encoder (DFC-VAE) were working correctly. The DFC-VAE is a modification of the VAE that replaces the usual reconstruction loss (MSE or probability) with a perceptual loss [57]. This perceptual loss uses a second multi-layer convolutional neural network. Unlike the conventional reconstruction loss, which measures the difference between the input and output images, the perceptual loss compares the representation of the input and output images layer by layer. This loss aims to do better than a pixel-by-pixel comparison by comparing the essential elements of each image instead.

Therefore, I needed to train two models for this experiment: The first is the classification model, identical to the classification model on MNIST described in the previous section. The difference lies in the learning rate and the use of schedulers. This model was trained with a Cyclical Learning Rate (CLR) whose minimum and maximum values are 1e-4 and 1e-2, respectively, and the

period is 8 epochs. The model was trained until early stopping, after more than 200 epochs.

The CVAE used here uses a convolutional encoder and decoder. The encoder takes two inputs, conditional and image data, and processes them through convolutional layers to generate a latent space's mean and log variance. These two latent vectors are then used to sample from the latent space. The decoder, taking the sampled latent vector and the conditional data as inputs, uses transpose convolutional layers to reconstruct the original image data.

- Conv2D: 32 neurons, kernel : (3), strides : (2,2) and activation=ReLU
- Conv2D: 64 neurons, kernel : (3), strides : (2,2) and activation=ReLU
- Flatten
- Concatenate (Flatten + category)
- 2 * Dense: 2, activation = Linear. This layer is for the latent space
- Concatenate (Latent space + category)
- Dense: 1568 (7*7*32), activation = ReLU
- Conv2D transpose: 64 neurons, kernel : (3), strides: 2, padding: same and activation=ReLU
- Conv2D transpose: 32 neurons, kernel : (3), strides: 2, padding: same and activation=ReLU
- Conv2D transpose: 1 neuron, kernel : (3), strides: 1, padding: same and activation=ReLU

Chapter 4

Results

4.1 VAE improvements

4.1.1 Architecture

The first objective of this Master thesis is to improve Tabular Variational Auto Encoder (TVAE) by checking whether the advances made in multidimensional VAEs (for image or sound) can be transferred to unidimensional VAEs such as TVAE. This verification was done by testing whether some simple improvements to the TVAE architecture would work adequately. The TVAE architecture in [20] consists only of dense layers, as explained in part 3.3.1, and the scientific community still needs to take the time to improve it. Similarly, Modeling Tabular Data using Conditional GAN [20] introduces the TCGAN, for which it uses the "Gumbel Max trick" activation function as an output, which is known to improve the performance of deep neural networks with categorical data (see part 2.2) [42, 75], but does not use it for TVAE without explaining why. These are all avenues I have followed to study the impact on the generative capabilities of the TVAE.

According to the metrics presented in part 3.2.2.1, the TVAE achieves an overall score of 0.965 on Berka, 0.947 on Census, and 0.89 on Creditcards, while the VAE achieves only 0.957, 0.779, and 0.83 on these respective datasets. The VAE needs to be trained for 300 epochs on Berka to perform similarly to the TVAE in 100 epochs on the same dataset. The TVAE performs better than the VAE on column similarity and data correlation. Conversely, the VAE is more effective at covering the range of data and, depending on the dataset, respecting the limits of that coverage. For example, for Creditcards, the TVAE scores are 0.988 for data similarity, 0.99 for correlation, 0.999

for limit compliance, and 0.6 for coverage. VAE scores only 0.938 for data similarity, 0.381 for correlation, 0.99 for limit compliance, and 1 for coverage. This difference is significant according to a Student T-test run on ten generated batches of synthetic data under the dummy form for each dataset, with a p-value inferior to 0.01, as explained in section 3.3.1. TVAE can produce data that correctly models a subset of the dataset, whereas VAE produces a relatively rough approximation of the dataset as a whole. These failures bring us back to two well-known weaknesses of VAE: blurriness and the lower correlations, which TVAE partially eliminates, sometimes at the cost of lower data coverage. The following improvements do not resolve this last point. The results of these studies can be found in the appendices. This performance difference, as well as the commercial use of the TVAE [80], convinced us to concentrate on the latter for the remainder of the experiments. I then looked at the initialisation functions of the different TVAE layers.

For dense layers with a ReLU activation function, the he_uniform function is the most suitable (see Figure 7), while for latent space initialisation, random_uniform is the most efficient for both the mean and the log variance of the latent space distribution (see also Figure 7). For the initialisation of the output layer, the log variance is initialised by a normal distribution centred at log(0.1), a value suggested by the SDV [81] and confirmed by tests. For the mean, I chose he_uniform because of its performance and stability. The case of the zero initialisation function is noteworthy for both the latent space and the output layer because, although its performance is excellent on Berka (0.968 compared to 0.965 for he_uniform), this function is accompanied by an increasingly large overfit as training progresses, which degrades its performance on, for example, credit cards (0.832 compared to 0.89 for he_uniform).

Different loss functions were also tested on the TVAE (see section 1.2 and 1.3 of the appendices for results). The loss proposed by the SDV (see part 2.1.2.4 in the TVAE section) gave the best results on the Berka dataset (0.965 compared to values below 0.85), and I kept this loss for the rest of the experiments. The other losses eventually stagnate as the KL divergence approaches 0 (KL collapse) and cannot produce credible numerical data, such as the MSE variants, or are less stable and less efficient (Log Likelihood loss) (see figure 8). The distribution of the data generated can be seen in figure 9.

The following test concerns the use of Softmax or the Gumbel max trick as the output function for categorical data. The Gumbel max trick is better than the Softmax function when the temperature of the Gumbel max trick is adjusted (at a temperature of 3.85, the Gumbel max trick gives 0.967 on Berka

compared to 0.965 for the Softmax function). Otherwise, the Softmax function is superior, with a score of 0.965 versus 0.962 for the Gumbel max trick at a temperature of 0.35. Again, this was tested with a Student's T-test with ten samples of synthetic data compared to the dummy version of each dataset, with a p-value of about 1e-15, far below 0.01. When the temperature of the Gumbel max trick is low, the generated distribution is closer to a One Hot encoding. In contrast, when the temperature is high, I am closer to a form of label smoothing, a process known to improve the performance of classification models by reducing the confidence in the generated responses [87]. Thus, the generative performance of TVAE is enhanced when combined with some form of label smoothing, such as the Gumbel Max trick, at high temperatures. In subsequent experiments, I used the Gumbel Max trick (see section 3.3.1). However, due to human error, I use the low-temperature version of the Gumbel Max trick. This error does not change the conclusions, but the baseline models perform slightly less.

Despite its performance, the SDV loss causes, at best, a slight instability during training and, at worst, an overfitting of the model (see Figure 10). To counteract this phenomenon, I studied functions known for their positive effects on VAE [32] for combating overfit and stabilising training: BN and DO. These two functions have very positive and comparable effects, which are reflected in the performance of the TVAE and the metrics of the SDV. I obtained 0.968 on Berka, 0.943 on Census, and 0.879 on Creditcards, compared to baselines of 0.962 on Berka, 0.935 on Census, and 0.864 on Creditcards in the case of BN, and 0.966, 0.944 and 0.892 in the case of DO. These performances reflect the stability brought about by these improvements and the disappearance of categories without representatives, which could previously be generated regardless of the amount of data generated. DO is accompanied by a pronounced underfitting phenomenon, regardless of the DO value (see Figure 11). BN does not suffer from this problem. Combining the two techniques slightly improves the results compared to DO alone, but not significantly (p-value > 0.05 for a Student's T-test as explained in 3.3.1).

I also wanted to compare alternatives to the ReLU function used in the models from Modeling Tabular Data using Conditional GAN [20]. I, therefore, examined LeakyReLU and Scaled Exponential Linear (SELU). The performance gain is unclear. LeakyReLU improves the three datasets according to the SDV metrics (0.967 vs. 0.962, 0.937 vs. 0.935 and 0.869 vs. 0.864, respectively). However, these results are not significant on Census, and the models suffer from very pronounced overfitting problems and constant column generation (see Figure 12). SELU worsens the performance of TVAE

Creditcards (0.844 vs. 0.864) and suffers from the same overfitting and generation problems as LeakyReLU. These shortcomings are corrected by DO, which achieves excellent performance on Creditcards (SELU + DO: 0.893, Leaky ReLU + BN + DO: 0.889). The regularisation capabilities of DO and BN are strong enough to compensate for the increased capabilities of the model with these activation functions.

Reducing the size of the latent space had a positive effect on all the datasets. The best results for Census and Creditcards were obtained with a latent dimension of 8 (with 0.969 and 0.889, respectively) and a size of 64 instead of 128 for Berka (0.968). Increasing the size of the encoder and decoder layers improves the generative performance. However, it is accompanied by much more severe overfitting problems and is within the performance of a dimension reduction. These results are consistent with the fact that I already encountered overfitting problems with the initial model and that increasing the number of neurons exacerbates the problem. Note that using DO in addition to latent dimension reduction significantly improves the results: 0.893 for credit cards with a latent dimension of 8 and DO, compared to 0.889 for the same model on the same dataset without DO, with a p-value around e1-15 for a Tukey's T-test realised in the conditions described in 3.3.1.

On the other hand, combining an increase in encoder and decoder size with a decrease in latent space size only sometimes has a significant positive effect. This increase in size affects the generative score on the Berka and Census datasets (0.965 instead of 0.968 and 0.941 instead of 0.969, respectively) for models of size 256-128-64-128-256 for Berka and 256-128-8-128-256 for Census. On the other hand, I get a score of 0.901, the best score for Creditcards, with a 256-128-2-128-256 model. A 256-128-2-128-256 model represents an architecture where the encoder consists of dense layers with 256 and 128 neurons. This configuration leads to a latent space with a dimensionality of 2, followed by a decoder with dense layers of 128 and 256 neurons. In summary, while it is better to favour a small latent space with TVAE, increasing the size of the encoder and decoder is a decision that must be made based on the complexity of the data set. The more complex the data set (but not necessarily the larger it is), the more the model could benefit from balancing encoder and decoder growth with a reduction in the size of the latent space.

As I have seen, the TVAE is a powerful model that overfits quickly. Any techniques that improve its performance must be used simultaneously as regularisation techniques, such as latent dimension reduction, DO, BN, or weight decay, which is used in these experiments and by the SDV (see section

3.3.1).

4.1.2 Schedulers

I also investigated VAE-specific techniques, such as schedulers. In particular, I studied KL Warm-up, a linear or cyclic variation in the weighting of the KL divergence during training, and β Disentanglement, which forces the KL divergence to take specific values to relax the constraint it imposes on training [32, 33](see section 2.1.2.1). These two techniques have been implemented with BN (as in Ladder Variational Autoencoders [32]) and without BN. Both methods improve the TVAE results on all datasets. However, where KL warm-up does not significantly differ from BN or DO alone on Berka and Census, B disentanglement is significantly less effective than these three techniques, even after searching for hyperparameters to improve its results. I ensured the significance of the results using a Tukey's test on the different combinations possible on each dataset, with ten samples for each combination on each dataset. The p-values are all under 0.01. I tested four ways of implementing the KL warm-up: first by epoch without BN and by epoch with BN, without any significant effect on Berka and Creditcards, then by batch with BN and by cycle with BN, as in the [32]. The KL warm-up by batch with BN proved to be the best, giving an overall score of 0.968 for Berka, 0.943 for Census, and 0.879 for Creditcards. In contrast, the B disentanglement, without BN, gave at best 0.967 on Berka, 0.936 on Census, and 0.874 on Creditcards, as BN adversely affected its performance. However, KL warm-up without BN does not prevent generation problems associated with constant columns, and B-disentanglement can lead to cases of KL collapse.

To confirm my implementation of KL warm-up and β Disentanglement, I also took the time to test these techniques in 2D with a CVAE on the MNIST dataset (see Figure 13 and section 3.3.3). The KL warm-up improves the sharpness of the generated images, and the B disentanglement produces even sharper images. Nevertheless, they are distorted and incompatible with the control of the generated class. On this last point, the cyclic KL warm-up makes it possible to maintain this control, whereas the linear KL warm-up only sometimes manages to do so after 100 epochs.

These tests confirm that these VAE-specific improvements, tested in 2D, can be successfully applied to the TVAE. However, only the batch KL warm-up or the cyclic KL warm-up with BN are viable options, both for their performance and for their possible use in conjunction with VAE variants such as the CVAE (as demonstrated in this thesis) or the Ladder Variational Auto

Encoder (LVAE) [32].

4.1.3 Variants

Most of my work on improvements to TVAE has focused on variants of VAE that address some of the shortcomings of the original model (see section 2.1). I tested different architectures for the TVAE, adapted from the following VAE variants: Info-VAE, IWAE, MIWAE, CIWAE, PIWAE, SVAE, VAE-GAN, VQ-VAE, CVAE. However, the VQ-VAE was quickly dropped due to its instability during training (collapse of the latent space on a single point), and an error in the weighting of the logarithmic variance of the loss prevented us from including the MIWAE and PIWAE in the final results. Except for the VQ-VAE, VAE-GAN, and SVAE, the other variants significantly improve the performance of the TVAE: compared with the Berka baseline of 0.962, the IWAE scores are 0.948, the CIWAE scores are 0.969, the CVAE scores are 0.938, and the info VAE scores are 0.927. However, under ideal conditions for the TVAE on the Berka dataset, with a latent dimension of 64, DO, and BN, this model performs better than these variants with the same adjustments. The IWAE scores 0.969, the CIWAE 0.968, the CVAE 0.967, and the info VAE 0.967, which is not significantly different from the classic TVAE, which scores 0.969. This increase in score can be explained by the fact that the improvements made to the TVAE are aimed at combating overfitting, but only on variants that are more efficient and less prone to overfitting and, therefore, benefit less from regularisation techniques.

The inability of the CVAE to retain its ability to control the category of data generated, regardless of latent dimension or training time, is a missed opportunity for the TVAE, as the CVAE can generate precisely the data needed rather than generating data until the required data appears. The training-by-sampling method is successfully used in [20] to train the TCGAN, as is the formatting of the data for the TVAE. All that remains is to use an output layer with log variance by batch rather than by item to explain the incompatibility of the TVAE with the CVAE.

Overall, the variations of the VAE in 2D are applied to the TVAE with some success, and this result opens the door to many possible improvements to the TVAE.

4.1.4 Pre-processing and post-processing

One area of work that was not initially envisaged but became apparent as the experiments progressed was the data processing required to enable the data to be processed by the TVAE and then returned to its original form. The complete process is explained in section 3.1.1 and is copied from that of the SDV. However, this preparation process often fails, making it impossible for the TVAE to reproduce the initial distribution. This case of failure is illustrated in Figure 14 on the Census dataset, where the transformed numerical column does not follow a Gaussian distribution. To reduce the risk of failure, I added a Yeo-Johnson transformation before the standard transformation and another after it. This modification effectively eliminated the errors for Creditcards and Berka, but not for Census. The benefits can be seen when I compare the data not in their original form but in their generated form, where I obtain 0.974 for Creditcards with DO and BN and 0.966 for Berka under the same conditions. However, for Census with DO and BN, I obtained only 0.91 at most. These performances are independent of the number of columns of the data set to be reproduced since Creditcards has almost 300, Berka 66, and Census almost 500. For Creditcards, I see an improvement from 0.9 without the Yeo-Johnson transformations to 0.977 with them.

This work on pre-processing must also be accompanied by work on post-processing, which can severely degrade model performance. Currently, this performance of 0.977 generation score according to SDV metrics, the best score obtained across all datasets, is not repeated after post-processing when the score is 0.88, the worst score of the three datasets. Worse still, in a practical case, the data generated this way is unusable (see below).

Faced with this transformation problem and the inability of the TVAE to correctly model the categories on Berka (maximum similarity score of 0.83 for the categorical data), I tried to add a new transformation to have only numerical data. I, therefore, added a Uniform Manifold Approximation and Projection (UMAP) transformation in 2, 3, 8, or 16 dimensions after formatting the Berka data (see section 3.1.1). The performance obtained was encouraging (0.97 points in 2D, 0.988 in 16D) when the synthetic data were compared with their real equivalent in the form reduced by UMAP. However, the reverse transformation proved difficult or even impossible. Only the 2D version could benefit from the reverse UMAP transformation. All other versions were too large and caused the hardware to crash. After this transformation, the score of the synthetic data was only 0.88, well below the usual performance of TVAE on Berka. There is too much loss of information in the UMAP

transformation for such a significant reduction in dimension (66 to 2), both in the transformation and in its inverse.

4.1.5 Synthetic data quality

The second objective of this Master Thesis was to evaluate the models obtained by

1. The SDV metrics
2. The evolution of the performance of the machine learning model trained on the synthetic data generated
3. The guarantees of confidentiality of the data generated

In previous experiments, I have addressed the first part of this objective. I wanted to address the second part by studying the case of Creditcards. Creditcards is a banking data set for anomaly detection (see section 3.1). The model trained on the real data obtained an F1 score of 0.899 in 100 epochs on the real test data (see section 3.2.2.2 for the metrics used in this section). Its performance deteriorated significantly on the synthetic test data, with an F1 score of 0.446. Almost 20% of the synthetic data was classified as anomalous, illustrating the gap between real and synthetic data. A model trained on the synthetic data only achieved an F1 score of 0.52, showing that the synthetic data generated in this case is not only very different from the real data but also unsuitable for this use.

The discrepancy between the real data and the synthetic data reflects the poor performance of TVAE-GAN. Even with an overall score of 0.958 on the Berka dataset, the loss of the discriminator was between 0.43 and 0.75 depending on the real, reconstructed, or synthetic data, compared to between 6 and 12 for the corresponding part of the generator loss (see section 2.1.2.3), which means that the discriminator succeeds in correctly identifying the origin of the data most of the time.

In retrospect, the choice of task was not the wisest: anomaly detection is a difficult task, complicated by the inherent imbalance between average data and anomalies, which is exacerbated by the generation of synthetic data [88]. As a result, the model trained on synthetic data contains half as many anomalies as the real data (252 compared to 492 anomalies in the real data set, i.e., 0.138% compared to 0.27% of the total data). Creditcards is also the test where the TVAE performs the worst. The maximum overall score for the data in their original form is 0.9, a low score mainly due to the coverage of

values, where less than two-thirds (0.655) of the values in the real data are represented in the synthetic dataset. Moreover, despite a correlation score of 0.986, the correlations are not realistic, as illustrated by Figure 15.

4.2 Differential privacy applicability

The third part of the second objective was to ensure the confidentiality of the data generated. As mentioned in section 3.2.2.1, I did not find a satisfactory metric to measure this confidentiality. For example, neither the SDV nor the [75] provide a more convincing metric than calculating the average distance between the real data and the synthetic data or the number of rows common to both data sets. Differential Privacy is a simple technique that provides guarantees of confidentiality and, therefore, makes it possible to achieve this second goal. However, Differential Privacy comes at a cost in terms of performance, and I wanted to study it to see under what conditions it could be used in a practical context. However, the failure to apply synthetic data to Creditcards and the lack of time led us to scale back my ambitions and not link TVAE and Differential Privacy.

However, I could still study the variation in performance as a function of the increased confidentiality constraints imposed (see sections 2.3 and 3.2.2). I saw the emergence of a plateau between two clear breaks in performance. The first break occurs when using differential Privacy (ϵ = 10), where I go from 85% accuracy in classification to 80% (cross-validation, in 10 parts). As section 2.3.4 explains, the DP-SGD used to train the model clips the gradient and adds noise. Clipping the gradient reduces the training speed of the model, which degrades its performance. This decrease in accuracy is followed by a plateau where the model's performance does not change regardless of the increase in confidentiality constraints. The noise gets louder and louder, but the model remains powerful enough to overcome it. The second break occurs at ϵ = 1, where the accuracy drops from 79% to 0.77% and even lower. This break is the point at which the noise exceeds the model's capabilities and begins to degrade the quality of the training. I also performed these experiments with a DO. I saw an increase in performance, as well as a shift in the second break from ϵ = 1 to ϵ = 0.7, making the DO an almost mandatory improvement when using the DP-SGD in a practical case.

The second experiment was motivated by the fact that current implementations of the DP-SGD require noise and sensitivity and then give a level of confidentiality that may prove insufficient after training. I wanted to take the opposite approach, taking advantage of the ability to do the calculations before

training to determine, for a given sensitivity and level of confidentiality, the noise to be added as a function of training time (see section 3.2.2.2). Since longer training means better performance but also more noise added, which means worse performance, I wanted to find the balance between better training and sufficient but not problematic noise. I set ϵ to 1. Without a DO, any increase in training time degrades the model's performance. With a DO, I show that performance improves during the first 40 epochs before slowly declining from the 40-60 epoch interval. This experiment reinforces the notion of a noise limit before performance degradation, as highlighted in the previous experiment. Thus, there is a balance to be struck between the desired privacy constraint and the number of training epochs required by the model, and setting one leads to the existence of a limit for the other. In the case of TVAE, where the number of epochs can be as low as 30 without affecting performances too much and which benefits from regularisation techniques, using DP-SGD seems possible.

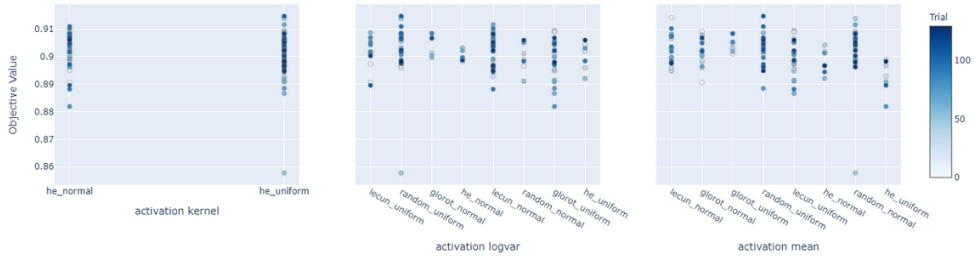


Figure 4.1: Overall score on data formatted for TVAE on the Berka dataset. Activation kernel is the initialisation function for the dense layers of the encoder and decoder, activation log var for the initialisation function of the log variance of the latent space, and activation mean for the initialisation of the latent space. Obtained with Optuna during this master thesis.

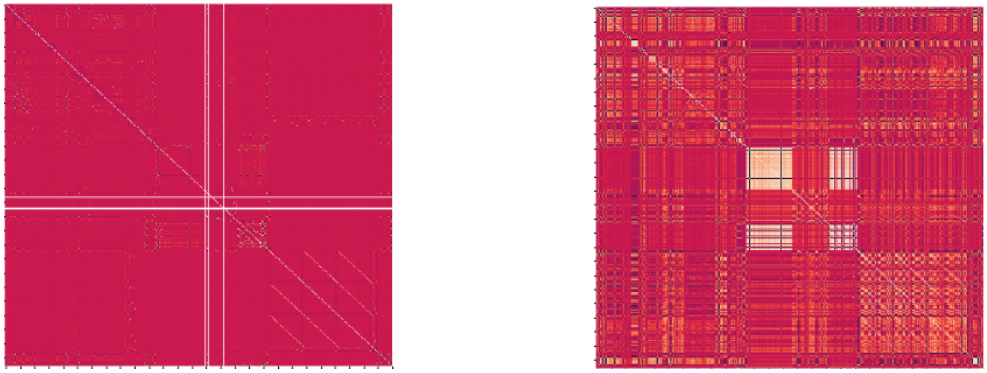


Figure 4.2: Census real correlation matrix on the left, and data generated with the ECB on the right Obtained in Experiment 1 of this Master thesis.

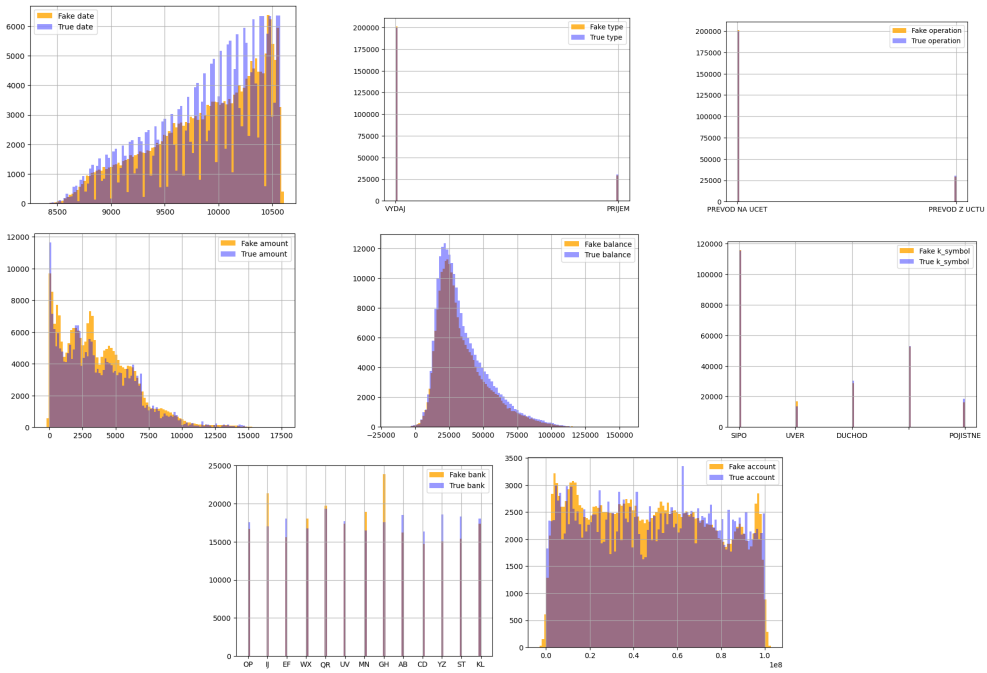


Figure 4.3: Example of the distribution of the data generated for the Berka dataset obtained with the TVAE. Note the difficulty the model has in mimicking the distribution peaks on the numerical data, despite having only 100 training epochs, and the slight discrepancies between the number of real elements in a given category and the number of synthetic elements. Obtained in Experiment 1 of this Master thesis.

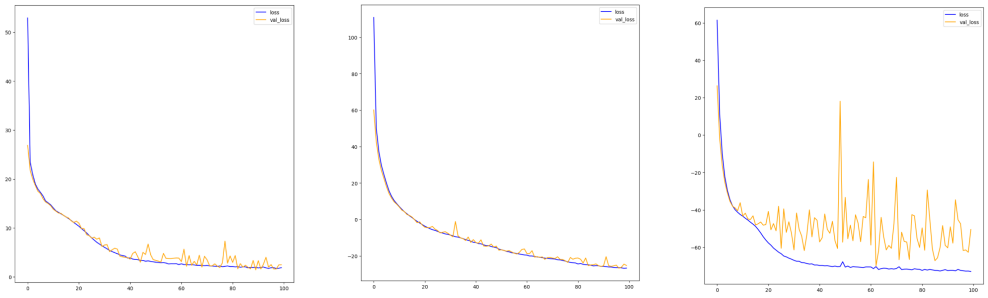


Figure 4.4: Training curve of the TVAE with no improvement, using the softmax activation for categorical data. From left to right: Berka Czech, Census, Creditcards. Obtained in Experiment 1 of this Master thesis.

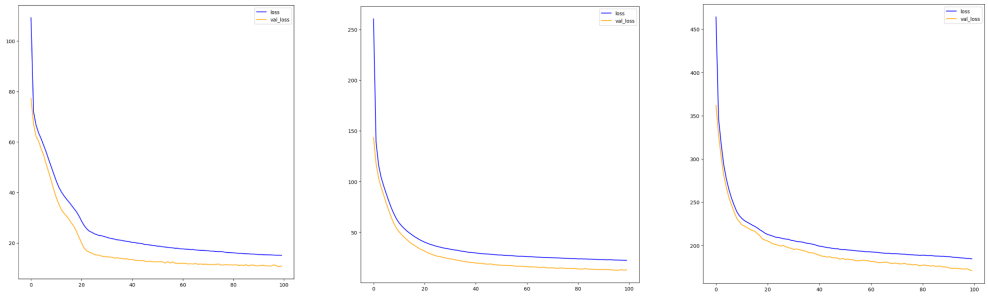


Figure 4.5: Training curve of the TVAE with DO (0.05) and BN, using the Gumbel Max trick activation for categorical data. From left to right: Berka Czech, Census, Creditcards. Obtained in Experiment 1 of this Master thesis.

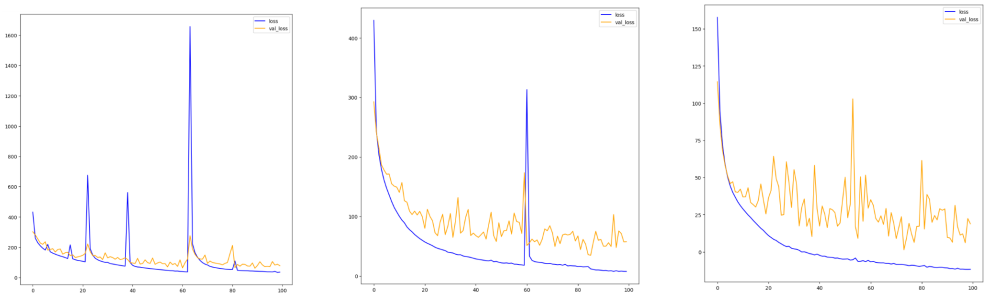
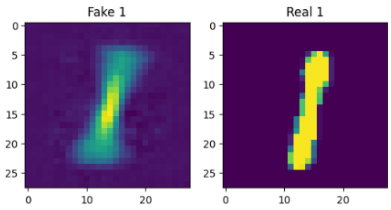
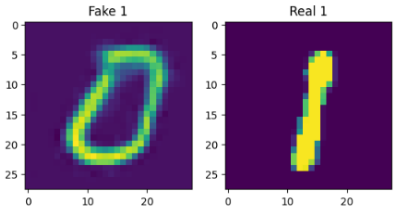


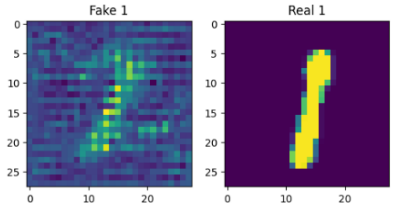
Figure 4.6: TVAE training curves on Creditcards. Left Leaky ReLU, centre SELU, right ReLU. . Obtained in Experiment 1 of this Master thesis.



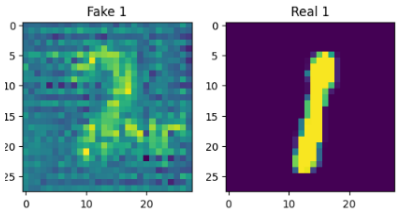
(a) without any enhancement



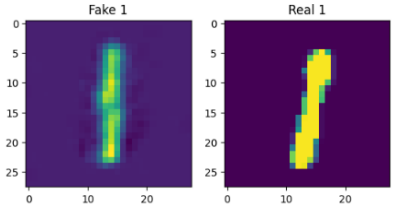
(b) using B disentanglement



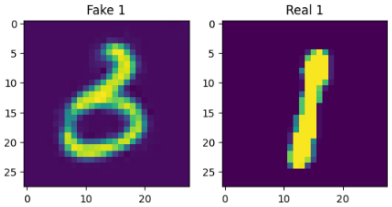
(c) in combination with the DFC VAE



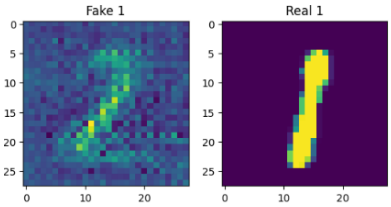
(d) in combination with the DFC VAE, and the B disentanglement



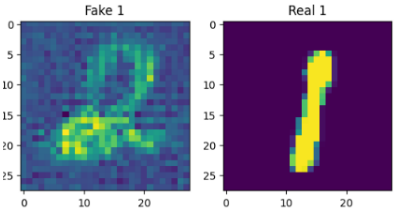
(e) using the KL warm up



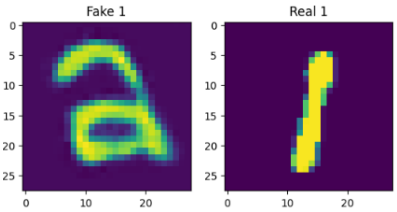
(f) using the KL warm up and the B disentanglement at the same time



(g) using the cyclical KL warm up for 2 cycles and DFC-VAE loss



(h) using the cyclical KL warm up for 2 cycles, B disentanglement and DFC-VAE loss



(i) Using the cyclical KL warm up for 10 cycles (300 epochs)

Figure 4.7: Examples of numbers generated by the CVAE.The real images are on the right, and the generated images are on the left.

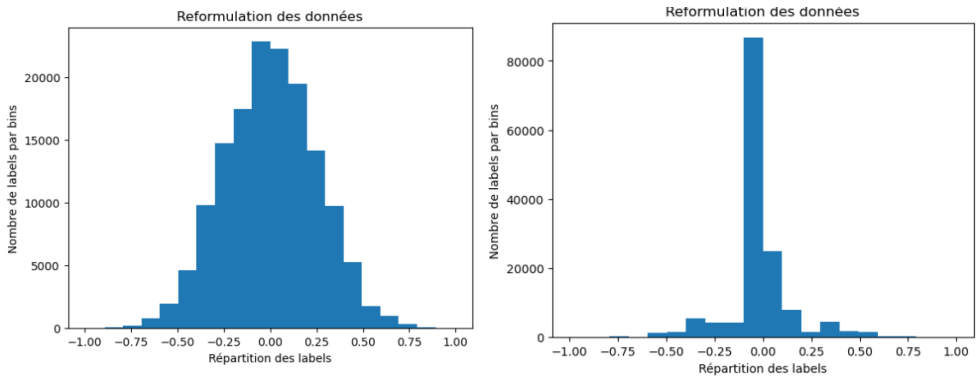


Figure 4.8: Instance of pre-processing success (left, column 73), and failure (right, column 0) on the dataset Census. . Obtained in pre-processing of this Master thesis.

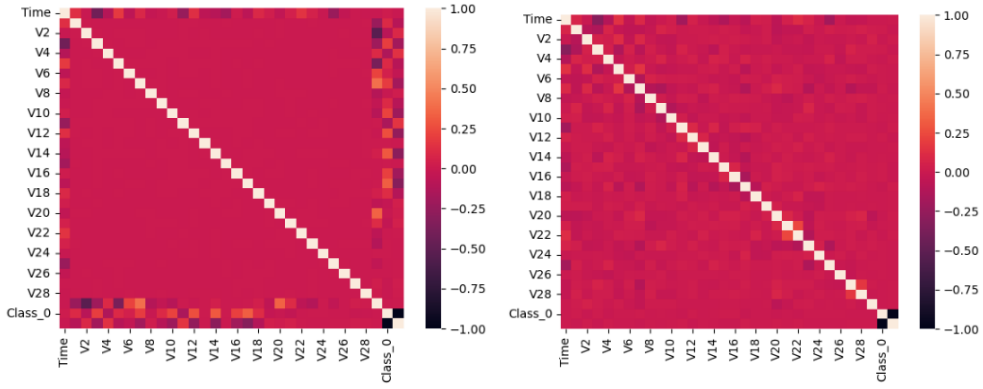


Figure 4.9: Left: correlation matrix for real Creditcards data. On the right, its synthetic counterpart (obtained with a TVAE trained with LEaky ReLU, DO and BN). . Obtained in Experiment 6 of this Master thesis.

Chapter 5

Discussion

5.1 Results

TVAE was first presented in Modeling Tabular Data using Conditional GAN [20] and consisted only of 5 dense layers of 128 neurons, plus unique data shaping and a special output layer. Even today, this is the form in which it is most widely used [80]. In this Master Thesis, I have studied the strengths and weaknesses of the TVAE. I have shown that TVAE is effective and generalisable, confirming the conclusion from Unlocking the Potential of Synthetic Tabular Data Generation with Variational Autoencoders [75]. However, it is prone to overfitting. It, therefore, benefits significantly from regularisation techniques such as BN or DO and can benefit from other improvements such as more powerful activation functions or larger encoders and decoders (see section 4.1.1). More importantly, TVAE can be combined with several variants of VAE in 2D and benefits from KL warm-up, opening the way to many other techniques that could overcome the limitations of the current model, which can be expected from synthetic data in general [88], but also specific to TVAE, such as the shaping of the associated data (sections 4.1.2, 4.1.3 and 4.1.4).

DP has been studied, and I have highlighted a noise limit that depends on the model's capabilities, making it possible to determine a balance between constraints and training time.

5.2 Limitations of the methods

The main limitation was the short training time of each experiment, 100 epochs. Teams behind publications such as in Ladder Variational

Autoencoders [32] train their models for 2000 epochs and then fine-tune them for 2000 additional epochs. Due to time and resource constraints, I worked under different conditions (see section 3.3), so my conclusions are based on models that may have behaved differently in the long run. Another limitation of my methods was the exclusive use of SDV metrics. These have the advantage of being multi-criteria, but their reliability can be questioned if a model produces data with a score of 0.9 according to these metrics without being usable. A more rigorous assessment would be more informative. Finally, I spent much time on the ablation trials. In retrospect, I should have used this time to perform more essential experiments, such as applying the synthetic census data to a classification case.

5.3 Industrial impact

So, there is still a long way to go to produce synthetic data indistinguishable from the real thing. However, the improvements in results are encouraging, and I can now envisage a more efficient variant of the TVAE at an affordable cost of operation. The major drawback of the TVAE is its incompatibility with the CVAE, which currently prevents us from effectively controlling the class of data generated.

I have also seen that the DP-SGD is simple to use, and its disadvantages can be calculated so that they can be compensated for as far as possible before it is used. This simplicity makes it much easier to use the DP-SGD, and I can imagine applying a relatively strong constraint ($\epsilon = 2$) with one DO to public models that require such protection for a robust guarantee of confidentiality and minimal loss of performance after training that has lasted the optimal number of epochs under this constraint.

5.4 Scientific contribution

Our experiments confirm the performance of the TVAE, as first proven in [20] and confirmed in [80]. This Master thesis provides a better understanding of its advantages and inconveniences, facilitating future research on this subject. I have also demonstrated the portability of some 2D VAE extensions and the incompatibility of other extensions with the TVAE. These results open the way to many possible studies with the many existing variants of VAE. The most promising seems to be LVAE and VAE with Vamp priors [32, 28].

I also confirm the benefits of KL warm-up on both VAE and TVAE, as

well as its superiority over B disentanglement, another method of similar use [32, 34].

Our demonstration of a maximum noise point for applying the DP-SGD is another notable point of this Master thesis. While the DO has shown its usefulness in increasing this maximum noise value, a study of other regularisation methods or performance improvements similar to that carried out in this Master thesis would allow characterising the behaviour of this maximum noise better. This study would enable us to further limit the DP-SGD's impact on the models' performance.

5.5 Ethical and sustainable aspects

This thesis has a strong ethical aspect regarding the protection of personal data, on the one hand, through the use of synthetic data and on the other, through the use of DP guarantees. Synthetic data adds a layer of protection in case the training data of the model is leaked, but also in case the model itself is leaked, breached, or attacked since, in all cases, it is data that is not linked to specific individuals affected.

However, this protection is incomplete and benefits from being combined with DP to provide a mathematical guarantee that no row can be retrieved individually. However, DP becomes less effective as its protection has to be applied to more rows simultaneously [17, 18, 19]. Therefore, there is no guarantee that the entire synthetic dataset is protected in case of a complete leak, making it impossible to use these techniques to violate GDPR rules.

Since synthetic data only mimics the distribution of a real dataset, users should ensure that the dataset is of the highest possible quality before using it as a source of generated data. Otherwise, the synthetic data reproduces the biases and errors of the original dataset.

Finally, this thesis also works on energy sobriety in AI by encouraging research and industry to consider models with less than 100,000 trainable weights instead of trying to solve all modern problems with LLMs.

5.6 Future work

The first step in continuing this work would be to complete the synthetic data application experiments, for example, in the Census, and combine TVAE and DP-SGD as initially planned. Other potential improvements specific to the TVAE include keeping only high-confidence generated data, improving its

transformation process and finding a way to make the CVAE compatible with the TVAE.

Many avenues have been opened for the application of variants of the VAE to the TVAE, and I would have liked to study the LVAE, the VQ-VAE II, or even the adversarial VAE.

As mentioned above, research into determining and increasing the maximum noise level for the DP-SGD is also possible.

Another possible avenue is research into diffusion models specialising in tabular data, where the latest techniques could be applied to a model with a not-so-distant architecture such as the LVAE, thus getting the best of both worlds.

Chapter 6

Conclusion

I investigated the performance of TVAE on three different datasets. The TVAE is efficient and generalisable, creating data with an overall score from 0.9 to 0.97 according to the SDV metrics. It benefits from improvements common to other deep neural networks, particularly those that limit its tendency to overfit. The TVAE is also compatible with many variants and techniques specific to multidimensional VAEs, but not all. In particular, the CVAE, one of the most promising and valuable VAE variants in data generation, is incompatible with the TVAE.

Nevertheless, the TVAE performances need to be improved to be applied to complex real-world cases such as anomaly detection and do not solve synthetic data generation problems such as the Matthew effect [88].

I also worked on differential privacy, where I verified the existence of a plateau where the model benefits from the protection of differential privacy while limiting the degradation of its performance. This plateau balances the privacy constraints and the learning time of the model before learning, allowing greater control of the protection offered by differential privacy and optimal training time.

This Master thesis, therefore, paves the way for future research on the promising TVAE model and its variants, where major discoveries could be made quickly, as most advances in VAE are compatible with TVAE.

References

[1] J. Drechsler and A.-C. Haensch, "30 Years of Synthetic Data," Apr. 2023, arXiv:2304.02107 [cs]. [Online]. Available: http://arxiv.org/abs/2304.02107 [Page 1.]

[2] V. Borisov, T. Leemann, K. Seßler, J. Haug, M. Pawelczyk, and G. Kasneci, "Deep Neural Networks and Tabular Data: A Survey," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, 2022. doi: 10.1109/TNNLS.2022.3229161 ArXiv:2110.01889 [cs]. [Online]. Available: http://arxiv.org/abs/2110.01889 [Pages 1 and 2.]

[3] C. A. Mami, A. Coser, E. Medvet, A. T. P. Boudewijn, M. Volpe, M. Whitworth, B. Svara, G. Sgroi, D. Panfilo, and S. Saccani, "Generating Realistic Synthetic Relational Data through Graph Variational Autoencoders," Nov. 2022, arXiv:2211.16889 [cs]. [Online]. Available: http://arxiv.org/abs/2211.16889 [Pages 1 and 2.]

[4] V. Borisov, K. Seßler, T. Leemann, M. Pawelczyk, and G. Kasneci, "Language Models are Realistic Tabular Data Generators," Apr. 2023, arXiv:2210.06280 [cs]. [Online]. Available: http://arxiv.org/abs/2210.06280 [Pages 1 and 2.]

[5] S. Kamthe, S. Assefa, and M. Deisenroth, "Copula Flows for Synthetic Data Generation," Jan. 2021, arXiv:2101.00598 [cs, stat] version: 1. [Online]. Available: http://arxiv.org/abs/2101.00598 [Pages 1 and 2.]

[6] H. Wilde, J. Jewson, S. Vollmer, and C. Holmes, "Foundations of Bayesian Learning from Synthetic Data," Nov. 2020, arXiv:2011.08299 [cs, stat] version: 2. [Online]. Available: http://arxiv.org/abs/2011.08299 [Page 1.]

[7] M. R. Behera, S. Upadhyay, S. Shetty, S. Priyadarshini, P. Patel, and K. F. Lee, "FedSyn: Synthetic Data Generation using Federated

Learning," Apr. 2022, arXiv:2203.05931 [cs, stat] version: 2. [Online]. Available: http://arxiv.org/abs/2203.05931 [Page 1.]

[8] M. Vero, M. Balunović, and M. Vechev, "Programmable Synthetic Tabular Data Generation," Jul. 2023, arXiv:2307.03577 [cs] version: 2. [Online]. Available: http://arxiv.org/abs/2307.03577 [Page 2.]

[9] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," Dec. 2022, arXiv:1312.6114 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1312.6114 [Pages 2, 7, and 20.]

[10] A. Desai, C. Freeman, Z. Wang, and I. Beaver, "TimeVAE: A Variational Auto-Encoder for Multivariate Time Series Generation," Dec. 2021, arXiv:2111.08095 [cs] version: 3. [Online]. Available: http://arxiv.org/abs/2111.08095 [Page 2.]

[11] D. Tang, D. Liang, T. Jebara, and N. Ruozzi, "Correlated Variational Auto-Encoders," Apr. 2020, arXiv:1905.05335 [cs, stat] version: 5. [Online]. Available: http://arxiv.org/abs/1905.05335 [Pages 2 and 3.]

[12] Y. Jin, M. Liu, Y. Li, R. Xu, L. Du, L. Gao, and Y. Xiang, "Variational Auto-encoder Based Bayesian Poisson Tensor Factorization for Sparse and Imbalanced Count Data," *Data Mining and Knowledge Discovery*, vol. 35, no. 2, pp. 505–532, Mar. 2021. doi: 10.1007/s10618-020-00723-7 ArXiv:1910.05570 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1910.05570 [Pages 2 and 3.]

[13] C. Chen, X. Feng, J. Zhou, J. Yin, and X. Zheng, "Federated Large Language Model: A Position Paper," Jul. 2023, arXiv:2307.08925 [cs]. [Online]. Available: http://arxiv.org/abs/2307.08925 [Page 2.]

[14] "The Role of Synthetic Data in Healthcare: From Innovation to Diagnosis," Aug. 2023. [Online]. Available: https://ydata.ai/resources/the-role-of-synthetic-data-in-healthcare-from-innovation-to-improved-diagnosis [Page 2.]

[15] "Credit Card Fraud Detection." [Online]. Available: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud [Pages 2 and 24.]

[16] "General Data Protection Regulation (GDPR) – Official Legal Text." [Online]. Available: https://gdpr-info.eu/ [Page 2.]

[17] C. Dwork, "Differential Privacy," in *Automata, Languages and Programming*, ser. Lecture Notes in Computer Science, M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, Eds. Berlin, Heidelberg: Springer, 2006. doi: 10.1007/11787006$_1$.$ISBN978-3-540-35908-1pp.1--12$. [*Pages* 3, 15, *and* 53.]

[18] J. P. Near and C. Abuah, "Programming Differential Privacy." [Online]. Available: https://programming-dp.com/ [Pages 3, 15, 16, 17, 18, and 53.]

[19] R. Danger, "Differential Privacy: What is all the noise about?" May 2022, arXiv:2205.09453 [cs]. [Online]. Available: http://arxiv.org/abs/2205.09453 [Pages 3, 15, 21, and 53.]

[20] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling Tabular data using Conditional GAN," Oct. 2019, arXiv:1907.00503 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1907.00503 [Pages ix, 3, 7, 14, 20, 21, 23, 24, 25, 28, 32, 37, 39, 42, 51, and 52.]

[21] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic Backpropagation and Approximate Inference in Deep Generative Models," May 2014, arXiv:1401.4082 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1401.4082 [Pages 7 and 20.]

[22] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, "A Hierarchical Latent Vector Model for Learning Long-Term Structure in Music," Nov. 2019, arXiv:1803.05428 [cs, eess, stat]. [Online]. Available: http://arxiv.org/abs/1803.05428 [Pages 7 and 20.]

[23] A. Razavi, A. v. d. Oord, and O. Vinyals, "Generating Diverse High-Fidelity Images with VQ-VAE-2," Jun. 2019, arXiv:1906.00446 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1906.00446 [Page 7.]

[24] D. Yarats, A. Zhang, I. Kostrikov, B. Amos, J. Pineau, and R. Fergus, "Improving Sample Efficiency in Model-Free Reinforcement Learning from Images," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 12, pp. 10 674–10 681, May 2021. doi: 10.1609/aaai.v35i12.17276 Number: 12. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/17276 [Page 7.]

[25] M. Zhang, T. Z. Xiao, B. Paige, and D. Barber, "Improving VAE-based Representation Learning," May 2022, arXiv:2205.14539 [cs, stat]. [Online]. Available: http://arxiv.org/abs/2205.14539 [Pages 7, 13, and 28.]

[26] L. Bergamin, T. Carraro, M. Polato, and F. Aiolli, “Novel Applications for VAE-based Anomaly Detection Systems,” Apr. 2022, arXiv:2204.12577 [cs]. [Online]. Available: http://arxiv.org/abs/2204.12577 [Page 7.]

[27] M. Sami and I. Mobin, “A Comparative Study on Variational Autoencoders and Generative Adversarial Networks,” in *2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT)*, Mar. 2019. doi: 10.1109/ICAIIT.2019.8834544 pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/document/8834544 [Page 7.]

[28] J. M. Tomczak and M. Welling, “VAE with a VampPrior,” Feb. 2018, arXiv:1705.07120 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1705.07120 [Pages 7, 20, 28, and 52.]

[29] S. Zhao, J. Song, and S. Ermon, “InfoVAE: Information Maximizing Variational Autoencoders,” May 2018, arXiv:1706.02262 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1706.02262 [Pages 8, 12, 20, and 28.]

[30] C. Chadebec, L. J. Vincent, and S. Allassonnière, “Pythae: Unifying Generative Autoencoders in Python – A Benchmarking Use Case,” Jul. 2023, arXiv:2206.08309 [cs, stat]. [Online]. Available: http://arxiv.org/abs/2206.08309 [Pages 8 and 20.]

[31] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, “-VAE: LEARNING BASIC VISUAL CONCEPTS WITH A CONSTRAINED VARIATIONAL FRAMEWORK,” 2017. [Pages 10, 12, and 20.]

[32] C. K. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby, and O. Winther, “Ladder Variational Autoencoders,” May 2016, arXiv:1602.02282 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1602.02282 [Pages ix, 10, 11, 12, 20, 39, 41, 42, 52, and 53.]

[33] H. Fu, C. Li, X. Liu, J. Gao, A. Celikyilmaz, and L. Carin, “Cyclical Annealing Schedule: A Simple Approach to Mitigating KL Vanishing,” Jun. 2019, arXiv:1903.10145 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1903.10145 [Pages 10, 13, 20, and 41.]

[34] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner, “Understanding disentangling in β-VAE,” Apr. 2018, arXiv:1804.03599 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1804.03599 [Pages 12, 20, and 53.]

[35] I. O. Tolstikhin, B. K. Sriperumbudur, and B. Schölkopf, "Minimax estimation of maximum mean discrepancy with radial kernels," in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Curran Associates, Inc., 2016. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2016/file/5055cbf43fac3f7e2336b27310f0b9ef-Paper.pdf [Page 12.]

[36] A. v. d. Oord, O. Vinyals, and K. Kavukcuoglu, "Neural Discrete Representation Learning," May 2018, arXiv:1711.00937 [cs]. [Online]. Available: http://arxiv.org/abs/1711.00937 [Pages 13 and 20.]

[37] T. R. Davidson, L. Falorsi, N. De Cao, T. Kipf, and J. M. Tomczak, "Hyperspherical Variational Auto-Encoders," Sep. 2022, arXiv:1804.00891 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1804.00891 [Pages 13 and 20.]

[38] M. Razghandi, H. Zhou, M. Erol-Kantarci, and D. Turgut, "Variational Autoencoder Generative Adversarial Network for Synthetic Data Generation in Smart Home," Jan. 2022, arXiv:2201.07387 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2201.07387 [Pages 13 and 20.]

[39] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schoelkopf, "Wasserstein Auto-Encoders," Dec. 2019, arXiv:1711.01558 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1711.01558 [Pages 13 and 20.]

[40] Y. Burda, R. Grosse, and R. Salakhutdinov, "Importance Weighted Autoencoders," Nov. 2016, arXiv:1509.00519 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1509.00519 [Pages 14 and 20.]

[41] T. Rainforth, A. R. Kosiorek, T. A. Le, C. J. Maddison, M. Igl, F. Wood, and Y. W. Teh, "Tighter variational bounds are not necessarily better," 2019. [Pages 14 and 20.]

[42] I. A. M. Huijben, W. Kool, M. B. Paulus, and R. J. G. van Sloun, "A Review of the Gumbel-max Trick and its Extensions for Discrete Stochasticity in Machine Learning," Mar. 2022, arXiv:2110.01515 [cs, stat]. [Online]. Available: http://arxiv.org/abs/2110.01515 [Pages 14 and 37.]

[43] A. Korkic, "Benchmarking the hyper-parameter sensitivity of vae models for cancer treatment." [Page 14.]

[44] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "L-diversity: privacy beyond k-anonymity," in *22nd International Conference*

on Data Engineering (ICDE'06), Apr. 2006. doi: 10.1109/ICDE.2006.1 pp. 24–24, iSSN: 2375-026X. [Online]. Available: https://ieeexplore.ieee.org/document/1617392 [Page 15.]

[45] N. Li, T. Li, and S. Venkatasubramanian, "t-Closeness: Privacy Beyond k-Anonymity and l-Diversity," in *2007 IEEE 23rd International Conference on Data Engineering*, Apr. 2007. doi: 10.1109/ICDE.2007.367856 pp. 106–115, iSSN: 2375-026X. [Online]. Available: https://ieeexplore.ieee.org/document/4221659 [Page 15.]

[46] J. Domingo-Ferrer and J. Soria-Comas, "From t-closeness to differential privacy and vice versa in data anonymization," *Knowledge-Based Systems*, vol. 74, pp. 151–158, Jan. 2015. doi: 10.1016/j.knosys.2014.11.011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705114004031 [Page 15.]

[47] I. Mironov, "Renyi differential privacy," *CoRR*, vol. abs/1702.07476, 2017. [Online]. Available: http://arxiv.org/abs/1702.07476 [Page 18.]

[48] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep Learning with Differential Privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: Association for Computing Machinery, Oct. 2016. doi: 10.1145/2976749.2978318. ISBN 978-1-4503-4139-4 pp. 308–318. [Online]. Available: https://doi.org/10.1145/2976749.2978318 [Page 18.]

[49] "TensorFlow Privacy," Nov. 2023, original-date: 2018-12-21T18:46:46Z. [Online]. Available: https://github.com/tensorflow/privacy [Pages 19, 21, and 33.]

[50] "Differential Privacy," Nov. 2023, original-date: 2019-09-04T13:04:15Z. [Online]. Available: https://github.com/google/differential-privacy [Pages 19, 21, and 33.]

[51] B. Balle, G. Barthe, and M. Gaboardi, "Privacy Amplification by Subsampling: Tight Analyses via Couplings and Divergences," Nov. 2018, arXiv:1807.01647 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1807.01647 [Pages 19 and 21.]

[52] I. Yeo and R. A. Johnson, "A new family of power transformations to improve normality or symmetry," *Biometrika*, vol. 87, no. 4, pp.

954–959, Dec. 2000. doi: 10.1093/biomet/87.4.954. [Online]. Available: https://doi.org/10.1093/biomet/87.4.954 [Page 19.]

[53] S. Lin, R. Clark, R. Birke, S. Schönborn, N. Trigoni, and S. Roberts, "Anomaly Detection for Time Series Using VAE-LSTM Hybrid Model," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020. doi: 10.1109/ICASSP40776.2020.9053558 pp. 4322–4326, iSSN: 2379-190X. [Online]. Available: https://ieeexplore.ieee.org/document/9053558 [Page 20.]

[54] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling, "Improving Variational Inference with Inverse Autoregressive Flow," Jan. 2017, arXiv:1606.04934 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1606.04934 [Page 20.]

[55] S. Mandal, A. A. Jammal, and F. A. Medeiros, "Assessing glaucoma in retinal fundus photographs using Deep Feature Consistent Variational Autoencoders," Oct. 2021, arXiv:2110.01534 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2110.01534 [Page 20.]

[56] R. T. Q. Chen, X. Li, R. Grosse, and D. Duvenaud, "Isolating Sources of Disentanglement in Variational Autoencoders," Apr. 2019, arXiv:1802.04942 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1802.04942 [Page 20.]

[57] J. Snell, K. Ridgeway, R. Liao, B. D. Roads, M. C. Mozer, and R. S. Zemel, "Learning to Generate Images with Perceptual Similarity Metrics," Jan. 2017, arXiv:1511.06409 [cs]. [Online]. Available: http://arxiv.org/abs/1511.06409 [Pages 20 and 34.]

[58] S. Ferdowsi, M. Diephuis, S. Rezaeifar, and S. Voloshynovskiy, "ρ-VAE: Autoregressive parametrization of the VAE encoder," Sep. 2019, arXiv:1909.06236 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1909.06236 [Page 20.]

[59] F. P. Casale, A. V. Dalca, L. Saglietti, J. Listgarten, and N. Fusi, "Gaussian Process Prior Variational Autoencoders," Nov. 2018, arXiv:1810.11738 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1810.11738 [Page 20.]

[60] E. Mathieu, C. L. Lan, C. J. Maddison, R. Tomioka, and Y. W. Teh, "Continuous Hierarchical Representations with Poincar\'e Variational Auto-Encoders," Nov. 2019, arXiv:1901.06033 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1901.06033 [Page 20.]

[61] Y. Kim, S. Wiseman, A. C. Miller, D. Sontag, and A. M. Rush, "Semi-Amortized Variational Autoencoders," Jul. 2018, arXiv:1802.02550 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1802.02550 [Page 20.]

[62] A. L. Caterini, A. Doucet, and D. Sejdinovic, "Hamiltonian Variational Auto-Encoder," Nov. 2018, arXiv:1805.11328 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1805.11328 [Page 20.]

[63] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial Autoencoders," May 2016, arXiv:1511.05644 [cs]. [Online]. Available: http://arxiv.org/abs/1511.05644 [Page 20.]

[64] I. Gulrajani, K. Kumar, F. Ahmed, A. A. Taiga, F. Visin, D. Vazquez, and A. Courville, "PixelVAE: A Latent Variable Model for Natural Images," Nov. 2016, arXiv:1611.05013 [cs]. [Online]. Available: http://arxiv.org/abs/1611.05013 [Page 20.]

[65] A. Thin, N. Kotelevskii, A. Doucet, A. Durmus, E. Moulines, and M. Panov, "Monte Carlo Variational Auto-Encoders," Jun. 2021, arXiv:2106.15921 [cs, stat]. [Online]. Available: http://arxiv.org/abs/2106.15921 [Page 20.]

[66] S. Cao, J. Li, K. P. Nelson, and M. A. Kon, "Coupled VAE: Improved Accuracy and Robustness of a Variational Autoencoder," Jul. 2021, arXiv:1906.00536 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1906.00536 [Page 20.]

[67] K. Gregor, G. Papamakarios, F. Besse, L. Buesing, and T. Weber, "Temporal Difference Variational Auto-Encoder," Jan. 2019, arXiv:1806.03107 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1806.03107 [Page 20.]

[68] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2015/file/8d55a249e6baa5c06772297520da2051-Paper.pdf [Page 20.]

[69] H. Shao, S. Yao, D. Sun, A. Zhang, S. Liu, D. Liu, J. Wang, and T. Abdelzaher, "ControlVAE: Controllable Variational Autoencoder," Jun. 2020, arXiv:2004.05988 [cs, stat]. [Online]. Available: http://arxiv.org/abs/2004.05988 [Page 20.]

[70] H. Kamata, Y. Mukuta, and T. Harada, "Fully Spiking Variational Autoencoder," Dec. 2021, arXiv:2110.00375 [cs]. [Online]. Available: http://arxiv.org/abs/2110.00375 [Page 20.]

[71] Y. Shi, N. Siddharth, B. Paige, and P. H. S. Torr, "Variational Mixture-of-Experts Autoencoders for Multi-Modal Deep Generative Models," Nov. 2019, arXiv:1911.03393 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1911.03393 [Page 20.]

[72] H. Kim and A. Mnih, "Disentangling by Factorising," Jul. 2019, arXiv:1802.05983 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1802.05983 [Page 20.]

[73] P. Ghosh, M. S. M. Sajjadi, A. Vergari, M. Black, and B. Schölkopf, "From Variational to Deterministic Autoencoders," May 2020, arXiv:1903.12436 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1903.12436 [Page 20.]

[74] C. Chadebec, E. Thibeau-Sutre, N. Burgos, and S. Allassonnière, "Data Augmentation in High Dimensional Low Sample Size Setting Using a Geometry-Based Variational Autoencoder," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 2879–2896, Mar. 2023. doi: 10.1109/TPAMI.2022.3185773 Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence. [Online]. Available: https://ieeexplore.ieee.org/document/9806307 [Page 20.]

[75] M. van Bree, "Unlocking the Potential of Synthetic Tabular Data Generation with Variational Autoencoders." [Pages 20, 21, 23, 24, 37, 45, 51, and 68.]

[76] "Gretel.ai — The synthetic data platform for developers." [Online]. Available: https://gretel.ai/ [Pages 21 and 27.]

[77] "Artificial Intelligence (AI) Services & Solutions | Turing." [Online]. Available: https://www.turing.com/services/ai#14 [Pages 21 and 27.]

[78] "_USReplicate." [Online]. Available: https://www.datomize.com/replicate/ [Pages 21 and 27.]

[79] X. Amatriain, "On the "usefulness" of the Netflix Prize," Jul. 2021. [Online]. Available: https://xamat.medium.com/on-the-usefulness-of-the-netflix-prize-403d360aaf2 [Page 21.]

[80] T. Hann, "SDV vs MOSTLY AI: Which synthetic data generator is better? - MOSTLY AI," Aug. 2022, section: Generate synthetic data. [Online].

Available: https://mostly.ai/blog/sdv-vs-mostly-ai-synthetic-data-generator-comparison [Pages 21, 28, 38, 51, and 52.]

[81] N. Patki, R. Wedge, and K. Veeramachaneni, "The Synthetic Data Vault," in *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, Oct. 2016. doi: 10.1109/DSAA.2016.49 pp. 399–410. [Online]. Available: https://ieeexplore.ieee.org/document/7796926 [Pages 21, 24, 32, and 38.]

[82] "Adult Census Income." [Online]. Available: https://www.kaggle.com/datasets/uciml/adult-census-income [Page 23.]

[83] P. Adams, M. Doman, and P. Kuchipudi, "The Berka Dataset." [Online]. Available: https://www.kaggle.com/datasets/marceloventura/the-berka-dataset [Page 23.]

[84] Y. LeCun, C. Cortes, and C. J. C. Burgess, "MNIST handwritten digit database." [Online]. Available: http://yann.lecun.com/exdb/mnist/ [Page 24.]

[85] S. Kumar, "Churn Modelling." [Online]. Available: https://www.kaggle.com/datasets/shubh0799/churn-modelling [Page 24.]

[86] A. Kotelnikov, D. Baranchuk, I. Rubachev, and A. Babenko, "TabDDPM: Modelling Tabular Data with Diffusion Models," Sep. 2022, arXiv:2209.15421 [cs]. [Online]. Available: http://arxiv.org/abs/2209.15421 [Page 27.]

[87] R. Müller, S. Kornblith, and G. Hinton, "When Does Label Smoothing Help?" Jun. 2020, arXiv:1906.02629 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1906.02629 [Page 39.]

[88] G. Ganev, B. Oprisanu, and E. De Cristofaro, "Robin Hood and Matthew Effects: Differential Privacy Has Disparate Impact on Synthetic Data," Jun. 2022, arXiv:2109.11429 [cs] version: 3. [Online]. Available: http://arxiv.org/abs/2109.11429 [Pages 44, 51, and 55.]

TRITA-EECS-EX- 2023:0000

Appendix: Experiment results

Experiment 1.1: VAE vs TVAE

Our first test was to compare the VAE with the TVAE to see what progress the latter had made.

Model	Overall score transformed	Column shape transformed	Numerical shape transformed	Categorical shape transformed	Correlation transformed	Coverage transformed	Boundary adherence transformed
TVAE Census	0,947	0,957	0,976	0,938	/0,953	0,88	0,999
VAE Census	0,779	0,667	0,521	0,813	/0,97	0,746	0,735
TVAE berka	0,965	0,895	0,976	0,814	0,987	0,982	0,995
VAE berka 300 epochs	0,965	0,895	0,976	0,81	0,987	0,977	1
VAE berka 100 epochs	0,957	0,89	0,963	0,82	0,965	1	0,973
TVAE creditcards	0,89	0,988	0,79	0,997	0,99	0,6	0,999
VAE creditcard	0,83	0,938	0,877	0,998	0,381	1	0,99

Figure 1: Results of experiment 1.1

Experiment 1.2: TVAE with Gaussian log likelihood (LL) vs TVAE

Mentioned in van Bree's master thesis [75] is the ST-VAE. It uses a Gaussian log-likelihood measure as the reconstruction loss instead of an MSE, and is said to perform better than a conventional VAE. I wanted to check this on TVAE.

Model	Overall score transformed	column shape transformed	numerical shape transformed	categorical shape transformed	correlation transformed	coverage transformed	boundary adherence transformed
TVAE Census LL	0,94	0,94	0,941	0,938	/0,918	0,88	0,999
TVAE Census base	0,947	0,957	0,976	0,938	/0,953	0,88	0,999
TVAE credicards LL	0,592	0,259	0,515	0,002	0,78	0,136	0,994
TVAE creditcards	0,89	0,988	0,79	0,997	0,99	0,6	0,999
TVAE berka LL	0,957	0,87	0,92	0,81	0,98	0,976	0,99
TVAE berka base	0,965	0,895	0,976	0,814	0,987	0,982	0,995

Figure 2: Results of experiment 1.2

Experiment 1.3: TVAE with BCE vs TVAE with CCE

The TVAE uses a CCE loss on categorical columns, applied by grouping from the same categorical column. We compared this to an ECB applied globally to all the categorical columns at the same time.

Model	Overall score transformed	column shape transformed	numerical shape transformed	categorical shape transformed	correlation transformed	coverage transformed	boundary adherence transformed
Berka - BCE	0,965	0,905	0,988	0,822	0,976	0,984	0,996
TVAE berka base	0,965	0,895	0,976	0,814	0,987	0,982	0,995
Census - BCE	0,946	0,956	0,975	0,937	NE (0,951)	0,884	0,999
TVAE Census base	0,947	0,957	0,976	0,938	/0,953	0,88	0,999

Figure 3: Results of experiment 1.3

Experiment 2: Softmax vs Gumbel Max Trick

We tested the Softmax activation function for the generation of categorical columns and compared it with the use of the Gumbel Max trick at different temperatures for the same columns.

	Overall score transformed	column shape transformed	numerical shape transformed	categorical shape transformed	correlation transformed	coverage transformed	boundary adherence transformed
Berka TVAE Gumbel t=3.85	0.967	0.896	0.981	0.812	0.992	0.986	0.995
Berka TVAE Gumbel t=0.35	0,962	0,888	0,962	0,814	0,984	0,983	0,996
TVAE berka base	0,965	0,895	0,976	0,814	0,987	0,982	0,995
Berka TVAE Gumbel t=3.85	0.948	0.955	0.974	0.937	/0.943	0.877	0.998
Berka TVAE Gumbel t=0.35	0,935	0,925	0,979	0,88	/0,942	0,88	0,999
TVAE Census base	0,947	0,957	0,976	0,938	/0,953	0,88	0,999
Creditcard TVAE Gumbel t=3.85	0.887	0.978	0.959	0.998	0.990	0.58	0.999
Creditcard TVAE Gumbel t=0.35	0,864	0,868	0,809	0,926	0,988	0,607	0,997
TVAE creditcards	0,89	0,988	0,979	0,997	0,99	0,6	0,999

Figure 4: Results of experiment 2

Experiment 3: KL warm-up and Batch Normalization

Model	Overall score transformed	column shape transformed	numerical shape transformed	categorical shape transformed	correlation transformed	coverage transformed	boundary adherence transformed
berka KL warm up cyclical + BN	0,966	0,899	0,977	0,82	0,98	0,99	0,994
berka KL warm up batch + BN	0,968	0,897	0,977	0,82	0,984	0,995	0,995
berka KL warm up epoch +BN	0,965	0,897	0,977	0,82	0,986	0,981	0,994
berka BN	0,968	0,9	0,983	0,817	0,987	0,993	0,994
berka KL warm up epoch no BN	0,964	0,89	0,962	0,82	0,98	0,993	0,994
Berka TVAE	0,962	0,888	0,962	0,814	0,984	0,983	0,996
census KL warm up cyclical + BN	0,943	0,949	0,967	0,931	/0,942	0,88	1
census KL warm up batch + BN	0,943	0,949	0,967	0,931	/0,954	0,88	1
census KL warm up epoch+ BN	0,944	0,95	0,968	0,932	/0,946	0,88	1
census BN	0,943	0,949	0,97	0,928	/0,953	0,88	1
census KL warm up epoch no BN	0,939	0,935	0,974	0,897	/0,94	0,88	1
Census TVAE	0,935	0,925	0,979	0,88	/0,942	0,88	0,999
creditcards KL warm up cyclical + BN	0,873	0,96	0,923	0,998	0,89	0,54	0,999
creditcards KL warm up batch + BN	0,879	0,958	0,919	0,998	0,989	0,57	0,999
creditcards KL warm up epoch+ BN	0,874	0,944	0,891	0,998	0,989	0,57	0,999
creditcards BN	0,879	0,946	0,894	0,998	0,988	0,586	0,997
creditcards KL warm up epoch no BN	0,875	0,895	0,853	0,936	0,622	0,997	0,985
Creditcard TVAE	0,864	0,868	0,809	0,926	0,988	0,607	0,997

Figure 5: Results of experiment 3

Experiment 4: Dropout

	Overall score transformed	column shape transformed	numerical shape transformed	categorical shape transformed	correlation transformed	coverage transformed	boundary adherence transformed
TVAE Berka DO + BN	0,967	0,896	0,976	0,985	0,816	0,992	0,994
TVAE Berka DO	0,966	0,892	0,97	0,991	0,814	0,984	0,997
TVAE Berka BN	0,968	0,9	0,983	0,817	0,987	0,993	0,994
TVAE Berka	0,962	0,888	0,962	0,814	0,984	0,983	0,996
Census DO + BN	0,946	0,957	0,982	0,932	/0,958	0,883	1
Census DO	0,944	0,954	0,977	0,923	/0,956	0,88	0,999
Census BN	0,943	0,949	0,97	0,928	/0,953	0,88	1
Census	0,935	0,925	0,979	0,88	/0,942	0,88	0,999
Creditcards DO + BN	0,892	0,981	0,964	0,999	0,988	0,6	0,999
Creditcards DO	0,892	0,984	0,971	0,998	0,991	0,593	1
Creditcards BN	0,879	0,946	0,894	0,998	0,988	0,586	0,997
Creditcards	0,864	0,868	0,809	0,926	0,988	0,607	0,997

Figure 6: Results of experiment 4

Experiment 5: Leaky ReLU and SeLU

	Overall score transformed	column shape transformed	numerical shape transformed	categorical shape transformed	correlation transformed	coverage transformed	boundary adherence transformed
TVAE Berka Leaky ReLU	0,967	0,901	0,985	0,817	0,985	0,985	0,995
TVAE Berka SeLU	0,966	0,9	0,98	0,82	0,98	0,99	0,99
TVAE berka ReLU	0,962	0,888	0,962	0,814	0,984	0,983	0,996
TVAE census Leaky ReLU	0,937	0,93	0,961	0,899	/0,945	0,883	0,999
TVAE census Selu	0,941	0,947	0,962	0,931	/0,928	0,88	0,998
Census TVAE ReLU	0,935	0,925	0,979	0,88	/0,942	0,88	0,999
TVAE creditcards Leaky ReLU	0,869	0,867	0,762	0,97	0,989	0,636	0,985
TVAE creditcards SELU	0,844	0,791	0,843	0,739	0,986	0,615	0,985
Creditcards TVAE ReLU	0,864	0,868	0,809	0,926	0,988	0,607	0,997
Creditcards TVAE SELU + DO	0.893	0.985	0.971	0.998	0.988	0.601	0.999
Creditcards TVAE Leaky ReLU + BN + DO	0.889	0.980	0.962	0.998	0.991	0.587	0.999
Creditcards DO + BN	0,892	0,981	0,964	0,999	0,988	0,6	0,999

Figure 7: Results of experiment 5

Experiment 6: B-disentanglement

Model	Overall score transformed	column shape transformed	numerical shape transformed	categorical shape transformed	correlation transformed	coverage transformed	boundary adherence transformed
Berka 0 25 BN	0,962	0,89	0,962	0,817	0,986	0,985	0,991
Berka 0 50 BN	0,962	0,889	0,963	0,815	0,978	0,989	0,993
Berka 0 128 BN	0,958	0,886	0,954	0,819	0,964	0,986	0,994
Berka 0,5 25 BN	0,965	0,897	0,977	0,817	0,985	0,985	0,992
Berka 0,5 50 BN	0,962	0,893	0,968	0,818	0,972	0,99	0,993
Berka 1 25 BN	0,965	0,893	0,97	0,816	0,987	0,985	0,995
Berka 1 50 BN	0,959	0,884	0,95	0,816	0,975	0,987	0,992
Berka 5 25 BN	0,963	0,89	0,965	0,816	0,986	0,984	0,993
Berka 5 50 BN	0,962	0,89	0,962	0,817	0,973	0,992	0,994
Berka 0 25 no BN	0,963	0,885	0,957	0,814	0,984	0,992	0,992
Berka 5 25 no BN	0,967	0,89	0,97	0,81	0,986	0,993	0,996
Berka 5 25 BN	0,963	0,89	0,965	0,816	0,986	0,984	0,993
Berka TVAE sans B disentanglement ni BN	0,962	0,888	0,962	0,814	0,984	0,983	0,996
Census 5 25 no BN	0,936	0,929	0,962	0,895	/0,944	0,882	1
Census TVAE sans B disentanglement ni BN	0,935	0,925	0,979	0,88	/0,942	0,88	0,999
Creditcards 5 25 no BN	0,874	0,895	0,853	0,936	0,985	0,62	0,997
Creditcards 5 25 no BN (KL collapse)	0,902	0,985	0,972	0,998	0,989	0,636	0,999
Creditcard TVAE sans B disentanglement ni BN	0,864	0,868	0,809	0,926	0,988	0,607	0,997

Figure 8: Results of experiment 6. Reads the model name: Dataset, initial value then final value of the B-disentanglement, presence or absence of batch normalisation.

Experiment 7: Latent space dimension modifications

	Overall score transformed	column shape transformed	numerical shape transformed	categorical shape transformed	correlation transformed	coverage transformed	boundary adherence transformed
TVAE creditcards 512	0,875	0,92	0,845	0,995	0,99	0,595	0,996
TVAE creditcards 8 DO	0,893	0,982	0,967	0,998	0,99	0,602	0,999
TVAE creditcards 8	0,889	0,962	0,926	0,998	0,992	0,604	0,999
TVAE creditcards 2	0,773	0,474	0,947	0,001	0,989	0,63	0,999
TVAE creditcards 128	0,864	0,868	0,809	0,926	0,988	0,607	0,997
TVAE census 2	0,935	0,936	0,963	0,908	/0,955	0,871	0,999
TVAE census 8	0,969	0,938	0,963	0,914	/0,940	0,882	0,998
TVAE census 24	0,938	0,931	0,966	0,896	/0,946	0,882	1
TVAE census 32	0,935	0,923	0,966	0,88	/0,959	0,882	1
TVAE census 64	0,935	0,924	0,968	0,88	/0,943	0,882	0,999
TVAE census 512	0,932	0,923	0,964	0,958	/0,942	0,87	1
TVAE census 512-DO	0,94	0,949	0,975	0,923	/0,953	0,88	0,999
TVAE census 128	0,935	0,925	0,979	0,88	/0,942	0,88	0,999
TVAE berka 2	0,962	0,89	0,97	0,81	0,968	0,999	0,994
TVAE berka 8	0,964	0,88	0,95	0,816	0,987	0,996	0,991
TVAE berka 32	0,965	0,898	0,979	0,817	0,983	0,986	0,994
TVAE berka 64	0,968	0,896	0,976	0,816	0,986	0,995	0,994
TVAE berka 256	0,965	0,893	0,972	0,814	0,985	0,99	0,994
TVAE berka 512	0,964	0,898	0,978	0,818	0,982	0,983	0,994
TVAE berka R128	0,962	0,888	0,962	0,814	0,984	0,983	0,996

Figure 9: Results of experiment 7

The number at the end indicates the dimension of the latent space. The baseline model uses a latent space of dimension 128.

Experiment 8: Architecture modifications

The models are named according to the size of their layers, where the first number indicates the size of the first layer of the encoder and the last layer of the decoder, the second number indicates the size of the second layer of the encoder and the first layer of the decoder, and finally the third number indicates the size of the latent dimension.

For example, TVAE burka 256-128-64 indicates a model with an encoder of size 256 for its first layer and 128 for its second layer, a latent space of size

64 and then a decoder with a first layer of size 128 and a second layer of size 256.

Model	Overall score transformed	column shape transformed	numerical shape transformed	categorical shape transformed	correlation transformed	coverage transformed	boundary adherence transformed
TVAE creditcards 256-128-2	0,901	0,962	0,938	0,986	0,99	0,655	0,998
TVAE creditcards 256-128-8	0,893	0,971	0,944	0,998	0,99	0,613	0,999
TVAE creditcards 256-256-128	0,877	0,917	0,838	0,996	0,988	0,605	0,998
TVAE creditcards 128-128-128	0,864	0,868	0,809	0,926	0,988	0,607	0,997
TVAE creditcards 128-128- 8	0,889	0,962	0,926	0,998	0,992	0,604	0,999
TVAE census 256-128-8	0,941	0,945	0,966	0,925	/0,94	0,878	0,999
TVAE census 256-128-64	0,934	0,92	0,97	0,872	/0,941	0,883	0,998
TVAE census 256-256-256	0,934	0,922	0,973	0,872	/0,946	0,88	1
TVAE census 256-256-128	0,934	0,921	0,973	0,868	/0,946	0,88	1
TVAE census 128-128-128	0,935	0,925	0,979	0,88	/0,942	0,88	0,999
TVAE census 128-128-8	0,969	0,938	0,963	0,914	/0,940	0,882	0,998
TVAE berka 256-128-64	0,965	0,894	0,97	0,82	0,98	0,99	0,993
TVAE berka 256-128-128	0,967	0,898	0,98	0,817	0,986	0,99	0,993
TVAE berka 256-256-256	0,965	0,902	0,987	0,817	0,986	0,978	0,995
TVAE berka 256-256-128	0,968	0,897	0,977	0,817	0,986	0,994	0,995
TVAE berka 128-128-128	0,962	0,888	0,962	0,814	0,984	0,983	0,996
TVAE berka 128-128-64	0,968	0,896	0,976	0,816	0,986	0,995	0,994

Figure 10: Results of experiment 8

Experiment 9: Variants of VAE applied to TVAE on Berka dataset

	Overall score transformed	column shape transformed	numerical shape transformed	categorical shape transformed	correlation transformed	coverage transformed	boundary adherence transformed
Info VAE 64 BN DO	0,967	0,897	0,979	0,816	0,986	0,993	0,993
Info VAE 64	0,964	0,89	0,963	0,817	0,983	0,985	0,995
Info VAE 128	0.963	0.892	0.967	0.816	0.986	0.979	0.994
IWAE 64 BN DO 8 samples	0,969	0,9	0,983	0,817	0,988	0,993	0,995
IWAE 64 8 samples	0,965	0,898	0,979	0,817	0,986	0,983	0,995
IWAE 128 8 samples	0.965	0.898	0.979	0.816	0.985	0.981	0.995
CIWAE 64 BN DO 64 samples	0,967	0,894	0,974	0,812	0,989	0,991	0,996
CIWAE 64 BN DO 8 samples	0,968	0,898	0,982	0,813	0,989	0,991	0,996
CIWAE 64 8 samples	0,968	0,898	0,979	0,816	0,987	0,993	0,996
CIWAE	0.969	0.899	0.980	0.817	0.986	0.994	0.995
SVAE 64 BN DO	0,961	0,891	0,954	0,827	0,965	0,995	0,996
SVAE 64	0,956	0,885	0,941	0,828	0,961	0,981	0,998
SVAE 128	0.957	0.889	0.949	0.828	0.962	0.982	0.997
VAE GAN 64 BN DO	0,958	0,89	0,957	0,823	0,958	0,995	0,99
VAE GAN 64	0,962	0,882	0,947	0,817	0,986	0,984	0,996
CIWAE 64 8 samples 128 dim	0,966	0,895	0,975	0,816	0,987	0,986	0,996
IWAE 64 4 samples 128 dim BN DO	0,968	0,899	0,981	0,816	0,989	0,99	0,995
CVAE 64 8	0,951	0,864	0,913	0,815	0,962	1	0,979
CVAE 8 DO BN - Gumbel	0,966	0,891	0,971	0,811	0,987	0,993	0,993
CVAE 8 DO BN - Softmax	0,967	0,898	0,987	0,81	0,99	0,985	0,996
CVAE 64 - DO BN	0.969	0.898	0.984	0.812	0.988	0.994	0.994
CVAE 128	0.964	0.898	0.978	0.817	0.983	0.981	0.994
TVAE classic BN DO 64	0,969	0,899	0,982	0,815	0,99	0,994	0,994

Figure 11: Results of experiment 9

Experiment 10.1: Differential privacy applied to Churn Modelling - Without Dropout

Epsilon reduced represents the actual value of epsilon used, since I used a budget of 10% of epsilon to protect the real size of the dataset in the formula where it appeared. An infinite epsilon means no DP.

espilon	epsilon reduced	Noise Multiplier	accuracy	precision	recall	f1-score	AUC-ROC	Accuracy CV	std CV
0,5	0,45	6,98	0,79	0,66	0,79	0,71	0,46	0,72	0,03
0,7	0,63	5,2	0,79	0,75	0,79	0,76	0,66	0,76	0,03
0,85	0,765	4,39	0,78	0,7	0,78	0,72	0,55	0,77	0,02
1	0,9	3,81	0,76	0,71	0,76	0,73	0,59	0,77	0,02
2	1,8	2,15	0,8	0,73	0,8	0,72	0,71	0,79	0,017
3	2,7	1,59	0,79	0,68	0,79	0,71	0,66	0,8	0,017
4	3,6	1,31	0,8	0,7	0,8	0,71	0,63	0,8	0,017
5	4,5	1,14	0,8	0,76	0,8	0,72	0,68	0,8	0,017
6	5,4	1,04	0,8	0,72	0,8	0,71	0,71	0,8	0,015
7	6,3	0,96	0,8	0,8	0,8	0,72	0,71	0,8	0,017
8	7,2	0,9	0,81	0,78	0,81	0,74	0,73	0,79	0,015
9	8,1	0,84	0,8	0,69	0,8	0,71	0,81	0,8	0,02
10	9	0,82	0,81	0,78	0,81	0,76	0,72	0,8	0,016
inf	inf	0	0,86	0,85	0,86	0,85	0,86	0,85	0,017

Figure 12: Results of experiment 10.1

Experiment 10.2: Differential privacy applied to Churn Modelling - With Dropout

espilon	epsilon reduced	Noise multiplier	accuracy	precision	recall	f1-score	AUC-ROC
0,5	0,45	6,98	0,71	0,69	0,71	0,7	0,56
0,7	0,63	5,2	0,78	0,69	0,78	0,71	0,55
0,85	0,765	4,39	0,8	0,76	0,8	0,75	0,67
1	0,9	3,81	0,79	0,76	0,79	0,77	0,69
2	1,8	2,15	0,8	0,76	0,8	0,74	0,6
3	2,7	1,59	0,8	0,69	0,8	0,71	0,67
4	3,6	1,31	0,81	0,79	0,81	0,77	0,76
5	4,5	1,14	0,8	0,64	0,8	0,71	0,7
6	5,4	1,04	0,79	0,71	0,79	0,71	0,73
7	6,3	0,96	0,8	0,74	0,8	0,72	0,74
8	7,2	0,9	0,8	0,77	0,8	0,73	0,72
9	8,1	0,84	0,8	0,77	0,8	0,73	0,76
10	9	0,82	0,81	0,77	0,81	0,76	0,75
inf	inf	0	0,86	0,86	0,86	0,85	0,86

Figure 13: Results of experiment 10.2

Experiment 11.1: Noise vs Epoch dilemma for Differencial Privacy - Without dropout

	Macro			Weighted			Train		Test		
Epoch	Accuracy	Recall	F1 score	Accuracy	Recall	F1 score	score	accuracy	score	accuracy	Noise multiplicator
10	0,88	0,88	0,88	0,88	0,88	0,88	0,428	0,869	0,391	0,881	0,9993
20	0,87	0,87	0,87	0,87	0,87	0,87	0,471	0,863	0,434	0,872	1,1159
40	0,82	0,82	0,82	0,82	0,82	0,82	0,846	0,813	0,791	0,824	1,3961
60	0,81	0,8	0,81	0,81	0,81	0,81	1,189	0,801	1,126	0,806	1,6376
80	0,81	0,8	0,8	0,81	0,81	0,81	1,68	0,797	1,61	0,806	1,8469
100	0,78	0,77	0,77	0,78	0,77	0,78	1,416	0,763	1,34	0,774	2,0343
150	0,8	0,8	0,8	0,81	0,8	0,8	3,228	0,797	3,195	0,803	2,4399
200	0,71	0,71	0,71	0,71	0,71	0,71	3,09	0,712	2,96	0,714	2,7867

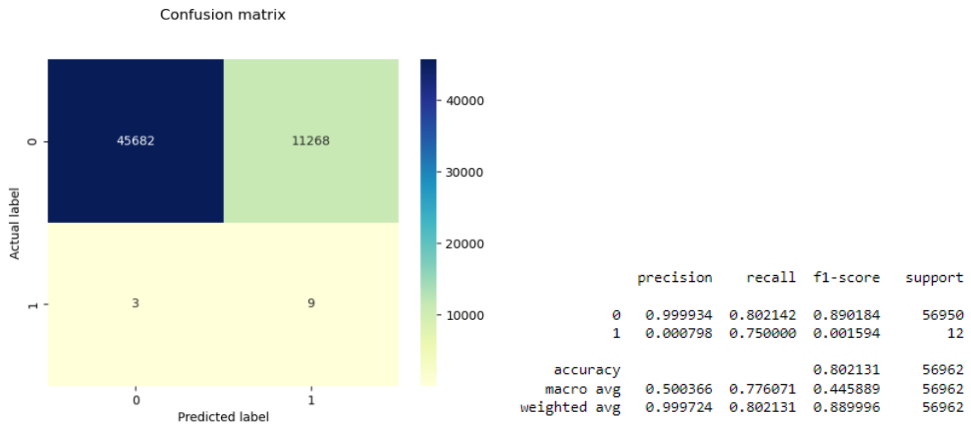
Figure 14: Results of experiment 11.1

Experiment 11.2: Noise vs Epoch dilemma for Differencial Privacy - With dropout

	Macro			Weighted			Train		Test		
Epoch	Accuracy	Recall	F1 score	Accuracy	Recall	F1 score	score	accuracy	score	accuracy	Noise multiplicator
10	0,89	0,88	0,88	0,89	0,89	0,89	0,433	0,874	0,408	0,886	0,9993
20	0,91	0,91	0,91	0,91	0,91	0,91	0,318	0,901	0,286	0,914	1,1159
40	0,91	0,91	0,91	0,91	0,91	0,91	0,311	0,903	0,276	0,912	1,3961
60	0,9	0,89	0,89	0,9	0,9	0,9	0,346	0,894	0,334	0,895	1,6376
80	0,9	0,89	0,89	0,9	0,9	0,9	0,369	0,886	0,328	0,896	1,8469
100	0,88	0,88	0,88	0,88	0,88	0,88	0,407	0,878	0,378	0,884	2,0343
150	0,83	0,83	0,83	0,83	0,83	0,83	0,588	0,824	0,558	0,829	2,4399
200	0,81	0,81	0,81	0,81	0,81	0,81	0,742	0,802	0,677	0,813	2,7867

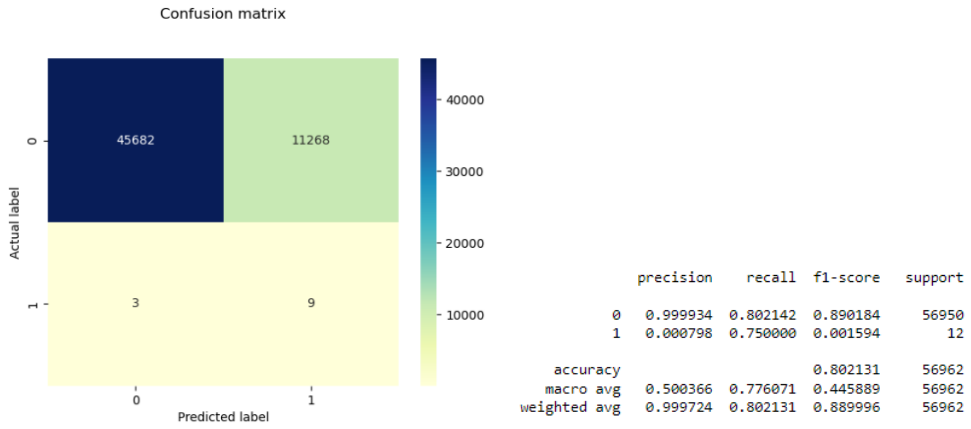
Figure 15: Results of experiment 11.2

Experiment 12 : Verification of the quality of synthetic data, on the creditcards dataset



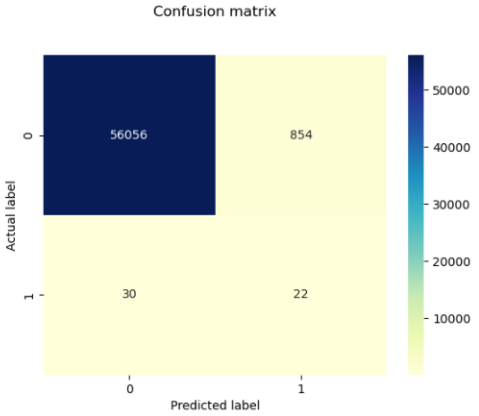
	precision	recall	f1-score	support
0	0.999934	0.802142	0.890184	56950
1	0.000798	0.750000	0.001594	12
accuracy			0.802131	56962
macro avg	0.500366	0.776071	0.445889	56962
weighted avg	0.999724	0.802131	0.889996	56962

Figure 16: Results of experiment 12: model trained on real data evaluated on real data



	precision	recall	f1-score	support
0	0.999934	0.802142	0.890184	56950
1	0.000798	0.750000	0.001594	12
accuracy			0.802131	56962
macro avg	0.500366	0.776071	0.445889	56962
weighted avg	0.999724	0.802131	0.889996	56962

Figure 17: Results of experiment 12: model trained on real data evaluated on synthetic data



	precision	recall	f1-score	support
0	0.999465	0.984994	0.992177	56910
1	0.025114	0.423077	0.047414	52
accuracy			0.984481	56962
macro avg	0.512290	0.704035	0.519795	56962
weighted avg	0.998576	0.984481	0.991314	56962

Figure 18: Results of experiment 12: model trained on synthetic data evaluated on synthetic data

TRITA-EECS-EX-2024:5