Norwegian University
of Life Sciences

Master's Thesis 2023 30 ECTS
Faculty of science and technology

Can Tabular Generative Models Generate Realistic Synthetic Near Infrared Spectroscopic Data?

Isak Finnøy
Data Science

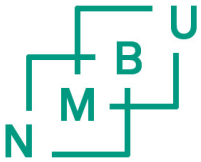
Master's Thesis

Can Tabular Generative Models Generate Realistic Synthetic Near Infrared Spectroscopic Data?

Written by:
Isak Finnøy

Supervisor:
Kristian Hovde Liland

Co-supervisors:
Ulf Geir Indahl, Per-Arne Andersen



Norwegian University
of Life Sciences

Master of Science
Faculty of Science and Technology
May 14, 2023

Abstract

In this thesis, we evaluated the performance of two generative models, Conditional Tabular Generative Adversarial Network (CTGAN) and Tabular Variational Autoencoder (TVAE), from the open-source library Synthetic Data Vault (SDV), for generating synthetic Near Infrared (NIR) spectral data. The aim was to assess the viability of these models in synthetic data generation for predicting Dry Matter Content (DMC) in the field of NIR spectroscopy. The fidelity and utility of the synthetic data were examined through a series of benchmarks, including statistical comparisons, dimensionality reduction, and machine learning tasks.

The results showed that while both CTGAN and TVAE could generate synthetic data with statistical properties similar to real data, TVAE outperformed CTGAN in terms of preserving the correlation structure of the data and the relationship between the features and the target variable, DMC. However, the synthetic data fell short in fooling machine learning classifiers, indicating a persisting challenge in synthetic data generation.

With respect to utility, neither synthetic dataset produced by CTGAN or TVAE could serve as a satisfactory substitute for real data in training machine learning models for predicting DMC. Although TVAE-generated synthetic data showed some potential when used with Random Forest (RF) and K-Nearest Neighbors (KNN) classifiers, the performance was still inadequate for practical use.

This study offers valuable insights into the use of generative models for synthetic NIR spectral data generation, highlighting their current limitations and potential areas for future research.

Preface

Writing this thesis has been a challenging and rewarding journey, one that I could not have completed without the support and encouragement of numerous individuals. First and foremost, I would like to express my deepest gratitude to my supervisor prof. Kristian Hovde Liland, and also my co-supervisors prof. Ulf Geir Indahl and prof. Per-Arne Andersen, for their invaluable guidance, expertise, and patience throughout this process.

My sincere thanks go to my fellow students, who have made the process of writing this thesis more enjoyable with their company and shared experiences. I also want to express my gratitude to my family for their continuous support throughout this journey. Lastly, I would like to thank my girlfriend for her patience and understanding during the demanding period of writing this thesis.

Writing this thesis has not only been a test of my academic abilities, but also a journey of personal growth. It has taught me the value of persistence, the importance of critical thinking, and the power of curiosity. I have learned to embrace challenges as opportunities for learning, and I am grateful for the lessons this process has imparted.

Abbreviations

ANN Artificial Neural Network.

CNN Convolutional Neural Network.

CTGAN Conditional Tabular Generative Adversarial Network.

DMC Dry Matter Content.

DT Decision Tree.

ELBO Evidence Lower Bound.

FCNN Fully-Connected Neural Network.

GAN Generative Adversarial Network.

GMM Gaussian Mixture Model.

HMM Hidden Markov Model.

Isomap Isometric Mapping.

KNN K-Nearest Neighbors.

KS Kolmogorov-Smirnov.

ML Machine Learning.

MLP Multilayered Perceptron.

MLR Multiple Linear Regression.

MWU Mann-Whitney U.

NIR Near Infrared.

PCA Principal Component Analysis.

PLSR Partial Least Squares Regression.

RF Random Forest.

RNN Recurrent Neural Network.

SDV Synthetic Data Vault.

SG Savitzky-Golay.

SVM Support Vector Machine.

TRTR Train on Real, Test on Real.

TSTR Train on Synthetic, Test on Real.

TVAE Tabular Variational Autoencoder.

VAE Variational Autoencoder.

VGM Variational Gaussian Mixtures.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Near Infrared (NIR) spectroscopy is a non-destructive technique used for the rapid analysis of various properties of organic and inorganic materials. It measures the absorption of light at different wavelengths and has found numerous applications in the fields of agriculture, food science, and environmental monitoring [1]. However, the limited availability of labeled data often impedes Machine Learning (ML) applications, such as NIR spectral data analysis [2].

To address this issue, this Master's thesis proposes the application of the Synthetic Data Vault (SDV) library's tabular data generators to model NIR spectra data [3]. The SDV library includes Conditional Tabular Generative Adversarial Network (CTGAN) and Tabular Variational Autoencoder (TVAE), both of which have shown efficacy in generating realistic synthetic tabular data [4].

NIR spectroscopy data can be represented as tabular data, with each row corresponding to a spectrum and each column representing a specific wavelength or frequency. Treating NIR spectra as tabular data provides several benefits. Firstly, it respects the inherent structure of the data: spectra are ordered sequences of measurements at different wavelengths, akin to the rows and columns of a table. Secondly, the data's tabular form ensures that features (i.e., wavelengths) are always in the same column, providing consistency and facilitating data interpretation and analysis. Lastly, tabular data formats are compatible with a wide range of readily available data analysis and ML tools. By treating NIR spectra as tabular data, we can leverage these tools, such as the SDV library, to generate synthetic NIR spectra that maintain the underlying structure, patterns, and relationships present in the real data without requiring expert knowledge.

1.1 Motivation

The primary motivation for this research is to explore the potential of generating synthetic NIR spectral data using the generative models provided by the SDV library, such as CTGAN and TVAE. Previous studies have utilized generative models in spectroscopy, but often with other techniques or highly specialized cases [5–7]. The primary objective of this thesis research is to create NIR spectra in a broad-based manner, without confining its reach to a specific use case.

A key aspect is the generation of synthetic data with corresponding target values. This is particularly important as obtaining reference measurements for NIR spectral data often involves significant financial and time resources. Hence, generating synthetic data that includes target values can provide considerable savings in time and money. Furthermore, this approach opens

up the potential to generate data points that might be physically impossible yet lie within the generative models' possible output range, providing additional opportunities for diverse and robust model training.

- **Addressing Data Scarcity**: Synthetic data can supplement limited real-world NIR spectra, offering a solution particularly for specific materials or conditions where data is scarce [8].
- **Enhancing Data Diversity**: Synthetic spectra can capture a wide range of conditions, thus enhancing the robustness of ML models by providing diverse training data [9].
- **Data Augmentation for Better Models**: Synthetic spectra can improve ML models' performance and help prevent overfitting by augmenting the available training data [9].
- **Cost and Time Efficiency**: Synthetic spectra can provide a more economical and time-efficient alternative to the process of collecting and annotating real-world spectra [8]. This can alleviate the financial and temporal burdens associated with obtaining reference measurements for NIR spectral data [10].
- **Facilitating Algorithm Development and Evaluation**: Synthetic spectra can serve as a testing ground for developing and refining algorithms, offering a practical means to trial and improve methodologies [8].

1.2 Dataset and Methodology

This thesis applies generative models to a mango dataset, which includes Dry Matter Content (DMC) as a response variable [11]. While a valuable case study in itself, the primary motivation for selecting this dataset is not its intrinsic value, but its suitability for illustrating the process and feasibility of generating synthetic NIR spectra. This dataset allows us to demonstrate the generation of synthetic spectra without confining ourselves to a specific scope or application. Furthermore, it underscores the potential utility of synthetic spectra in ML tasks, such as predicting specific target variables like DMC [11].

Simultaneously, this work involves a thorough benchmarking process on the synthetic data produced. This important step provides an assessment of the synthetic data quality, offering insights into its potential utility in tasks involving NIR spectral data. It is through this benchmarking process that we aim to validate the synthetic data and demonstrate its potential to serve as a viable complement, or in some cases alternative, to real-world data in NIR spectroscopy tasks.

1.3 Objectives

The key objectives of this thesis include:

- **Model Evaluation**: Assess the proficiency of generative ML models, particularly those housed within the SDV library, in producing synthetic NIR spectral data.
- **Synthesis of Realistic Spectra**: Employ tabular generative models to generate authentic and usable synthetic NIR spectral data from the mango dataset.
- **Predictive Modeling from Synthetic Spectra**: Generate synthetic NIR spectral data robust enough to support the development of effective ML applications within NIR spectroscopy.

- **Benchmarking Synthetic Data**: Undertake a comprehensive evaluation of the quality and performance impact of synthetic NIR spectral data on ML models.
- **Unearthing Limitations**: Identify and articulate the constraints and limitations inherent in the chosen approach for synthetic data generation.
- **Future Research Directions**: Highlight potential avenues for future investigations aimed at enhancing the generative modeling of NIR spectra.

Chapter 2

Theory

2.1 Near infrared spectroscopy

Spectroscopy is a branch of science that deals with the study of interaction between electromagnetic radiation and matter. It involves using electromagnetic radiation to probe the properties of a material, and the information obtained can be used to identify and quantify the chemical composition of a sample. This is done by thoroughly and carefully examining the absorption, scattering and emission of electromagnetic radiation from the inspected compounds [12]. Figure 2.1 illustrates the spectroscopy process.

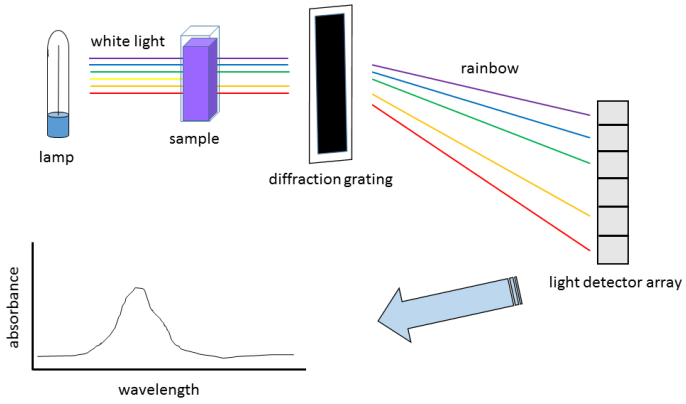


Figure 2.1: Process behing spectroscopy [13]. The process involves interaction between light and matter, where the absorption, emission, or scattering of electromagnetic radiation by the material provides valuable information about its structure or properties. Figure by Wilsonee, distributed under CC-BY-SA-4.0

NIR spectroscopy is a type of spectroscopy that operates in the range of wavelengths between 700 and 2500 nanometers [14]. This range of wavelengths corresponds to the region of the electromagnetic spectrum just beyond the visible range of light. NIR spectroscopy is a non-destructive and non-invasive technique that can be used to analyze a wide range of materials, including solids, liquids, and gases.

The basic principle of NIR spectroscopy is that different chemical bonds absorb light at different wavelengths, specifically through overtones and combinations of the fundamental vibrations

from the mid-infrared range. By measuring the amount of light absorbed at various wavelengths, it is possible to identify the presence and quantity of various chemical functional groups in a sample. This information can be used to quantify the composition of a sample, such as the amount of protein, fat, or carbohydrate in a food sample.
NIR spectroscopy is widely used in a variety of industries, including food and agriculture, pharmaceuticals, and materials science. It is particularly useful for rapid, non-destructive analysis of large numbers of samples. For example, in the food industry, NIR spectroscopy can be used to measure the quality and nutritional content of grains, fruits, and vegetables, without destroying the sample.

In order to perform NIR spectroscopy, a sample is illuminated with a beam of light, and the amount of light absorbed by the sample at different wavelengths is measured using a spectrometer. The resulting spectrum is then analyzed using chemometric techniques, such as Principal Component Analysis (PCA) or Partial Least Squares Regression (PLSR), to extract information about the sample.

Although conventional chemometric methods like Principal Component Analysis and PLSR have demonstrated effectiveness in analyzing NIR spectral data, they have some limitations. Beer-Lambert's law [15] states that the concentration of substances analyzed with NIR should result in a spectrum that is an additive mix of the real spectra of the substances, given proper preprocessing. While this implies a linear relationship, there are cases where complex relationships between spectral features and target variables arise, making non-linear methods more suitable for obtaining better predictions [16]. Additionally, conventional chemometric techniques require the selection of an appropriate number of components, which can be challenging and may affect model performance [17].

2.1.1 Chemometric Techniques for NIR Spectroscopy

Traditional approaches for the analysis of NIR spectra involve chemometric techniques that are widely used to extract meaningful information from complex and high-dimensional spectral data. Some of the most common chemometric techniques for NIR spectroscopy include:

- **PCA:** PCA is an unsupervised technique used for reducing the dimensionality of spectral data and identifying the underlying structure or patterns in the data. It achieves this by transforming the real variables into a new set of orthogonal variables (principal components) that capture the maximum variance in the data [18].
- **PLSR:** PLSR is a popular supervised technique in NIR spectroscopy for building quantitative models relating spectral data (predictors) to the properties of interest (responses). Unlike traditional multiple linear regression, which may suffer from multicollinearity issues when the predictors are highly correlated, PLSR is specifically designed to handle collinear and noisy data. It achieves this by constructing a set of orthogonal latent variables, called PLS components, that capture the maximum covariance between the predictors and responses. In addition to handling multicollinearity, PLSR, like PCA, is effective at filtering out random noise in the data, which further contributes to its robustness when dealing with noisy spectral data. These PLS components are then used to build a linear regression model, which can be employed for predicting the properties of interest based on the spectral data [19].
- **Multiple Linear Regression (MLR):** MLR is a widely used statistical technique that models the relationship between a continuous response variable and multiple predictor variables. In NIR spectroscopy, the response variable represents a property of interest

(e.g., protein content, moisture), while the predictor variables correspond to absorbance values at various wavelengths [20]. MLR aims to establish the best-fitting linear equation relating the predictors to the response variable by minimizing the residual sum of squares, which represents the sum of squared differences between the actual and predicted response values. Although MLR is commonly employed in chemometrics, it may encounter issues like multicollinearity when predictor variables exhibit high correlation [21].

These techniques have been instrumental in the analysis of NIR spectra, but they often rely on linear assumptions and may struggle to capture complex non-linear relationships in the data [22]. ML methods, particularly deep learning techniques, have the potential to overcome these limitations by automatically learning hierarchical representations of the data and capturing non-linear relationships between variables [16, 23].

2.2 Machine Learning

ML, a subfield of artificial intelligence, focuses on developing algorithms and models capable of learning patterns from data and making predictions or decisions based on these learned patterns [24, par. 4.4-4.5]. It has significantly impacted various domains, such as healthcare, finance, marketing, and natural sciences, enabling data-driven decision-making and automating complex tasks [25, 26]. The core aim of ML is to build models that can generalize and adapt to new, unseen data by extracting and representing the underlying structure in the available data. This process typically involves selecting an appropriate model architecture, determining the optimal model parameters through training, and validating the model's performance on a separate set of data. ML models can be broadly categorized into three main types: supervised learning, unsupervised learning, and reinforcement learning [24, par. 4.7].

Supervised learning involves learning a mapping from input data to output labels using a labeled dataset, and it is commonly used for classification and regression tasks. In classification, the goal is to assign discrete labels or categories to input data points based on their features. With regression the objective is to predict a continuous numerical value based on input features. Unsupervised learning, on the other hand, does not rely on labeled data and instead aims to discover hidden patterns or structures within the input data, making it suitable for tasks such as clustering, dimensionality reduction, and data synthesis. Reinforcement learning focuses on training agents to make decisions based on interactions with their environment, learning to maximize a reward signal over time. This type of learning is particularly well-suited for control and optimization problems, where the goal is to learn an optimal policy for decision-making in dynamic environments [24, par. 4.21-4.24].

In addition to the aforementioned ML categories, two major subcategories in ML are deep learning and shallow learning, differentiated primarily by the complexity and depth of their model architectures [27, pg. 8].

Artificial Neural Network (ANN) provide the foundational framework for deep learning models. ANNs are computational models inspired by biological neural networks, consisting of a series of interconnected nodes or neurons organized in layers. Each neuron processes inputs from the previous layer, applies a weighted sum to these inputs, and then applies an activation function to produce an output that is passed to the next layer. The learning process in ANNs, known as backpropagation, adjusts the weights of the connections to minimize the error between the network's predictions and the ground truth by propagating the error gradients backward through the network layers [28, 29]. The invention of backpropagation marked a significant milestone in neural network training, as it facilitated efficient training of multi-layered large

networks [27][par. 1.2.2]. In other words, this innovation made the concept of deep learning feasible. This increased depth allows them to automatically learn hierarchical representations of the data and capture complex, non-linear relationships between variables [30, 31].

Specific types of ANN are Fully-Connected Neural Network (FCNN), Convolutional Neural Network (CNN), and Recurrent Neural Network (RNN). FCNNs, also known as Multilayered Perceptron (MLP), contain multiple layers where each neuron in a layer is connected to all neurons in the previous layer. These types of networks are widely used due to their simplicity and general capability. An example of a FCNN can be seen in Figure 2.2

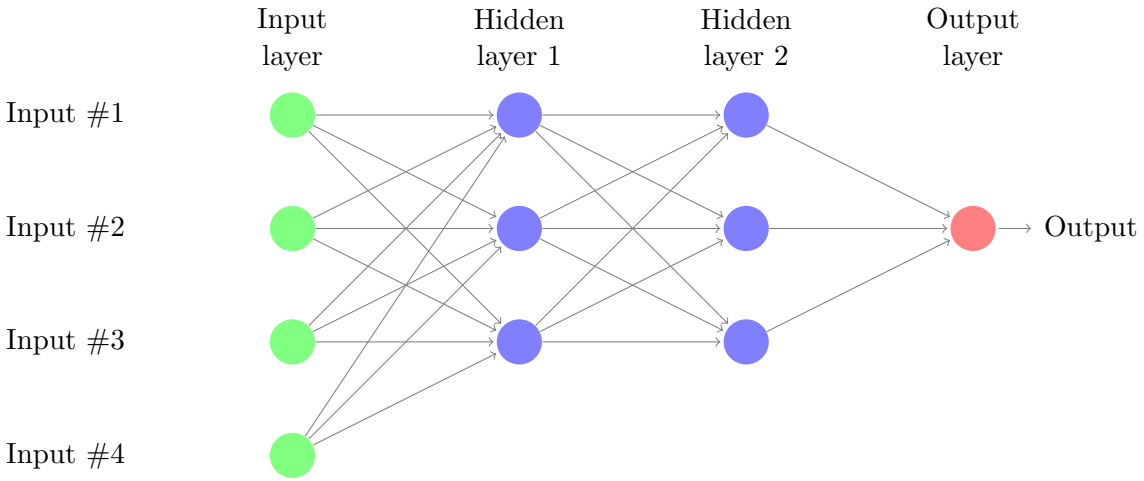


Figure 2.2: Illustration of Fully-Connected Neural Network, with two hidden layers

Deep learning models like CNNs and RNNs extend the foundational structure of FCNNs. CNNs, drawing inspiration from the animal visual cortex, employ filters to systematically capture local patterns, making them particularly adept at processing high-dimensional and unstructured data, such as images. RNNs, conversely, are designed to handle sequential data, utilizing loops within the network to maintain information across the sequence, which proves useful for tasks involving time-series data or text.

However, these deep learning models, including FCNNs, usually require extensive data and computational resources for effective training [32]. Additionally, their interpretability can be challenging due to the complexity of the learned representations, making it difficult to understand how they make their predictions [33].

Shallow learning techniques, on the other hand, involve models with a limited number of layers or computational steps, such as Support Vector Machines, Random Forests, and k-Nearest Neighbors. These models can also capture non-linear relationships in the data but do not automatically learn hierarchical representations the way deep learning models do, which stack multiple layers to extract increasingly complex features from the input data [27, pg. 18]. While they may require more feature engineering compared to deep learning techniques [27, pg. 18], shallow learning methods have proven effective for various tasks and usually offer simpler and more interpretable models [34], depending on the specific problem and dataset. These models are generally more computationally efficient and require less data for training compared to their deep learning counterparts, making them suitable for applications with limited resources or smaller datasets [35].

2.2.1 Machine Learning in NIR Spectroscopy

ML methods, especially deep learning techniques, can address the limitations of traditional chemometric approaches by automatically learning hierarchical representations and capturing non-linear relationships between variables [16, 23]. These techniques, which encompass supervised and unsupervised learning methods, offer the potential to improve chemometric capabilities by providing more accurate and robust predictions through the recognition of complex patterns in the data.

In the context of NIR spectroscopy, supervised learning techniques, such as Support Vector Machines and Random Forests, have been employed for building predictive models that relate spectral data to properties of interest, such as chemical composition or physical attributes [36, 37]. Meanwhile, unsupervised learning techniques, including clustering algorithms like K-Means, have proven valuable for data exploration, dimensionality reduction, and grouping similar samples, which can reveal the underlying structure in the data and facilitate further analysis [38].

When applying ML to NIR spectroscopy, it is essential to consider the importance of data preprocessing. Techniques such as normalization, scaling, baseline correction, and smoothing can help mitigate the effects of noise, varying instrument conditions, and other sources of variability in the spectral data [39]. In NIR spectroscopy, normalization is the process of adjusting the intensity of the spectra so that they all share a common scale, while smoothing is the technique of reducing noise or fluctuations in the spectral data to reveal underlying trends or features more clearly. Both techniques are important for mitigating the effects of noise, varying instrument conditions, and other sources of variability in spectral data.

One popular smoothing technique is the Savitzky-Golay (SG) method, which employs a moving polynomial fit to reduce high-frequency noise while preserving important features in the data, such as peaks and valleys. By enhancing the signal-to-noise ratio using the SG smoothing technique, the performance and interpretability of the ML models being applied to NIR spectroscopic data can be improved [40]. Figure 2.3 shows how effective SG is at estimating the trend in noisy data.

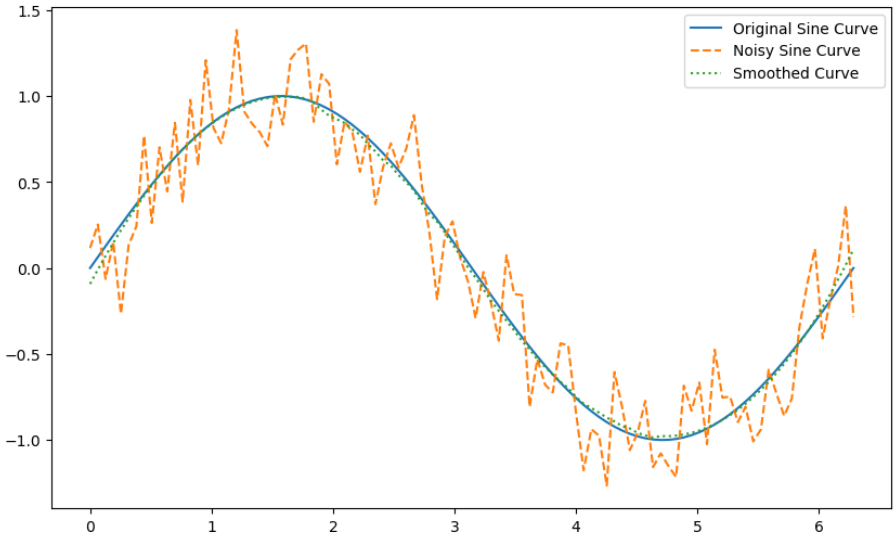


Figure 2.3: Illustration of SG filter applied to a sine curve with added noise. It demonstrates how efficient it is at estimating the trend, the estimated curve strongly overlapping with the original sine curve

Building on the advances in ML and computational capabilities, more complex models and algorithms have been explored for NIR spectroscopy, such as the aforementioned deep learning techniques CNNs and RNNs. These advanced techniques have demonstrated significant potential in enhancing the analysis and interpretation of NIR data, leading to more accurate, efficient, and reliable applications across diverse industries [23]. However, it is important to note that the interpretation of the underlying relationships in the data can be more challenging with these deep learning techniques compared to traditional shallow ML methods and PCA/PLSR.

2.2.2 Shallow Machine Learning Techniques for NIR Spectroscopy

Shallow ML techniques have gained popularity in the field of NIR spectroscopy due to their effectiveness in processing high-dimensional spectral data and adaptability for diverse applications. These methods, including Random Forest (RF), Support Vector Machine (SVM), and K-Nearest Neighbors (KNN), require less computational resources than deep learning models, making them practical for modeling data structure. They have proven effective in capturing non-linear relationships between variables, yielding accurate predictions in numerous spectroscopy tasks [41–43].

- **RF:** An ensemble learning method, RF constructs multiple decision trees during training and combines their outputs for improved prediction accuracy and overfitting control. Effective in handling high-dimensional data, it naturally models non-linear relationships between variables [24, par. 6.182-6.193]. Applied to classification and regression tasks within this field, it has outperformed traditional chemometric methods in several studies [37, 41, 44].
- **SVM:** SVMs are ML models capable of linear or non-linear classification and regression tasks by identifying the optimal hyperplane or decision boundary separating the data. Effective with high-dimensional data, they have robust generalization capabilities [24, par. 6.112-6.153]. In the realm of NIR spectroscopy, SVMs have been utilized for various applications, achieving impressive results and surpassing traditional chemometric methods in multiple studies [16, 36, 45–48].
- **KNN:** KNN, a non-parametric and instance-based learning algorithm, is used for classification and regression tasks. It predicts outputs by finding the K training samples closest to a new input and considering the majority vote or average value of these neighbors [24, par. 6.194-6.208]. Within the NIR spectroscopy domain, KNN has been employed for various applications, demonstrating promising results compared to traditional chemometric methods in some studies [47–50].

2.2.3 Deep Machine Learning Techniques for NIR Spectroscopy

Deep learning techniques have emerged as a natural progression of ML advancements, offering powerful alternatives to traditional chemometric methods for analyzing complex and high-dimensional NIR spectral data. These models are particularly adept at automatically learning hierarchical representations of the data and capturing non-linear relationships between variables, making them highly suitable for various NIR spectroscopy applications. Furthermore, deep learning models can extract patterns from raw spectra with minimal feature engineering, streamlining the analysis process [23].

- **CNN:** CNNs are a class of deep learning models specifically designed for handling grid-like data, such as images or time-series data. They consist of multiple layers of convolutional filters followed by pooling layers, which enable the model to learn local features and spatial hierarchies in the data [25]. Figure 2.4 shows the process flow for a typical CNN. In the

context of NIR spectroscopy, CNNs can be employed to learn local patterns in the spectral data, which can then be used for tasks such as classification, regression, or anomaly detection. Several studies have demonstrated the effectiveness of CNNs for analyzing NIR spectra, achieving improved performance compared to traditional chemometric methods [5, 6].

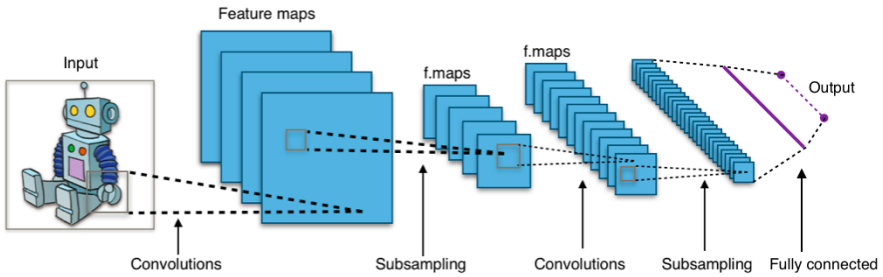


Figure 2.4: Depiction a generic CNN, showing an input image passing through multiple convolutional and pooling layers, which extract hierarchical features, followed by fully connected layers that output the final classification or prediction [51]. Figure by Aphex34, distributed under CC-BY-SA-4.0

- **RNN:** RNNs are a class of deep learning models designed to handle sequential data, making them particularly suitable for time-series analysis or data with temporal dependencies. RNNs consist of recurrent layers that maintain a hidden state, allowing the model to capture long-range dependencies and learn patterns across time [52]. For NIR spectroscopy, RNNs can be used to model the sequential nature of spectral data, capturing dependencies between neighboring wavelengths, and improving the accuracy of predictions [53]. Several studies have explored the use of RNNs and their variants, such as Long Short-Term Memory (LSTM) networks, for NIR spectroscopy applications, demonstrating their potential for various tasks [53, 54]. Figure 2.5 illustrates the fundamental principle underpinning RNNs.

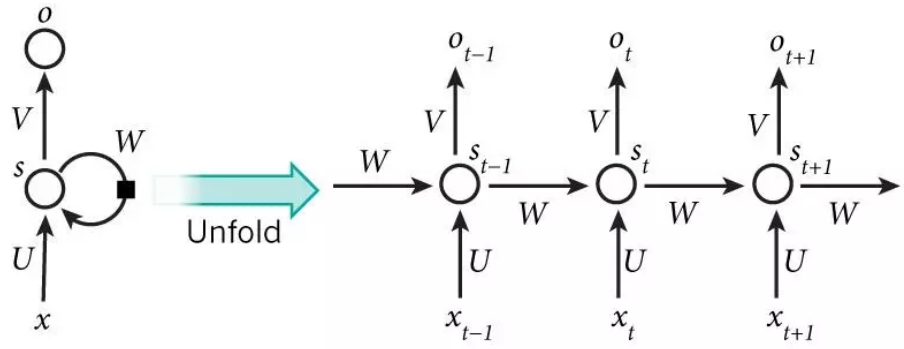


Figure 2.5: Portrayal of a generic RNN, where sequential data is processed through interconnected layers that loop back on themselves, enabling the network to retain information from previous inputs as it makes predictions for the current step in the sequence. [55]. Figure by MingxianLin, distributed under CC-BY-SA-4.0

Deep learning techniques like CNNs and RNNs have shown great promise in the analysis of NIR spectral data, offering improved performance over traditional chemometric methods.

2.3 Generative modeling

In recent years, various ML techniques have been employed to generate synthetic data, which can be particularly valuable in addressing data scarcity and improving model performance across a range of applications [9]. By harnessing the power of ML, it is possible to create realistic and diverse artificial datasets that can augment existing data and facilitate the development of more robust algorithms tailored for specific domains, such as NIR spectral data analysis.

Generative ML models are a class of techniques that focus on learning the underlying probability distribution of the data, aiming to produce new instances that closely mirror the real data [56, pg.27-29]. These models have garnered significant interest lately for their capacity to generate authentic-seeming data. This synthetic data can be utilized to mitigate issues of data scarcity, enrich data variety, and bolster the efficacy of ML methods across a range of applications. [9].

In contrast to discriminative models, which aim to model the conditional probability of the target outputs given the input data, generative models attempt to capture the joint probability distribution of both the inputs and outputs [57, preface]. By learning this joint distribution, generative models are able to produce new data points by sampling from the estimated distribution. This ability to generate new data points has made generative models particularly valuable for tasks such as data augmentation [9], unsupervised learning [58], and representation learning [56, pg. 72-79]. Figure 2.6 illustrates the difference between discriminative and generative modeling

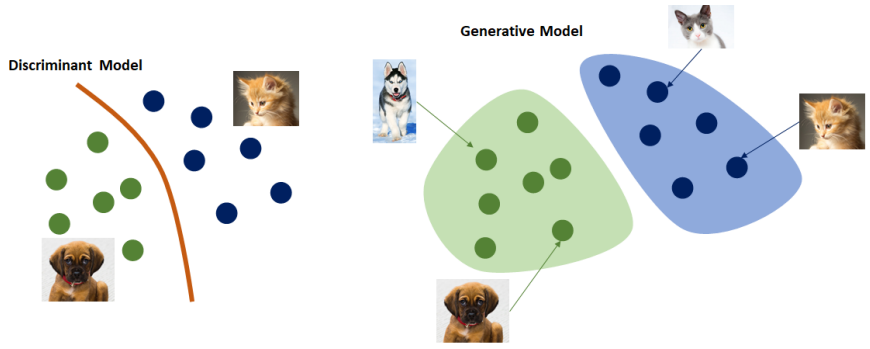


Figure 2.6: Illustration of the main difference between discriminative and generative modeling: Discriminative modeling focuses on the boundary between classes, while generative modeling learns the distribution of individual classes [59]. Figure by Jordi Esteve Sorribas, distributed under CC-BY-SA-4.0

Generative models are particularly proficient in comprehending and replicating the patterns present in their training data. This unique characteristic enables them to synthesize novel data that maintains significant resemblance to the original, while incorporating distinctive elements. Moreover, generative models have the distinct capacity to investigate what is referred to as the 'latent space' of the data. The concept of latent space can be analogized to a simplified or abstracted representation of the data in a lower-dimensional space, retaining only the most important characteristics. This condensed representation can expose concealed patterns and relationships within the data, thereby enhancing our comprehension of the data and aiding in the identification of salient features for subsequent analytical tasks [60].

Various types of generative models exist, each with unique advantages and drawbacks, making the choice of model dependent on the specific problem and data at hand. Popular generative models include Bayesian models, such as Gaussian Mixture Model (GMM) and Hidden Markov

Model (HMM) [61], as well as deep learning-based models like VAEs and GAN [62, 63], which have achieved high levels of performance.

Bayesian models, a class of generative models based on the principles of Bayesian statistics, involve estimating probability distributions using prior knowledge and observed data. Their objective is to learn the probability distribution of the real data, allowing them to produce new samples that closely align with the real data. These models offer high interpretability and flexibility, but their performance can be constrained by the choice of priors and the computational complexity of the inference process. Alongside deep learning-based models, Bayesian generative models such as GMMs and HMMs have been widely used across various applications [64].

Deep learning-based generative models, on the other hand, leverage the power of neural networks to learn complex, non-linear representations of the data. These models can capture high-dimensional and intricate structures in the data, but their training can be challenging due to the optimization of deep architectures and the need for large amounts of training data [65, 66].

2.3.1 Generative Adversarial Networks

GAN, an intriguing class of generative models, was first introduced by Ian Goodfellow and his colleagues in 2014, captivating the research community with their ability to generate high-quality synthetic data [62]. Structurally, a GAN consists of two concurrent neural networks — the generator and the discriminator — that engage in a game-like adversarial training process.

The generator network, denoted as G, begins its operation by receiving a random noise vector, z, as input. Its mission is to transform this noise into a synthetic data sample, G(z), that mimics real data so closely that it becomes indistinguishable.

On the other hand, the discriminator network, D, is tasked with the job of a vigilant gatekeeper. It receives both real data samples, x, and the synthetic samples, G(z), generated by its adversary. For each input, the discriminator must decide whether it's real or artificially produced by the generator, expressing its verdict as a probability value. Thus, while the generator constantly endeavors to create convincing fakes, the discriminator hones its ability to discern real from synthetic.

The training process of GANs can therefore be characterized as a minimax game, where the generator and discriminator networks are optimized using the following objective function:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]$$

Here, $p_{data}(x)$ denotes the real data distribution, and $p_z(z)$ represents the noise distribution. The generator aims to minimize the objective function, while the discriminator tries to maximize it.

During the training process, the generator and discriminator networks are updated using backpropagation and gradient descent. The discriminator is trained to improve its ability to differentiate between real and generated samples, and the generator is trained to produce more realistic samples that can deceive the discriminator. The training continues until an equilibrium is reached, where the generator produces samples that the discriminator cannot distinguish from the real data [62].
As Refer to Figure 2.7 for an illustration of the process.
which are trained concurrently within a competitive setting akin to a zero-sum game.

The generator is responsible for fabricating synthetic data instances, whereas the discriminator's role is to distinguish between real and synthetic samples.

""" The generator network, G, takes a random noise vector, z, as input and generates a synthetic data sample, G(z). The goal of the generator is to produce samples that are indistinguishable from the real data. The discriminator network, D, takes both real data samples, x, and generated samples, G(z), as input and outputs a probability value representing its confidence in whether the input sample is real or generated. The goal of the discriminator is to correctly identify real data samples and distinguish them from the generated samples. The process is showed in Figure 2.7 """

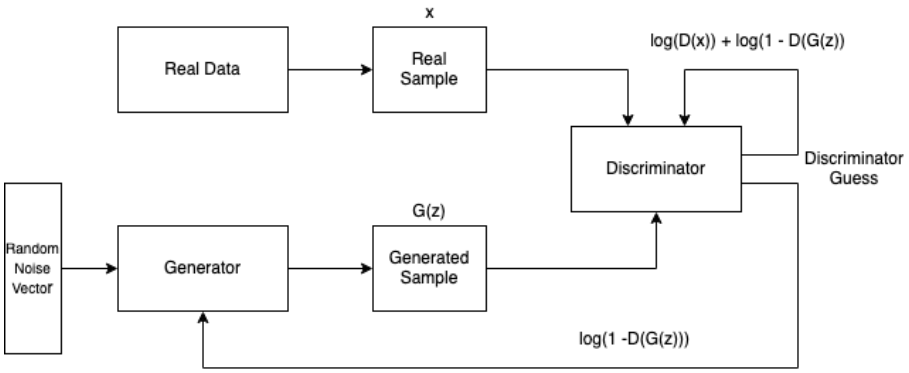


Figure 2.7: Training process of vanilla GAN. The generator is fed a random noise vector, which makes the GAN generate a sample. The discriminator is fed both generated and real samples, and then tries to guess if a sample is real or synthetic.

The training process of GANs is based on a minimax game, where the generator and discriminator networks are optimized using the following objective function:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]$$

Here, $p_{data}(x)$ denotes the real data distribution, and $p_z(z)$ represents the noise distribution. The generator aims to minimize the objective function, while the discriminator tries to maximize it.

During the training process, the generator and discriminator networks are updated using backpropagation and gradient descent. The discriminator is trained to improve its ability to differentiate between real and generated samples, and the generator is trained to produce more realistic samples that can deceive the discriminator. The training continues until an equilibrium is reached, where the generator produces samples that the discriminator cannot distinguish from the real data [62].

GANs have demonstrated impressive results in various applications, such as image synthesis, style transfer, and data augmentation [67]. However, they also face some known challenges. One such challenge is mode collapse, where the generator produces only a limited variety of samples, focusing on a few modes of the data distribution rather than capturing the full diversity of the real data [68].

Another challenge is training instability, which can cause oscillations and divergence in the learning process [69]. This can occur due to factors such as vanishing gradients, when the discriminator becomes too strong and can easily distinguish between real and generated samples,

making it difficult for the generator to improve, and non-convergence, where the training process may not converge to an equilibrium if the learning rates, model architectures, or training procedures are not properly balanced [69, 70].

2.3.2 Variational Autoencoders

VAEs are a powerful class of generative models that combine deep learning with Bayesian inference to learn the underlying probability distribution of the data and generate new samples that closely resemble the real data. VAEs were introduced by Kingma and Welling in their seminal paper "Auto-Encoding Variational Bayes" [63]. Before diving into the specifics of VAEs, it is essential to understand the foundations of autoencoders, which form the basis for the encoder-decoder principle and data reconstruction.

Autoencoders are a type of unsupervised learning model that aim to learn a compact and efficient representation of input data by encoding it into a lower-dimensional latent space and then reconstructing the real data from this latent representation. This process involves two main components: an encoder network that maps the input data to the latent space, and a decoder network that reconstructs the real data from the latent representation [71].

Building upon the basic concept of autoencoders, VAEs consist of two main components: an encoder network and a decoder network, which can be seen in the. The basic architecture can be seen in Figure 2.8. The encoder network takes input data and maps it to a latent space representation, which is a lower-dimensional continuous space that captures the essential features of the data. The encoder network models the approximate posterior distribution over the latent variables given the input data, denoted as $q_\phi(z|x)$, where z represents the latent variables, x represents the input data, and ϕ are the parameters of the encoder network.

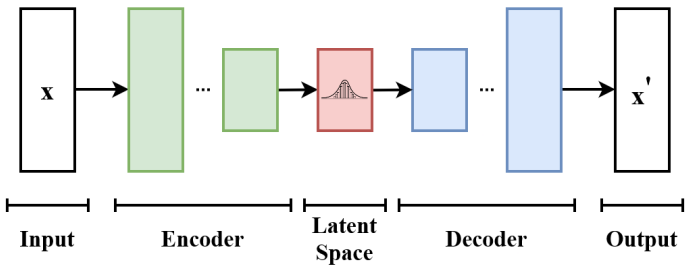


Figure 2.8: Architecture of VAE [72]; An encoder receives a random noise vector as input, and compresses it to the latent space with the constraint that it is normally distributed. The decoder then decodes the vector to a sample that resembles real data. Figure by EugenioTL, distributed under CC-BY-SA-4.0

The decoder network, on the other hand, takes a sample from the latent space and reconstructs the real data. The decoder models the likelihood of the data given the latent variables, denoted as $p_\theta(x|z)$, where θ are the parameters of the decoder network.

VAEs are trained by maximizing the Evidence Lower Bound (ELBO) on the marginal likelihood of the data. The ELBO is given by:

$$\mathcal{L}(\theta, \phi; x) = \mathbb{E}q_\phi(z|x)[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x)||p(z))$$

where D_{KL} represents the Kullback-Leibler divergence between the approximate posterior $q_\phi(z|x)$ and the prior distribution $p(z)$ on the latent variables. The first term in the ELBO measures the reconstruction error, while the second term acts as a regularization term that encourages the approximate posterior to be close to the prior distribution.

By optimizing the ELBO, VAEs learn to generate new samples that are similar to the real data while ensuring that the learned latent space is smooth and well-structured. This allows VAEs to generate diverse and realistic synthetic data, which can be employed to ameliorate lack of data, enhance data diversity, and improve the performance of ML algorithms in various applications [73].

2.3.3 CTGAN and TVAE

CTGAN and TVAE are two deep learning models designed for generating synthetic tabular data [4]. They are both capable of modeling complex relationships and distributions of columns in tabular data and have been shown to outperform other models on real datasets in various benchmark tests.

CTGAN, is a model that incorporates the benefits of GANs and conditional generation to effectively synthesize tabular data. In the realm of generative models, conditional generation pertains to the generation of data samples that adhere to specific input conditions or constraints, providing more targeted and controlled data synthesis [74].

CTGAN treats continuous and categorical features differently, employing a specific encoding procedure for each. For continuous variables, CTGAN uses mode-specific normalization. Mode-specific normalization plays an important role in accurately representing continuous values with intricate distributions. This approach tackles each continuous column independently, employing a Variational Gaussian Mixtures (VGM) to estimate the number of modes and fit a Gaussian mixture. The probabilities of values belonging to each mode are calculated based on the Gaussian mixture, and a mode is sampled accordingly. This sampled mode is then used to normalize the value by representing it as a one-hot vector indicating the chosen mode and a scalar denoting the value within that mode. The resulting row representation is formed by concatenating the normalized continuous and discrete columns. By following this process, the mode-specific normalization in CTGAN ensures that complex distributions of continuous values are appropriately handled, thereby enhancing the accuracy of generated synthetic tabular data [4].

For categorical variables, CTGAN adds a conditional vector, fostering a method of conditionality that enhances its ability to handle discrete columns and deal with data imbalance. This ability to handle varying data types makes CTGAN an effective model for synthesizing tabular data.

Both the generator and the discriminator in CTGAN is composed of a FCNN. They are trained using a wasserstein loss with gradient penalty. This configuration allows for robust and efficient training of the model, further improving its ability to generate high-quality synthetic data [75].

Additionally, CTGAN features a conditional generator and training-by-sampling components, which are specifically designed to tackle issues related to imbalanced training data. These components enable the generation of data with specific discrete values and allow for targeted augmentation of underrepresented categories or classes in the data. By addressing data imbalance and enabling controlled generation of tabular data samples, CTGAN proves to be a useful tool for data augmentation tasks, particularly when working with datasets that have limited

samples or display significant class imbalance.

Shaped by the core strengths of VAE, TVAE has been tailored to effectively model tabular data. It parallels CTGAN in its ability to handle intricate relationships in such data, demonstrating solid performance across diverse benchmark tests. While CTGAN can occasionally outstrip TVAE, each brings its unique advantages to the table, and the selection largely hinges on the distinct requirements of the task or application at hand. For instance, differential privacy, a desirable trait in many applications, may be more readily achieved with CTGAN as its generator is not privy to real data during the training phase [75].

TVAE, an adaptation of the conventional VAE, is fine-tuned to handle tabular data. Like its CTGAN counterpart, it differentiates between categorical and continuous variables, employing an encoding process similar to that utilized by CTGAN. In the case of continuous variables, it adopts a VGM model to transform values with arbitrary distributions into a bounded vector representation, thus aligning well with the structure of the neural network and enabling the accurate capture of inherent patterns in the data.

In TVAE, the encoder is slightly modified compared to traditional VAEs, while the decoder retains a standard structure. The loss function also adopts a nuanced approach, incorporating the ELBO loss, a fundamental component of VAEs that strikes a balance between data reconstruction and latent variable regularization.

When juxtaposed with other tabular data synthesis methods such as tableGAN, both TVAE and CTGAN have demonstrated superior performance, with TVAE excelling in several metrics. However, it's crucial to note that CTGAN may hold an upper hand when it comes to ensuring differential privacy, given that its generator does not interact with real data during the training process. Thus, the choice between TVAE and CTGAN is largely influenced by the specific needs of a task or application, with each model offering distinct benefits.

Considering application in NIR spectroscopy, both CTGAN and TVAE could potentially be applied to generate synthetic spectral data. By modeling the complex relationships between spectral features and sample properties, these deep learning models can generate realistic and diverse synthetic NIR spectra that closely resemble real-world data. This augmentation of existing data can lead to improved model performance and facilitate the development of more robust algorithms tailored for specific applications in NIR spectral data analysis.

NIR spectra often exhibit strong collinearity among variables because of very high intercorrelation between absorbances [76]. In addition, NIR spectra data do not involve sensitive or private information, eliminating privacy concerns that might be relevant in other applications of synthetic data generation.

Given these unique properties of NIR spectra, some aspects of CTGAN and TVAE might be less relevant or applicable. For instance, CTGAN's ability to handle discrete columns and its potential for differential privacy may not be particularly important when working with continuous NIR spectra data without privacy constraints.

However, CTGAN and TVAE have the potential to work well with NIR spectroscopy data due to their ability to model complex, high-dimensional continuous data. In particular, these deep learning models excel at capturing intricate relationships and dependencies between variables, which can be advantageous when dealing with the strong collinearity often found in NIR spectra. Furthermore, CTGAN's mode-specific normalization and TVAE's powerful latent variable

modeling can effectively handle continuous data with varying ranges and distributions, such as those found in NIR spectra. Hence, despite initial concerns regarding the relevance or applicability of CTGAN and TVAE when it comes to NIR spectra, these models may possess the potential to generate high-quality synthetic data that replicates the essential properties of the real NIR spectra.

2.3.4 Applications of Generative Models in NIR Spectroscopy

Generative modeling has emerged as a promising approach for data augmentation in NIR spectroscopy. Researchers have applied this technique to improve the quality and quantity of available data, leading to better performance in various applications.

Nagasawa et al. focused on enhancing fNIRS-BCI accuracy using a data augmentation method based on Wasserstein GAN (WGAN) [77]. In their study, they evaluated the effectiveness of the generated artificial fNIRS data in improving the classification performance of support vector machines and simple neural networks. The results indicated that the generated data, when combined with the real training data, led to improved classification accuracy across different task types.

Dehua Zhu et al. explored the use of boundary equilibrium GAN (BEGAN) to generate synthetic spectra, particularly for applications with limited calibration samples [78]. They demonstrated that the synthetic spectra produced by BEGAN maintained high quality and improved the predictive performance of a consensus algorithm called creating diversity partial least squares (CDPLS). The study also found a negative correlation between the quality and diversity of the generated spectra, indicating that adjusting the diversity ratio can help control these factors according to the specific needs of an application. The success of BEGAN in generating high-quality synthetic spectra for small sample sets highlights its potential for expanding the number of spectra and enhancing the performance of spectral analysis models.

Kaixun He et al. proposed a new modeling method based on WGANs to overcome the challenges of acquiring a sufficient number of labeled samples for NIR spectroscopy, particularly for predicting octane numbers in gasoline blending [79]. They demonstrated that WGAN, combined with a sample selection method, could generate artificial labeled samples that, when used alongside real samples, improved the performance of NIR models in predicting octane numbers.

Considering the success of generative models in NIR spectroscopy, it is worth exploring the use of tabular data generation tools like CTGAN and TVAE. These methods have been successful in generating synthetic data for various applications, and their ability to capture complex data distributions makes them suitable for NIR spectroscopy as well. By utilizing these tools, researchers can potentially generate high-quality synthetic data, which can then be used to improve the performance of various NIR spectroscopy applications. In conclusion, the advancements in generative modeling, combined with the adaptability of tabular data generation tools like CTGAN and TVAE, present promising opportunities for further improving NIR spectroscopy data augmentation and performance.

2.4 Evaluation of Synthetic Data

Synthetic data is data that has been generated by a computer. While the concept does not have formal or universal definition, James Jordon et al. offers the following definiton: "*Synthetic data is data that has been generated using a purpose- built mathematical model or algorithm, with the aim of solving a (set of) data science task(s)*" [80]. The goal with synthetic data is often

to approximate real-world data in terms of its underlying structure, statistical properties, and relationships among variables. It has gained significant attention in recent years due to its potential applications in a wide range of areas, including data augmentation, addressing data scarcity challenges, improving model performance, and, in some cases, privacy preservation. The generation of synthetic data often involves employing advanced algorithms and techniques that are capable of learning the patterns and dependencies present in the real data to produce realistic and diverse synthetic samples.

Evaluating synthetic data typically involves assessing its fidelity and utility [81]. Fidelity is a measure of how closely the synthetic data matches the real data, reflecting the extent to which the produced data maintains the fundamental characteristics and structure of the real data. On the contrary, utility signifies the practical value of the synthetic data, for instance, its application in training ML algorithms and enhancing their efficiency in real-world operations.

It is worth noting that while privacy preservation is a key aspect of synthetic data generation in certain contexts, particularly when dealing with sensitive or personally identifiable information, in the case of NIR spectroscopy, privacy concerns are generally less relevant as the data often does not involve sensitive or private information. This happens to be the case for our data too. As such, the main focus when evaluating synthetic NIR spectroscopy data is to ensure that it maintains the critical properties of the real data and proves useful in practical applications.

2.4.1 Fidelity

In terms of synthetic data, fidelity denotes the extent of similarity between the real data and the data that has been synthesized [80, 81]. This concept assesses how accurately the artificially generated data mimics the fundamental structure, relationships, and traits of the original dataset. A substantial degree of fidelity suggests that the generative model has effectively generated synthetic data that retains the attributes and trends of the real data, thereby making the synthetic data a dependable alternative for a range of analyses and uses.

Assessing fidelity is an important aspect when evaluating synthetic data, as it helps determine the quality and usefulness of the generated data in replicating or augmenting the real dataset. By evaluating fidelity, researchers can establish the effectiveness of the generative model in producing realistic and high-quality synthetic data, which is critical for maintaining the validity and usefulness of the synthetic data in various applications. This assessment ultimately leads to more robust algorithms and improved performance in data-driven applications.

Cosine Similarity and Pearson Correlation: Key Metrics for Assessing Fidelity

When evaluating the fidelity of synthetic data, it is essential to compare the linear relationships among the features and between the target and features in both the real and synthetic datasets. Two key metrics used for this purpose are cosine similarity and Pearson correlation.

Cosine similarity is a metric that measures the cosine of the angle between two non-zero vectors in a multi-dimensional space. This results in a value ranging from -1 to 1. A cosine similarity closer to 1 signifies that the directions of the two vectors are more aligned, suggesting a high degree of similarity. Conversely, a value closer to -1 indicates that the vectors are in opposite directions, implying dissimilarity, while a value near 0 means the vectors are orthogonal or dissimilar. The cosine similarity is calculated as follows:

$$\text{cosine similarity} = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}} \tag{2.1}$$

In the formula, $\mathbf{A}$ and $\mathbf{B}$ are the two vectors being compared, n is the dimension of the vectors, and A_i and B_i are the components of the vectors $\mathbf{A}$ and $\mathbf{B}$, respectively.

Pearson correlation is a measure of the linear relationship between two continuous variables [82]. It ranges from -1 to 1, with a value of 1 indicating a perfect positive linear relationship, -1 indicating a perfect negative linear relationship, and 0 suggesting no linear relationship. The Pearson correlation coefficient between two variables X and Y can be calculated using the following formula: $\text{Pearson}(X, Y) = \frac{\text{Cov}(X,Y)}{\sigma_X \times \sigma_Y}$, where $\text{Cov}(X, Y)$ represents the covariance between the variables, and σ_X and σ_Y denote the standard deviations of the respective variables.

By computing the Pearson correlation coefficients for the feature-feature and target-feature pairs in both the real and synthetic datasets and comparing these coefficients using cosine similarity, researchers can assess the fidelity of the generated data. High cosine similarity values indicate that the synthetic data effectively captures the underlying structure and relationships present in the real data, signifying the generative model's success in producing realistic and high-quality synthetic data.

Univariate Resemblance

Univariate resemblance refers to the similarity between the real and synthetic data in terms of their individual features [81]. Evaluating univariate resemblance is an essential step in assessing the fidelity of synthetic data, as it ensures that each feature in the generated data maintains the same distribution and statistical properties as its counterpart in the real dataset. By comparing summary statistics, such as mean, median, standard deviation, and other measures of central tendency and dispersion, researchers can determine whether the synthetic data adequately captures the properties of the real data for each feature. Additionally, univariate resemblance can be assessed using statistical tests such as the Kolmogorov-Smirnov (KS) test, Student's t-test, and Mann-Whitney U (MWU) test, which help in comparing the distributions of the real and synthetic data. High univariate resemblance indicates that the generative model has successfully learned the individual characteristics of each feature, which is critical for maintaining the overall quality and usefulness of the synthetic data in various applications.

Aggregate Statistics Aggregate statistics are summary measures that describe the central tendencies, dispersion, and shape of a dataset's distribution. These statistics can provide valuable insights into the univariate resemblance between the real and synthetic datasets by offering a straightforward comparison of their individual features.

- Mean: The arithmetic average of the data points in a feature. It represents the central location of the distribution and is sensitive to extreme values (outliers).
- Median: The middle value of a dataset when sorted in ascending order. It represents the central location of the distribution and is less sensitive to outliers compared to the mean.
- Mode: The most frequently occurring value in a dataset. It indicates the peak of the distribution.
- Standard Deviation: A measure of the dispersion or spread of a dataset. A higher standard deviation indicates a greater degree of variability among the data points.

- Skewness: A measure of the asymmetry of a dataset's distribution. Positive skewness indicates that the tail on the right side is longer or fatter, while negative skewness implies that the tail on the left side is longer or fatter.
- Kurtosis: A measure of the "tailedness" of a dataset's distribution. High kurtosis indicates heavy tails and more outliers, whereas low kurtosis suggests light tails and fewer outliers.

Researchers can compute these aggregate statistics for each feature in both the real and synthetic datasets to evaluate the fidelity of synthetic data. By comparing the values of these statistics, they can assess the univariate resemblance between the datasets and determine the extent to which the generative model has captured the individual characteristics of each feature.

A substantial similarity in the overarching statistics between the real and synthetic datasets indicates the generative model's success in encapsulating the central tendencies, variance, and distribution form of the real data. This feature is an integral aspect of fidelity, as it guarantees that the synthetic data sustains the fundamental properties of the real data, an aspect that is indispensable for its practical implementations.

KS Test The KS test is a non-parametric method, which means it makes no assumptions about the underlying distribution of the data [83]. It can be used to compare the distributions of two samples or a sample against a reference distribution [83]. In the context of evaluating the fidelity of synthetic data, the KS test is employed to assess the univariate resemblance between the real and generated data. The test measures the maximum difference between the empirical cumulative distribution functions (ECDFs) of the two samples, providing a test statistic, D, which represents the greatest absolute difference between the ECDFs.

The null hypothesis of the KS test states that the two samples are drawn from the same continuous distribution. A low p value (< 0.05) indicates that the null hypothesis can be rejected, implying a significant difference between the distributions of the real and synthetic data for a given feature. Conversely, a high p value (≥ 0.05) suggests that the null hypothesis cannot be rejected, meaning the distributions of the real and synthetic data are not significantly different.

The KS test is particularly useful for assessing the fidelity of synthetic data, as it is sensitive to both location and shape differences in the distributions. By applying the KS test to each feature in the real and synthetic datasets, researchers can effectively evaluate the univariate resemblance and determine whether the generative model has successfully captured the individual characteristics of each feature.

Student T-test The Student's t-test is a parametric statistical method used to compare the means of two independent samples, assuming these samples have a normal distribution with equal variances [84]. When assessing the quality of synthetic data, the Student's t-test is used to measure the similarity between the real and generated data by checking if there's a significant difference in the averages of each attribute.

The null hypothesis of the Student's t-test is that the averages of the two samples are the same. A small p value (less than 0.05) suggests that we can reject the null hypothesis, indicating a meaningful difference between the averages of the real and synthetic data for a certain attribute. However, a large p value (equal to or greater than 0.05) means that we can't reject the null hypothesis, implying that the averages of the real and synthetic data are not significantly different.

It is essential to note that the Student's t-test relies on the assumption that the data follows a normal distribution, and the variances of the two samples are equal. If these assumptions are

not met, the results may not be reliable. In such cases, non-parametric tests like the MWU test may be more appropriate.

By applying the Student's t-test to each feature in the real and synthetic datasets, researchers can evaluate the univariate resemblance and determine whether the generative model has successfully captured the mean values of each feature. This analysis helps in understanding if the synthetic data maintains the central tendencies of the real data.

Mann-Whitney U-test The MWU-test, sometimes referred to as the Wilcoxon rank-sum test, is a type of non-parametric statistical test used to compare the distributions of two independent sets of data [85]. This test does not assume anything about the distribution of the data, which makes it a good choice for situations where the data doesn't fit a normal distribution or when the requirement of equal variances isn't met.

When examining the fidelity of synthetic data, the MWU-test is used to gauge the univariate similarity between the real and the synthetic data. This is achieved by investigating if there's a significant discrepancy in the distributions of each feature.

The null hypothesis in the MWU-test asserts that the two samples have identical distributions. When the resulting p-value is less than 0.05, it is indicative of rejecting the null hypothesis, implying a noteworthy distinction between the distributions of the actual and synthetic data for a particular characteristic. Conversely, if the p-value is greater than or equal to 0.05, it suggests that the null hypothesis cannot be rejected, implying that there is no significant difference between the distributions of the real and synthetic data.

The MWU-test works by ranking the combined data from both samples and calculating the sum of the ranks for each sample. The test statistic, U, is then computed based on these rank sums, and its significance is assessed using either a lookup table or by approximating a normal distribution for larger sample sizes.

By applying the MWU-test to each feature in the real and synthetic datasets, researchers can evaluate the univariate resemblance and determine whether the generative model has successfully captured the overall distribution of each feature. This analysis helps in understanding if the synthetic data maintains the distributional properties of the real data, which is an important aspect of the underlying patterns and relationships in the real data.

Multivariate Relationships

Multivariate relationships refer to the associations and dependencies between multiple features in a dataset. Regarding synthetic data evaluation, assessing multivariate relationships is crucial for determining whether the generative model has effectively captured the underlying structure and interactions between features present in the real data. A comprehensive evaluation of multivariate relationships helps ensure that the synthetic data maintains the real data's complex patterns.

There are several methods to assess and compare multivariate relationships in the real and synthetic datasets [81]. One common approach is to compute the pairwise correlations between features, which provides insight into the linear relationships among them. Pearson correlation is a widely used measure to calculate these pairwise correlations. Additionally, comparing correlational similarities using a metric like cosine similarity can help quantify the resemblance between the real and synthetic datasets. Visualization techniques, such as correlation plots, can also be employed to support the assessment of multivariate relationships.

Pearson Pairwise Correlations Evaluating the fidelity of synthetic data requires assessing the multivariate relationships among features. One effective method to compare the numerical features of real and synthetic datasets is by calculating the Pearson pairwise correlations between the corresponding features in both datasets [81]. This allows researchers to determine how well the generative model has captured the linear relationships among those features.

By comparing the Pearson correlation coefficient matrices for the real and synthetic datasets, researchers can identify similarities and differences in the multivariate relationships between features. High similarity in the Pearson pairwise correlation matrices indicates that the generative model has effectively maintained the linear relationships among features, to ensure that the synthetic data captures the real data's underlying structure and interactions.

Additionally, the Pearson correlation coefficient can be used to analyze the linear relationship between a target variable and each feature in a dataset. Researchers can evaluate the fidelity of synthetic data by comparing the Pearson correlations between the target and features for both the real and synthetic datasets. This allows them to assess the extent to which the generative model has preserved the linear relationships between the target and the features.

A strong resemblance in the Pearson correlation coefficients between the target and features for both datasets indicates successful preservation of linear relationships by the generative model. This aspect is vital to fidelity, as it guarantees that the synthetic data upholds the fundamental patterns and connections pertinent to the target variable, which is of great importance for areas like predictive modeling and ML.

Dimensional Resemblance Analysis

Dimensional resemblance analysis employs dimensionality reduction techniques to facilitate the comparison of real and synthetic datasets in lower-dimensional spaces. Typical techniques for this purpose are PCA and Isometric Mapping (Isomap). Before we delve further into the details of dimensional resemblance analysis, a definition of Isomap is in order.

While PCA is a linear transformation, Isomap is a non-linear dimensionality reduction technique that aims to preserve the intrinsic geometric structure of high-dimensional data by approximating the geodesic distances between data points in a lower-dimensional space[86]. Isomap is based on manifold learning and is particularly useful for analyzing data with non-linear structures. The algorithm for Isomap involves three primary steps:

1. Construct a neighborhood graph: For each data point, connect it to its nearest neighbors, typically using the k-nearest neighbors or epsilon-radius approach. This graph captures the local structure of the data.

2. Compute shortest path distances: Calculate the shortest path distances (also known as geodesic distances) between every pair of data points in the graph, usually using Dijkstra's or Floyd-Warshall's algorithm. Geodesic distances represent the shortest paths between data points on the manifold, approximated by the graph distance in the neighborhood graph.

3. Embed data points in lower-dimensional space: Apply classical multidimensional scaling to the matrix of shortest path distances to obtain a lower-dimensional embedding that preserves the pairwise geodesic distances between data points as closely as possible.

In the realm of synthetic data evaluation, dimensionality reduction techniques like Isomap and PCA play a crucial role by enabling the comparison of real and synthetic datasets within

lower-dimensional spaces. Isomap, in particular, shines in contexts where the data's underlying manifold structure is non-linear, adeptly capturing the complex relationships between data points that linear techniques like PCA might overlook. By projecting the synthetic data onto the top two lower-dimensional components extracted from the real data using these techniques, we can directly inspect the overlap and consistency between the datasets, particularly focusing on their non-linear relationships.

Subsequently, this approach allows us to assess the synthetic data's ability to preserve the underlying structure, patterns, and relationships inherent in the real data. Visualization tools such as score plots and scree plots further aid in this comparative analysis. Score plots visualize the data in its top two lower-dimensional components, providing insights into shape similarities and variabilities, while scree plots graphically display the variance explained by each principal component, illustrating the proportion each contributes to the overall data variation.

Moreover, the choice of projection technique can provide differentiated insights into the synthetic data's capacity to capture diverse relationships. For instance, while projecting onto the real data's principal components allows us to assess feature relevance and interpretability in the synthetic data, using Isomap lower-dimensional embeddings helps evaluate its ability to encapsulate complex, non-linear relationships and manifold structures inherent in the real data.

Data Labeling Analysis

Data Labeling Analysis serves as a method to evaluate the semantic resemblance between real and synthetic data. This analysis involves assessing the performance of ML classifiers in their ability to distinguish between real and synthetic records [81]. To conduct this analysis, the real and synthetic datasets are first combined and labeled accordingly.

Subsequently, the combined dataset is split into a train and test set, with the specific ratio determined by the researchers. The data is then pre-processed, which involves the standard practices of normalizing numerical features and one-hot encoding categorical features. The classifiers are then trained with train data, and their ability to distinguish between real and synthetic data in the test set is observed.

The performance of the classifiers is inversely related to the fidelity of the synthetic data. A high classification accuracy suggests that the classifier can easily distinguish between real and synthetic records, suggesting a low fidelity of the synthetic data. Conversely, a low classification accuracy signifies that the classifier struggles to differentiate between real and synthetic records, implying a high fidelity of the synthetic data. This analysis provides insights into how well the generative model has captured the underlying structure and semantics of the real data, which is crucial for ensuring the synthetic data's quality and usefulness in various applications.

Recent research [87] has shown that strong classifiers, such as XGBoost, are able to easily distinguish between state-of-the-art synthetic data and real data on several tabular datasets. In reality, they exhibit near-perfect differentiation capabilities, indicating that generating realistic synthetic NIR spectra by treating it as tabular data can prove to be a formidable challenge. In-depth analyses of the important features of these classifiers have highlighted that mixed-type and ill-distributed numerical columns, which are often present in NIR spectroscopy data, are the least faithfully reconstituted. This indicates that these types of features are more challenging for generative models to capture accurately, which can negatively impact the fidelity of the synthetic spectra. The ability of strong classifiers to easily distinguish between real and synthetic NIR spectra reinforces the importance of evaluating synthetic data using data labeling analysis,

which can help assess the quality of the synthetic spectra and identify areas for improvement in generative models.

2.4.2 Utility

The concept of utility in the context of synthetic data fundamentally hinges on how effective this data is for training ML models. Two key methodologies come into play here: Train on Synthetic, Test on Real (TSTR), Train on Real, Test on Real (TRTR), which is suggested in the benchmarking framework of Mikel Hernandez et al. [81]. In the TSTR approach, models are trained on synthetic data and then evaluated on real data. Meanwhile, the TRTR method, where models are both trained and tested on real data, provides a benchmark for comparison. This benchmark serves as a standard of excellence that the TSTR performance should ideally meet or surpass. Hence, the utility of synthetic data is essentially evaluated by how well it helps in training ML models, as gauged against the established TRTR benchmark.[80].

Evaluation Metrics

To evaluate the utility of the data, TSTR and TRTR methodologies are employed. The performance assessment within these frameworks can utilize a variety of evaluation metrics. The selection of an appropriate metric is contingent upon the type of problem, and whether it is a classification or regression task.
In classification tasks, commonly used evaluation metrics include:

- **Accuracy:** The proportion of correctly classified instances out of the total instances.
- **Precision:** The proportion of true positive instances among instances predicted as positive.
- **Recall:** The proportion of true positive instances among actual positive instances.
- **F1-score:** The harmonic mean of precision and recall, providing a single score that balances the trade-off between precision and recall.
- **Receiver Operating Characteristic (ROC) Curve:** A graphical representation of a classifier's true positive rate against its false positive rate, quantifying the overall performance of the classifier across different decision thresholds.

In regression tasks, commonly used evaluation metrics are:

- **Coefficient of Determination (R^2):** A measure of how well the predicted values match the actual values is the coefficient of determination (R^2). R^2 values range from 0 to 1, where a value of 1 indicates a perfect fit, and a value of 0 implies no relationship between the predicted and actual values. It is worth noting that negative R^2 values can occur, which indicates that the model's performance is worse than a baseline model.

  A baseline model, in this context, is a simple model used as a reference point to compare the performance of more complex models. It typically predicts a constant value, such as the mean or median of the target variable, for all instances. If a more sophisticated model achieves negative R^2 values, it suggests that the model does not capture the underlying relationships in the data and is outperformed by the simplistic baseline model.

- **Mean Absolute Error (MAE):** The average of the absolute differences between predicted and actual values, providing an easy-to-interpret measure of the prediction error.

- **Root Mean Squared Error (RMSE):** The square root of the average of the squared differences between predicted and actual values, emphasizing the impact of larger errors on the overall error measurement.

TSTR and TRTR

TSTR and TRTR is an established benchmark for utility in the literature [80, 81, 88, 89]. This approach involves training a ML algorithm on both real and synthetic data, and then comparing the performance of the resulting models using a holdout dataset derived from the real data.

The TRTR method establishes a benchmark for optimal performance by training and testing the model on real data. On the other hand, the TSTR method trains the model on synthetic data and tests it on real data. Assessing the utility of the synthetic data is achieved by comparing the performance of the TSTR model against the established TRTR benchmark.

The evaluation of utility is pivotal in ascertaining the efficacy of synthetic data for diverse ML tasks. Performance comparisons between models trained on synthetic and real data, using fitting evaluation metrics for both classification and regression problems, enable researchers to measure the feasibility of synthetic data as a substitute for real data in model training, and whether it affects performance negatively. Guaranteeing high utility is vital for the effective use of synthetic data in practical situations, enhancing confidence in the synthetic data generation process and promoting its widespread use across various sectors and fields.

Chapter 3

Data Exploration

Data exploration is a crucial step in the ML process, as it allows researchers to gain insights into the characteristics of the dataset, identify any data quality issues, and understand the underlying data distribution. This section introduces the dataset used in this thesis and presents an overview of the data collection process, the spectral features, and the sample selection.

3.1 Dataset Description

The dataset used in this study was obtained from a comprehensive analysis of mango samples [11]. The spectra were collected using a F750 Produce Quality Meter (Felix Instruments, Camas, USA) equipped with a MMS1 (Monolithic Miniature Spectrometer; Zeiss, Oberkochen, Germany) that employed an interactance optical geometry to collect apparent absorption spectra in the wavelength region of 300-1100 nm. The instrument has a pixel resolution of approximately 3.3 nm, an optical resolution of about 10 nm, and a repeatability of around 1 mAbs units.

The dataset consists of mango samples from eight different cultivars and three National Mango Breeding Program (NMBP) lines, collected over four seasons between 2015 and 2019. A total of 112 unique populations, 4685 mango samples with reference values, and 11,834 spectra were obtained from two distinct growing regions in Australia (Northern Territory and Central Queensland). The fruit samples were scanned at physiological stages ranging from 'hard green' to 'ripening', covering early softening stages through to 'eating ripe'. The authors of the study where the data originates from removed 143 spectra as outliers. That left the dataset with 11,691 samples.

Spectra collection and destructive reference analysis were performed within a single day. Each fruit was scanned twice on the widest section of each cheek (approximately the middle of the fruit) orthogonal to the endocarp plane. A subset of 744 samples were scanned at three different fruit temperatures (approximately 15, 25, and 35 °C). Fruit cheeks were sliced, and a cylindrical core (29 mm diameter) was taken at the location of spectral acquisition and trimmed to 10 mm length (from skin side). The plug was quartered, weighed, and then dehydrated for further analysis.

3.2 Data Exploration

The data was thoroughly investigated in this subsection using various visualization techniques and statistical measures to gain a deeper understanding of its characteristics and relationships between features.

3.2.1 Preliminary Investigation

Figure 3.1 contains all the spectroscopic measurements from the dataset. As can be seen, most of the samples have zero measurements for the spectral wavelengths in the beginning and in the end of the dataset.

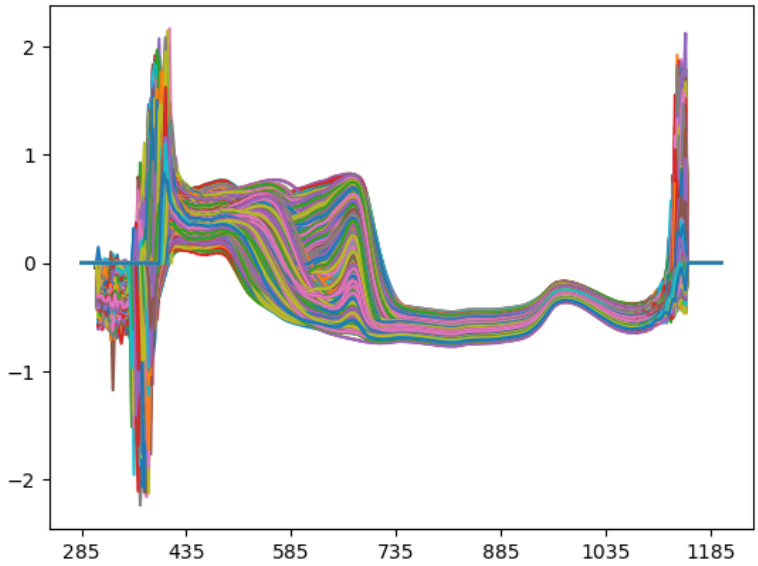


Figure 3.1: Plot of all spectra in mango dataset

Figure 3.2 displays the count of samples with zero measurement for each wavelength. It can be seen that on the very edge of the spectral band provided all samples are measured at zero, and for the nearby wavelengths a substantial number of the samples contain zeros. Totally, there is 403 spectral wavelengths that contains only zeros. Additionally, there is a lot of noise at the beginning and the end of the spectra.

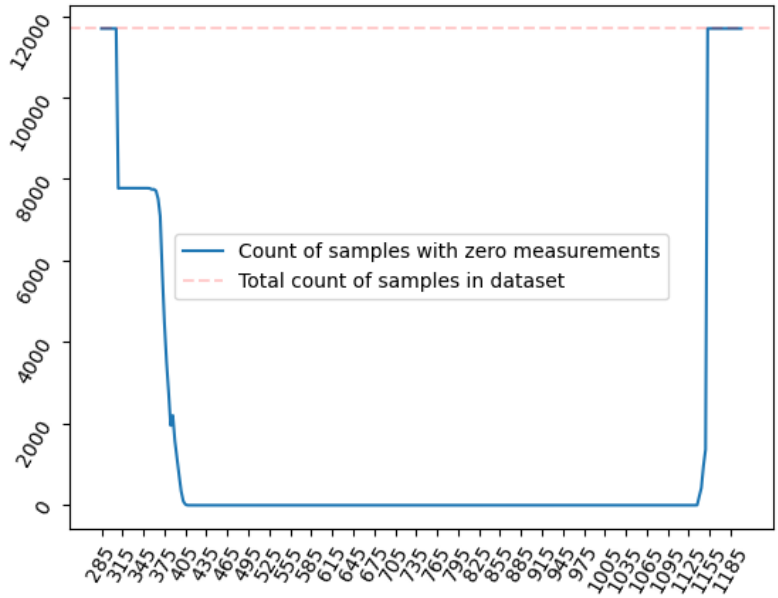


Figure 3.2: Count of samples with zeros for each wavelength

As this master thesis is primarily concerned with the NIR spectrum, and we define the NIR spectrum as 700-2,500 nm, a plot of the spectral measurements in the range of 699-1200 nm is shown in Figure 3.3.

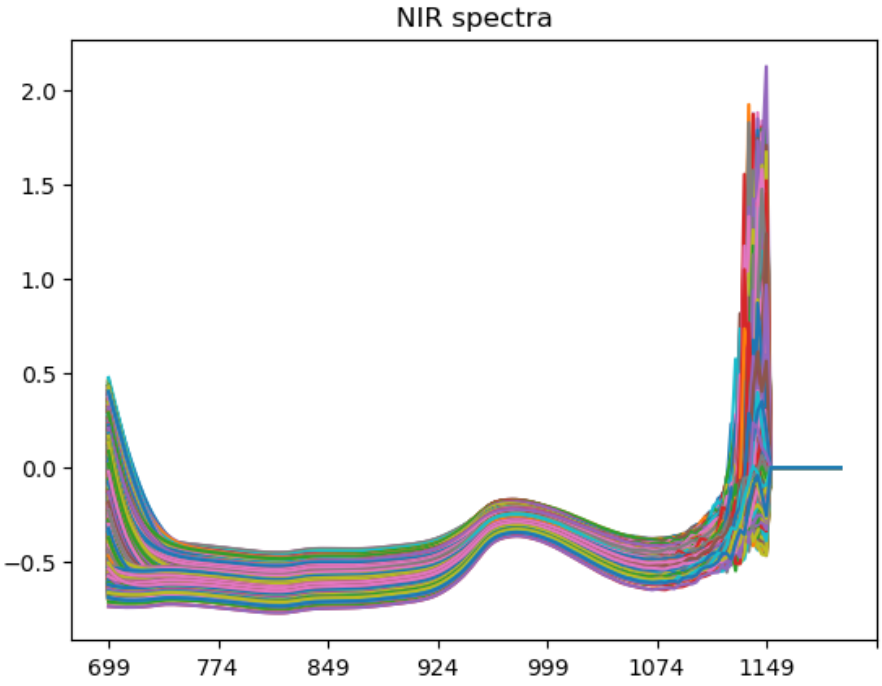


Figure 3.3: Plot of NIR spectra of samples. We can see that the end range of the NIR spectra contains a lot of zeros

We see that the end range of the NIR spectra first contain a lot of apparent noise and subsequently a lot of zero measurements toward the very end. Figure 3.4 zooms into where the

spectral measurements start to get really noisy. The noise seem to apparently increase drastically after wavelengths of 1030 nm and beyond. When modelling the spectra, this could serve as an interesting cutoff point to make the job easier for the generative models.

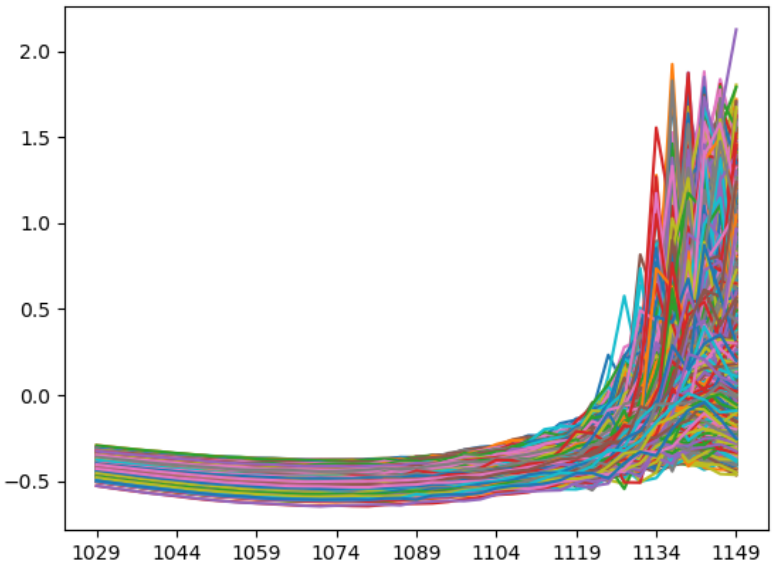


Figure 3.4: Plot that zooms in on the point where spectra start to get noisy

Figure 3.5 zooms into the elbow point for when the zero measurements start to increase drastically for wavelengths in the NIR spectra. The elbow point occur at wavelength 1137 nm, and can also potentially serve as a cutoff point when trying to model the spectra.

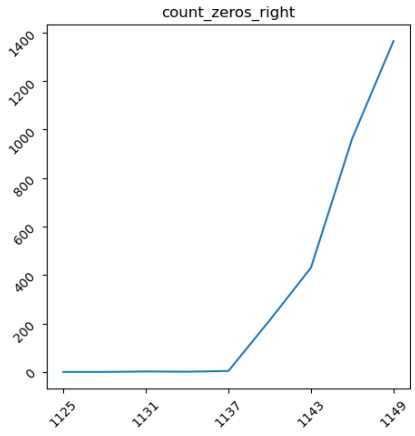


Figure 3.5: Elbow point of zero count at end of spectra

The NIR spectra contains no missing values, though the columns that contain only zeros could be considered as missing values, though they are not encoded as so. Figure 3.6, which is plotted with open-source library missingno, shows that there are no missing values.

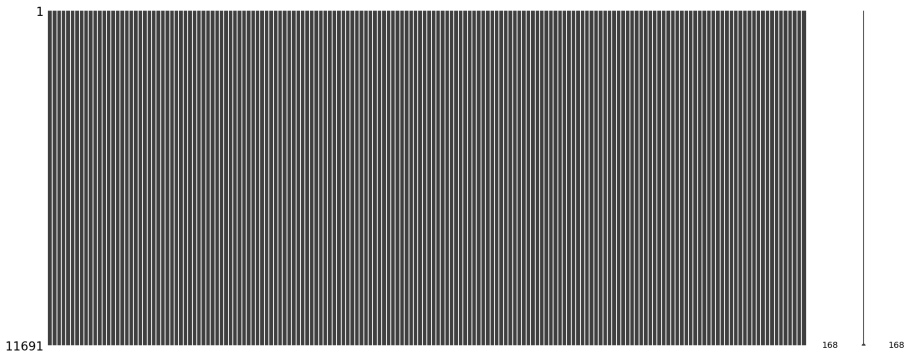


Figure 3.6: Missingno plot shows that we have no missing values

Figure 3.7 shows a subset of NIR spectra that does not contain excessive noise. Interestingly, Figure 3.7 reveals two distinct subgroups of spectra within the real data: towards the left end of the spectra, one group exhibits an upward curve, while the other maintains a flat trend.

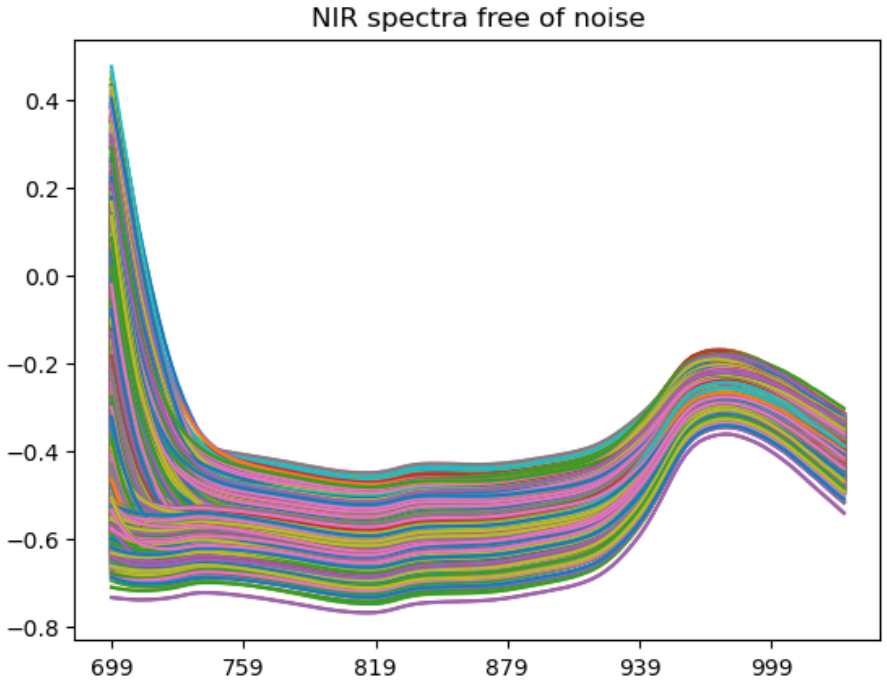


Figure 3.7: NIR spectra free of noise

Figure 3.8 provides a closer look at the point where the two subgroups diverge from one another, occurring at approximately 744 nm wavelength.

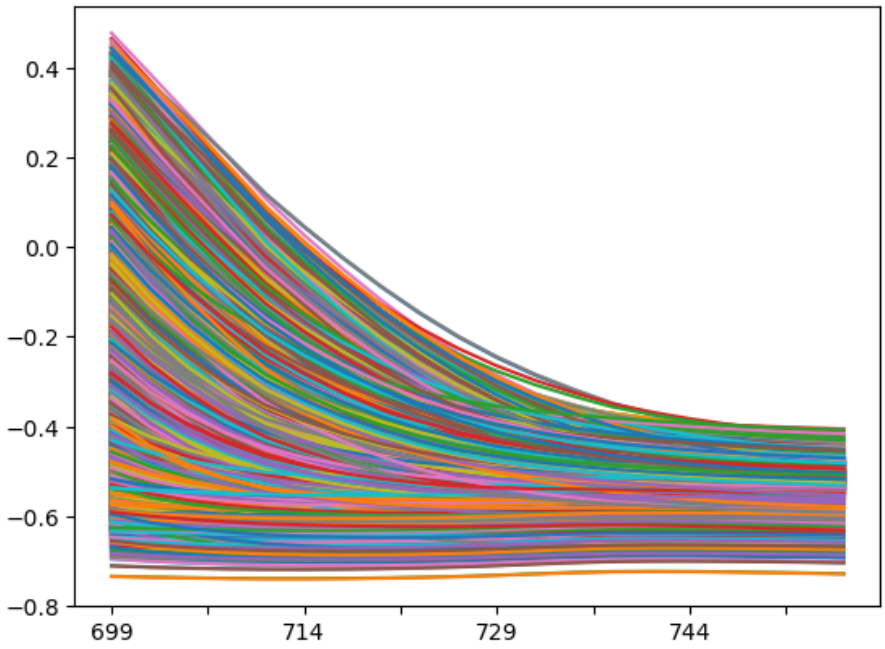


Figure 3.8: Point of divergence between two subgroups of NIR spectra, around 744 nm

3.2.2 Statistical Properties

The dataset's basic statistical descriptions, as shown in Figure 3.9, reveal a curved pattern across various measures for the NIR spectra. The mean values start at -0.123162 and dip to a minimum of -0.549416 around the index of 738 before rising back towards zero, suggesting that the average absorbance values in the NIR spectra are below zero.

The standard deviation values initially decrease from 0.235237 to a minimum of 0.036566 at index 780, then rise slightly, indicating that the dispersion of absorbance values in the NIR spectra narrows before widening again. The minimum values exhibit a similar curved pattern, starting at -0.734356, reaching a minimum of -0.745980 around index 837, and then ascending. This demonstrates that the lowest absorbance values in each dataset are below zero.

The 25th, 50th (median), and 75th percentile values also follow a curved pattern similar to the mean and minimum values. The lower quartile, middle value, and upper quartile of the absorbance values in the NIR spectra all reach a minimum and then increase, emphasizing the curved nature of the spectra data. In essence, the provided statistics indicate that the NIR spectra follow a curved pattern with changing dispersion, as the absorbance values become more tightly clustered before spreading out slightly again.

The skewness of the NIR spectra, a measure of the asymmetry in the distribution of the data, also presents an interesting pattern. The values commence at -0.339706, gradually increase to a peak of 0.287913 around index 7, and then gradually decrease to a minimum of -0.965078. This fluctuation in skewness suggests a shift from a left-skewed distribution (where the left tail is longer or the mass of the distribution is concentrated to the right), through a near-normal distribution, to a right-skewed distribution (where the right tail is longer or the mass of the distribution is concentrated to the left). This skewness pattern, combined with the aforementioned statistical measures, provides a comprehensive picture of the complex, non-linear patterns in the NIR spectra data.

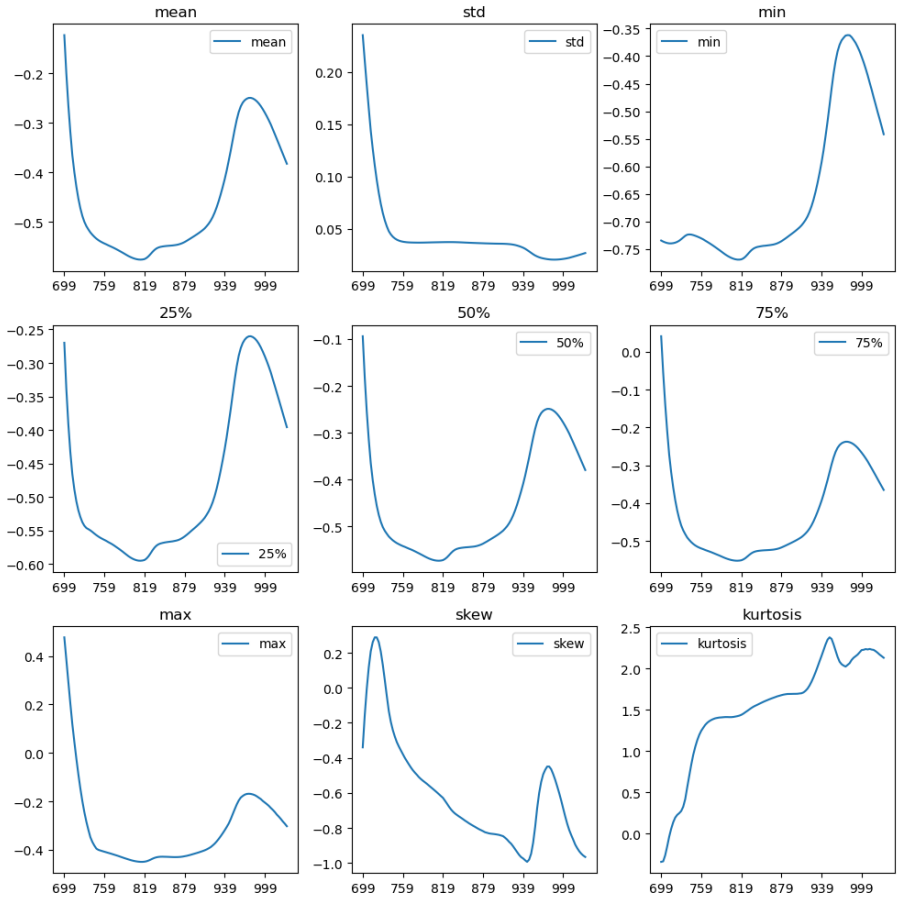


Figure 3.9: Plot of basic statistical descriptions of the dataset

Chapter 4

Method

The code for used in this project can be found here.

Spectroscopic data is typically represented in a tabular format. Our hypothesis is that generative ML models specifically designed for tabular data would be well-suited for generating artificial spectra. SDV, an open-source library established at MIT's Data to AI Lab in 2016, serves as a comprehensive ecosystem for synthetic data generation [3]. Two models within the SDV library, TVAE and CTGAN, have been explicitly designed for handling tabular data. Both models were introduced by Lei Xu et al. in the same publication [90].

Synthetic Data Vault is an open-source library that conveniently lets researchers generate synthetic tabular data utilizing state-of-the-art synthesizers such as CTGAN and TVAE. This thesis aims to investigate if this library is suitable for generating synthetic NIR spectra that correspond well with real NIR spectra.

4.1 Preprocessing

We ultimately chose to restrict the dataset's features between wavelength 699 and 1032, which encapsulate the NIR spectra free from noise, and we chose 'DM' (DMC) as response variable. This leaves the final rendition of the dataset with 113 columns. The definition of the range of NIR spectra vary, but we chose to define it from 700 nm to 2,500 nm. This choice was inspired by a study by Puneet Mishra and Dario Passos [91], who used the same dataset. DMC is an important measure of fruit quality as it is directly linked to various quality attributes such as taste, flavor, texture, nutritional value, shelf life, and processing suitability [92, 93]. Several studies have shown that DMC serves as a strong indicator of fruit quality [92, 93] and can be reliably assessed using NIR spectroscopy [94].

Consequently, using DMC as the response variable in this study is well-founded, and being able to generate synthetic data that retains the relationship between NIR spectra and DMC would be highly valuable. In the dataset, spectral measurements originating from the same sample were arranged sequentially, resulting in consecutive rows representing data from identical samples. To ensure that the generative models did not emphasize any arbitrary patterns resulting from consecutive spectral measurements of the same sample, the order of the rows was randomized, thereby minimizing potential biases when generating new data.

Normalization is a common preprocessing step when working with neural networks, as it helps to standardize the input data and ensure that the model can learn effectively from the data.

This often involves scaling the features to have a mean of zero and a standard deviation of one or transforming them into a specific range. However, when using SDV's generative models, such as CTGAN and TVAE, this preprocessing step is not required. These models are designed to automatically apply mode-specific normalization techniques during the data generation process, allowing them to handle diverse data types and distributions effectively. As a result, users can directly work with the generated data without the need for additional normalization steps, simplifying the data preparation process.

4.2 Training

Both CTGAN and TVAE was trained with batch sizes of 50 and for 500 epochs. CTGAN was also trained with 'verbose=True', which prints out the loss values for both generator and discriminator. Other than that, they were trained with default hyperparameters. For CTGAN, that means a learning rate of 2e-4 for both generator and discriminator, and a weight decay of 1e-6 for the adam optimizer for both the generator and discriminator. For TVAE the regularization term l2scale defaults to 1e-5. Both CTGAN and TVAE have a hyperparameter `enforce_min_max_values`, which makes the synthetic data adhere to the min/max boundaries set by the training data.

During the training process, we performed manual experimentation with several hyperparameters to identify the best configuration for our models. However, we found that the default hyperparameters provided the most stable results across our experiments. Despite testing various combinations of learning rates, batch sizes, and other hyperparameters, we did not observe significant improvements in the training stability, especially when adding dry matter as a response variable. Consequently, we decided to proceed with the default settings for both CTGAN and TVAE.

The models were trained on Google Colab, which provides a convenient way to access free GPUs for training neural networks on the data. Unfortunately, to the best of our knowledge, SDV has not made it possible to plot loss curves from CTGAN training by default. This leaves us with the option to inspect the training process by printing the loss values for each epoch. Since directly sharing these printed values here is impractical, we copied the values and used customized code to extract the loss values for plotting. The loss curves are displayed in Figure 4.1.

Figure 4.1: The Loss curves of CTGAN were quite unstable during training. While reducing learning rate seemed to improve stability, it did not produce any better results.

We note that the training process was quite unstable, with the loss oscillating around 0 for both generator and discriminator. Attempts at modifying the learning rates were made, but to little avail. The model was plagued by unstable training, particularly when adding dry matter as a response variable.

4.3 Savitzky-Golay Smoothing

Upon visualizing the synthetic data, it became evident that the synthetic data, particularly the data generated using CTGAN, was considerably less smooth than the real data, as can be seen in Figure 4.2. To address this issue, we employed the SG filter to smooth the synthetic data, making it more akin to the training data. The decision to use the SG filter for smoothing was inspired by the article from which the mango dataset originates [11], and by the fact that it is a well-established algorithm for this purpose. SG filters are frequently employed to separate signal from noise in NIR spectroscopy [95].

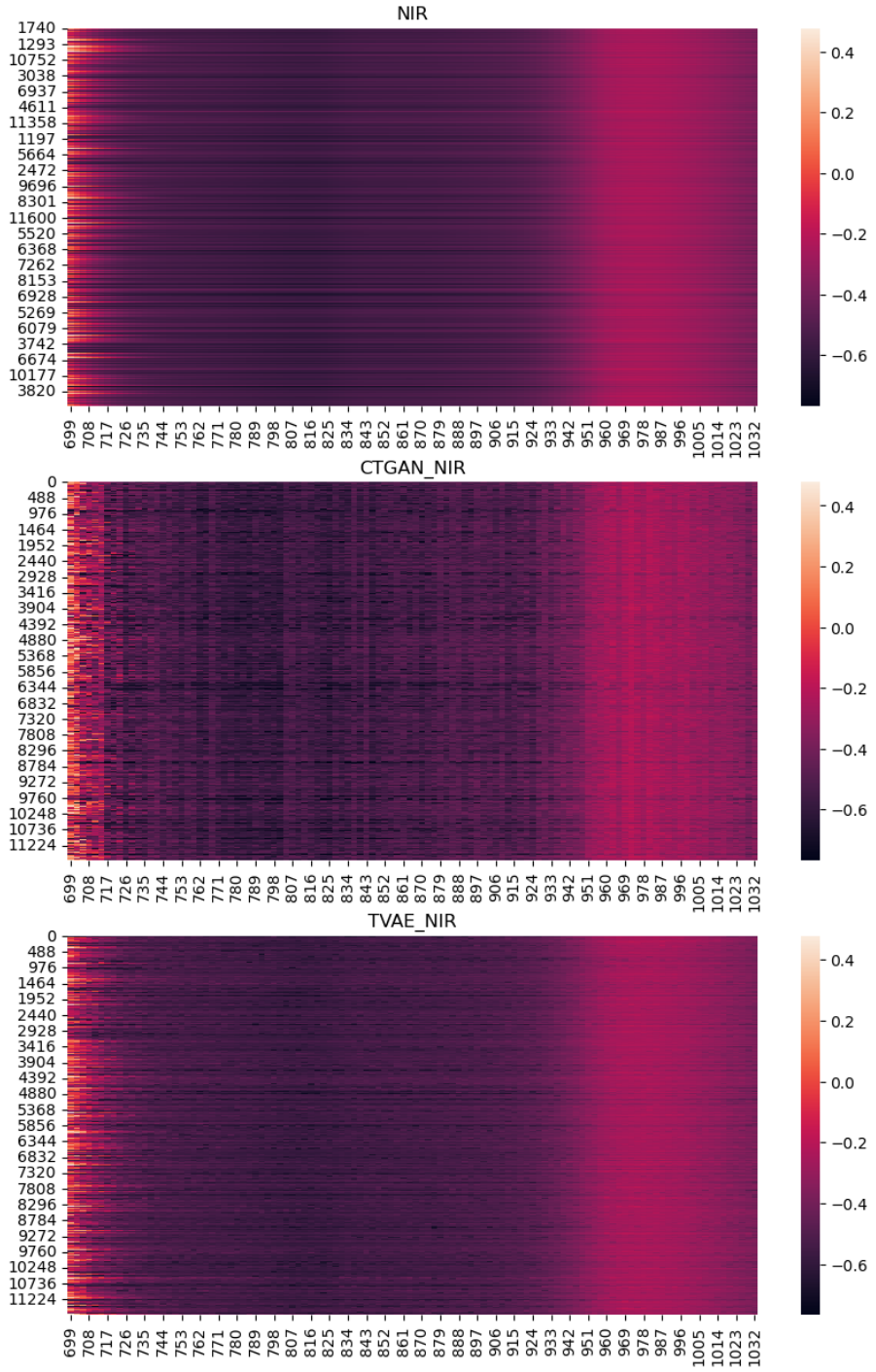


Figure 4.2: Initial heatmaps show that synthetic data is less smooth than real data

To capture the smoothness of the real data, we chose a relatively wide window of 21. This window size was selected based on the observation that a wider window can effectively capture the underlying trends in the data while reducing the impact of noise, resulting in smoother synthetic data that resembles the real dataset. In addition, we chose a polynomial order of 3 to account for the complexities in the data. A higher polynomial order can better approximate the variations present in the spectral signal, and a polynomial order of 3 offers a balance between the simplicity of a linear fit (polynomial order 1) and the complexity of higher-order polynomials. This combination of hyperparameters aims to provide a balance between preserving the essential features of the data and eliminating unwanted noise, ensuring that the synthetic data closely

resembles the real dataset.

4.4 Evaluation

In this thesis, our primary goal is to investigate the ability of tabular generative models from the SDV library to synthesize high-quality spectral data. Typically, the evaluation of synthetic data is based on three key aspects: fidelity, utility, and privacy [81]. However, as NIR spectral data rarely involves privacy concerns, this aspect will not be considered in our assessment. Instead, we focus extensively on examining fidelity and utility, utilizing a range of techniques to provide a comprehensive evaluation of the synthetic data's quality. Thus, our judgment of the data generated by the models will predominantly rest on these two key areas of evaluation.

Our evaluation process drew inspiration from the work by Hernandez et al. [81], but we tailored it to accommodate the specific characteristics of our data. Given that we are working with a high-dimensional dataset consisting solely of numerical features, we found it necessary to modify the presentation of certain metrics or even discard them entirely. Presenting statistics for each column individually in a table is impractical, as our working dataset comprises 113 columns (112 features and 1 target). Additionally, our dataset does not include any categorical features. Moreover, privacy concerns, which were a key dimension for the authors, are not relevant in our case. In order to assess the fidelity of the synthetic spectra, we examined univariate similarity, multivariate relationships, dimensional similarity, and the ability of common ML classifiers to distinguish between the real and synthetic data. To evaluate the utility of the synthetic data, we trained ML regressors on both real and synthetic data and then assessed their performance on a separate, identical holdout set of real data using the TRTR and TSTR methodology. Our approach is illustrated in Figure 4.3, inspired by Mikel Hernandez et al. [81].

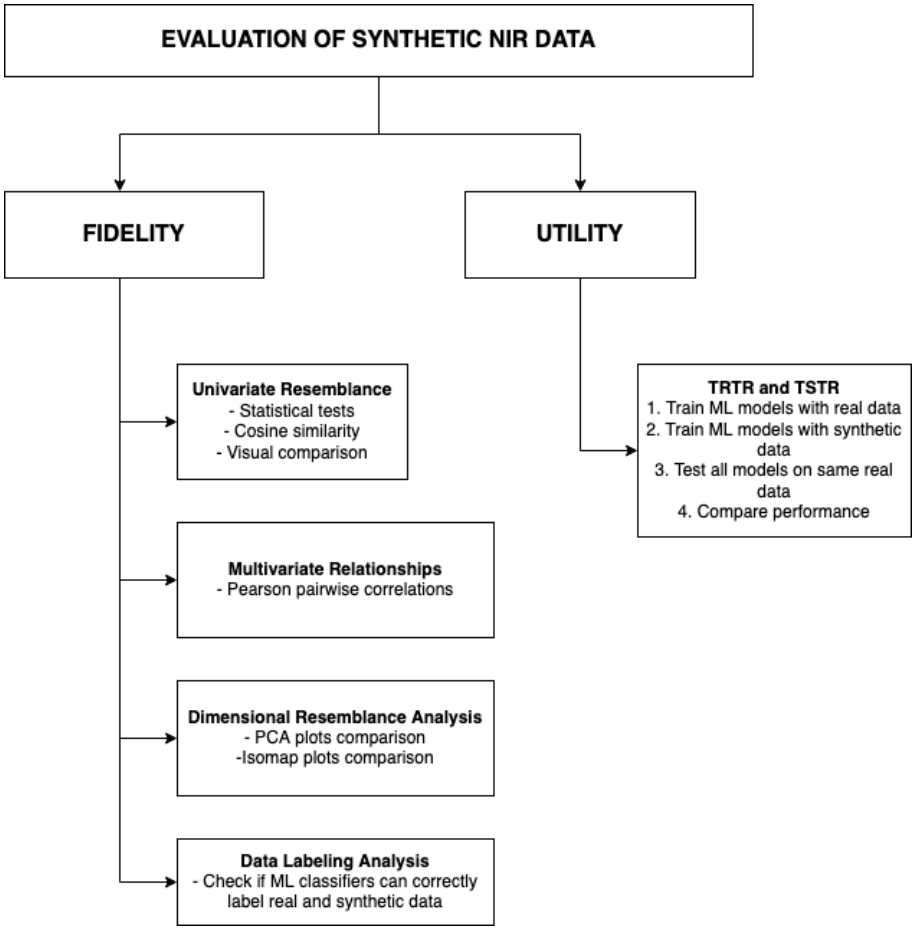


Figure 4.3: Figure that outlines our modified approach to evaluation of NIR data. Inspired by Mikel Hernandez et al. [81]

The dataset we analyzed consists of NIR spectra and the dry matter content as a response variable. During the evaluation of accuracy, we separated the features from the response variables. This makes sense because the features, which represent the spectral wavelength intensity, describe the same phenomenon, while the response variable, dry matter content, represents something very different. Our primary focus was on evaluating the authenticity of the NIR spectra, but we also visualized the Pearson correlation between the spectra and dry matter content, which visualized how faithful this relationship was preserved across synthetic datasets, and used the ability to predict dry matter from the spectra as a basis for assessing the utility of the synthetic data.

Due to the high-dimensional nature of the data, presenting univariate statistics for individual columns is not feasible. Instead, we computed basic univariate statistics for both the real and generated datasets using a custom function that calculates essential aggregate metrics for each column. This function employs built-in *pandas* functions, such as pandas.DataFrame.describe, which calculates the standard deviation, mean, minimum value, 25th percentile, median (50th percentile), 75th percentile, and maximum value for each feature. Additionally, it includes *pandas.DataFrame.skew* to calculate skewness and *pandas.DataFrame.kurtosis* to compute kurtosis. Since the columns are continuous, we determined that mode was not a suitable metric to include, as continuous data can take on an infinite number of values within a given range, making it less likely for any specific value to repeat. This, in turn, means that the mode might not provide meaningful information about the distribution or central tendency of the data.

Manually comparing the feature values for the 112 columns is impractical, so we computed the cosine similarity between the row vectors containing these aggregate measurements for each wavelength from the real and synthetic datasets. This approach offers an objective way to quantify similarity. We performed this calculation for datasets generated by both CTGAN and TVAE, in both their raw and smoothed forms. This analysis provides insight into how effectively the generators preserve the basic numerical properties of the real dataset.

We assessed the univariate resemblance between the real and synthetic datasets by performing three widely recognized statistical tests: the Student's T-test for comparing feature means, the MWU-test for population comparison, and the KS test for comparing distributions. If the null hypothesis is accepted for each test, the properties are considered preserved. A commonly employed threshold for accepting the null hypothesis is a p value above 0.05. Since these tests compare similarities between corresponding columns from both datasets, we computed aggregated statistics, specifically the mean p value and the percentage of columns with a p value above 0.05. These results can be conveniently compared in a tabular format. Additionally, we visualized the test scores for each pair of columns from all tests using a line plot.

In our approach to evaluating generative models, it is crucial to ascertain whether they preserve the multivariate relationships present in the original data. To this end, we calculated the pairwise Pearson correlations for numerical features, as our dataset consists solely of numerical data. This method enabled us to quantify the relationships between the columns by creating vectors for each dataset, containing all unique Pearson correlation pairs. Given that there are 6,216 correlation pairs with 112 spectral wavelengths as features, manual comparison becomes quite challenging.

In light of this, we utilized cosine similarity to compare the Pearson pairwise correlations between the real and synthetic datasets, as well as to determine the similarity of the correlation pairs across datasets. We computed the mean, minimum, and maximum differences for these correlation pairs. Importantly, we also evaluated the preservation of correlation between the response and features by the generative algorithms.

These methods allowed us to assess the linear relationships among the features and between the target and features in the synthetic and real datasets. A cosine similarity value approaching 1 would signify a high degree of resemblance, indicating that the generative model has effectively captured the underlying structure and relationships in the real dataset, an essential aspect for ensuring the synthetic data's utility in various applications.

We further scrutinized the data by examining their dimensional resemblance. To achieve this, we utilized both a linear dimensionality reduction technique, PCA, and a non-linear dimensionality reduction technique, Isomap. It is recommended to scale the data before using PCA or Isomap, so we scaled the data before performing them. We generated scree plots for the principal components that accounted for at least 95% of the variance in the real data, allowing us to compare the number of principal components required to explain the majority of the variance in both datasets. Moreover, we created score plots where data points were plotted against their top two principal components and projected the synthetic data onto the first principal components of the real data. We also both visualized the data on their respective Isomap lower-dimensional embeddings, as well as projecting the synthetic data onto the lower-dimensional embeddings of the real dataset.

The final approach we employed to evaluate fidelity involved data labeling analysis. Initially, we labeled the data as either real or synthetic and then combined both into a single dataset.

This was done separately for CTGAN- and TVAE-generated data, both raw and smoothed. The dataset was then split into a training and testing set, using an 80:20 split ratio. We scaled the data since some classifiers are sensitive to scale. To assess utility, we applied a wide range of classifiers with predefined hyperparameters, as outlined by Mikel Hernandez et al. [81]. The classifiers and their respective hyperparameters are as follows: RF (n_estimators = 100, random_state = 9), KNN (n_neighbors = 10), Decision Tree (DT) (random_state = 9), SVM (C = 100, max_iter = 300, kernel = "linear", probability = True, random_state = 9), and MLP (hidden_layer_sizes = (128,64,32), max_iter = 300, random_state = 9). These algorithms provide diversity, offering a comprehensive understanding of the synthetic data's applicability in various modeling contexts.

Following our assessment of fidelity, we moved forward to evaluate the utility of the synthetic data. We chose to use the established TRTR and TSTR methodology to measure utility, a method recognized for this purpose [81, 88, 96]. To maintain consistency and to test the synthetic data's utility across various model types, we utilized the regression variants of the algorithms used in our data labeling analysis (RF, KNN, DT, SVM, and MLP), with the same hyperparameters where applicable.

By using TRTR as a benchmark for ML model performance and comparing it with TSTR, we were able to gauge whether synthetic data could potentially be used for training ML models. The holdout test set comprised 20% of the real dataset. Again, we scaled the data, as some of the models are sensitive to scale of the features. We calculated the coefficient of determination (R^2) for all datasets and organized them into two tables: one for the real and raw synthetic data generated by CTGAN and TVAE (Table 5.12), and another for the real and smoothed synthetic data (Table 5.13).

Chapter 5

Results

In the Results section of this thesis, we present and describe the findings obtained from employing the SDV library to generate artificial NIR spectral data using the mango dataset as a case study. Our analysis demonstrates the performance and feasibility of the proposed generative ML models in synthesizing NIR spectra with respect to fidelity and utility, which are key synthetic data attributes. Throughout this section, we will elaborate on the quantitative metrics and visual comparisons used to assess the quality of the synthetic data.

5.1 Fidelity

In the field of synthetic data generation, fidelity refers to the degree to which the generated data resembles the real data in terms of structure, patterns, and statistical properties. A high fidelity synthetic dataset should maintain the essential characteristics of the real dataset while still exhibiting variation, ensuring that the synthesized data is useful for its intended purpose. To assess the fidelity of the NIR spectral data generated by CTGAN and TVAE, we will consider both raw and smoothed versions of the synthetic data and employ a combination of quantitative metrics and visual comparisons. These evaluations will involve assessing the similarity between the summary statistics of the real and generated data, examining the preservation of spectral features and the relationships between them. By investigating these aspects, we aim to provide a thorough assessment of the fidelity of the synthetic NIR spectral data generated by the two models and determine their suitability for potential applications.

5.1.1 Visualizing spectra

We used lineplots and heatmaps to visualize the spectra. Lineplots and heatmaps allows us to visualize all samples simulataneously, and lets us easily compare the spectra from all samples with each other. Lineplots also lets us conveniently compare single samples with each other.

Lineplots

In Figure 5.1, we observe that the raw spectra generated by both CTGAN and TVAE successfully replicate the shape of the real data, with TVAE appearing to more faithfully capture the smoothness of the real data. However, it is evident that the synthesized spectra are considerably rougher than the real data. By comparing this to Figure 5.2, we can see that the smoothed data, while still rougher than the real data, exhibit a closer resemblance to the real data's smoothness compared to the raw synthetic data. This observation highlights the differences in fidelity between the two generative models and emphasizes the importance of data

preprocessing, such as smoothing, to achieve more accurate representations of the real data.

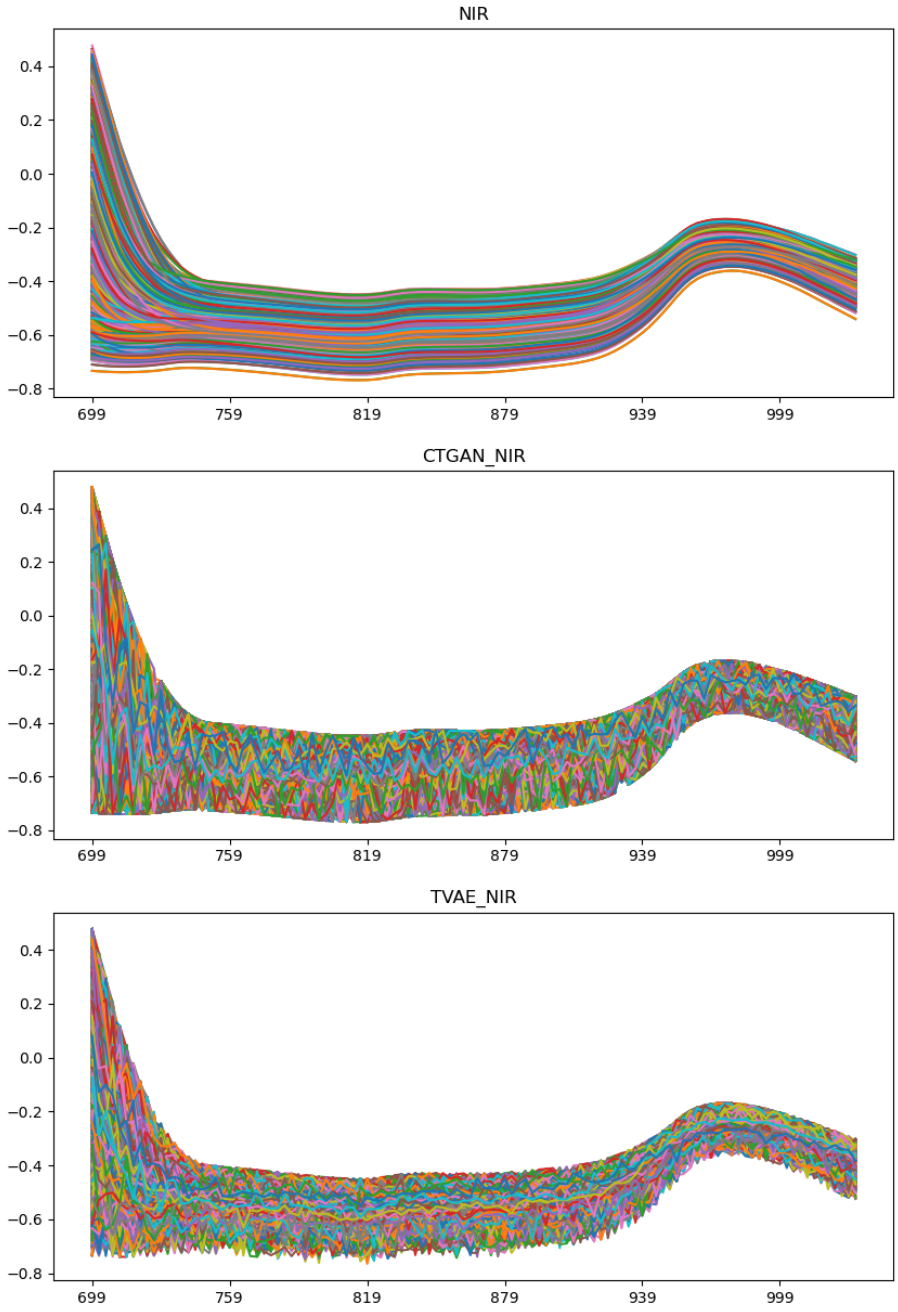


Figure 5.1: Lineplot of all spectra from real and raw synthetic datasets

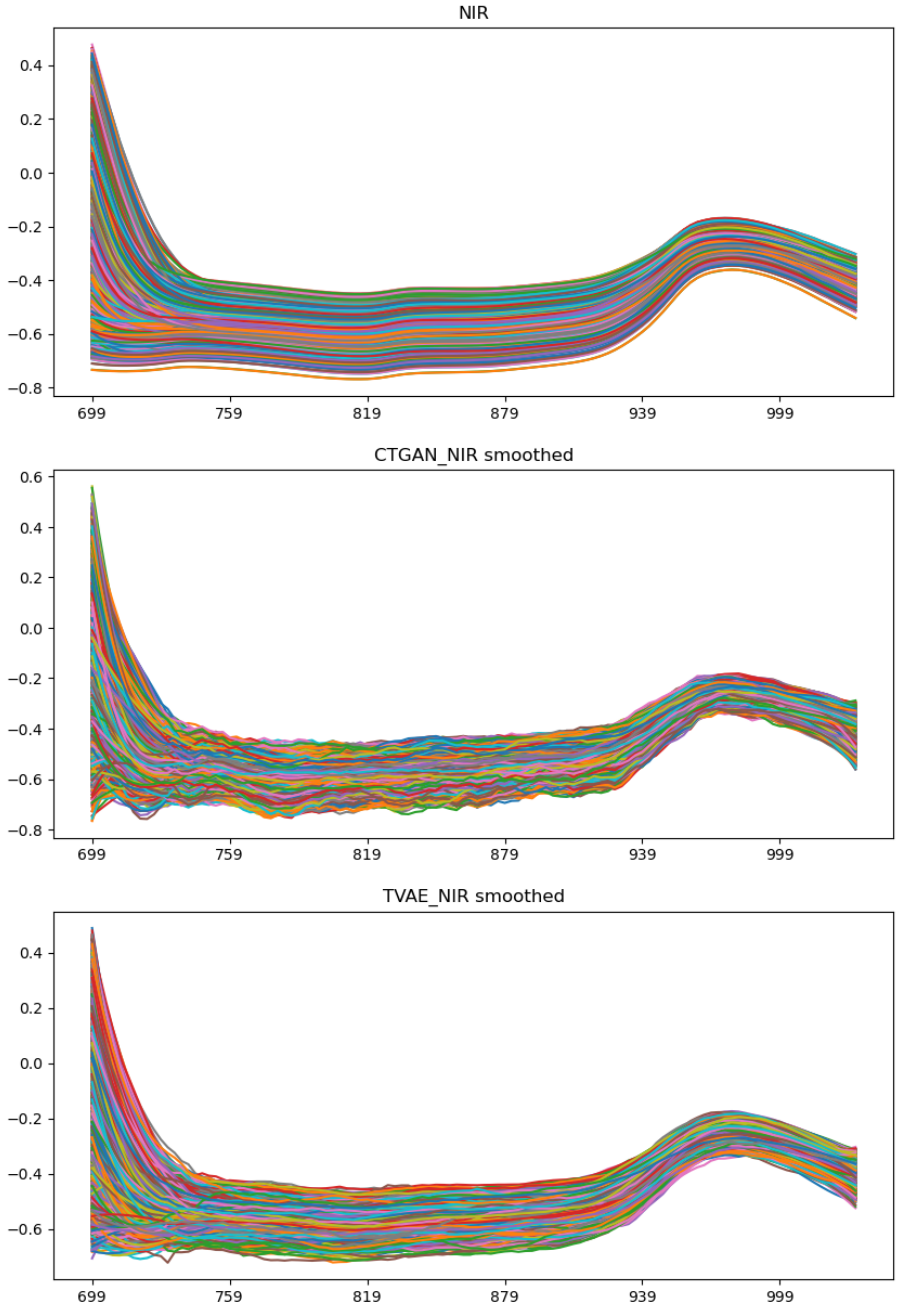


Figure 5.2: Lineplot of all spectra from real and smoothed synthetic datasets

In the real spectra, we can discern two distinct subgroups: one demonstrating an upward trend on the left, and the other exhibiting a flat trajectory. The line plots make it difficult to ascertain whether the generative models have accurately captured these trends, as the synthetic spectra are substantially more dense than the real spectra, concealing this phenomenon. Table 5.1 presents the number of smoothed spectra with a mean value below -0.5 for the initial 15 spectra (encompassing all wavelengths up to 744 nm). While CTGAN fell short in reproducing the quantity within the flat trend subgroup, TVAE came remarkably close. We deliberately computed this quantity solely for the smoothed spectra, given that row means inherently smooth the spectra, rendering a second application redundant.

Table 5.1: Number of smoothed spectra that has a row mean below -0.5 for 15 first wavelengths

	NIR	CTGAN_NIR	TVAE_NIR
count	2375	1230	2291

Heatmaps

In Figure 5.3, the heatmaps corroborate the findings from the line plots, indicating that the synthetic spectra generated by both CTGAN and TVAE capture the overall pattern of the real data, albeit with a rougher texture. CTGAN spectra, in particular, appear to be more jagged than those produced by TVAE. On the other hand, Figure 5.4 demonstrates that the smoothed synthetic spectra bear a closer resemblance to the real data, with smoothness levels approximating those of the real spectra.

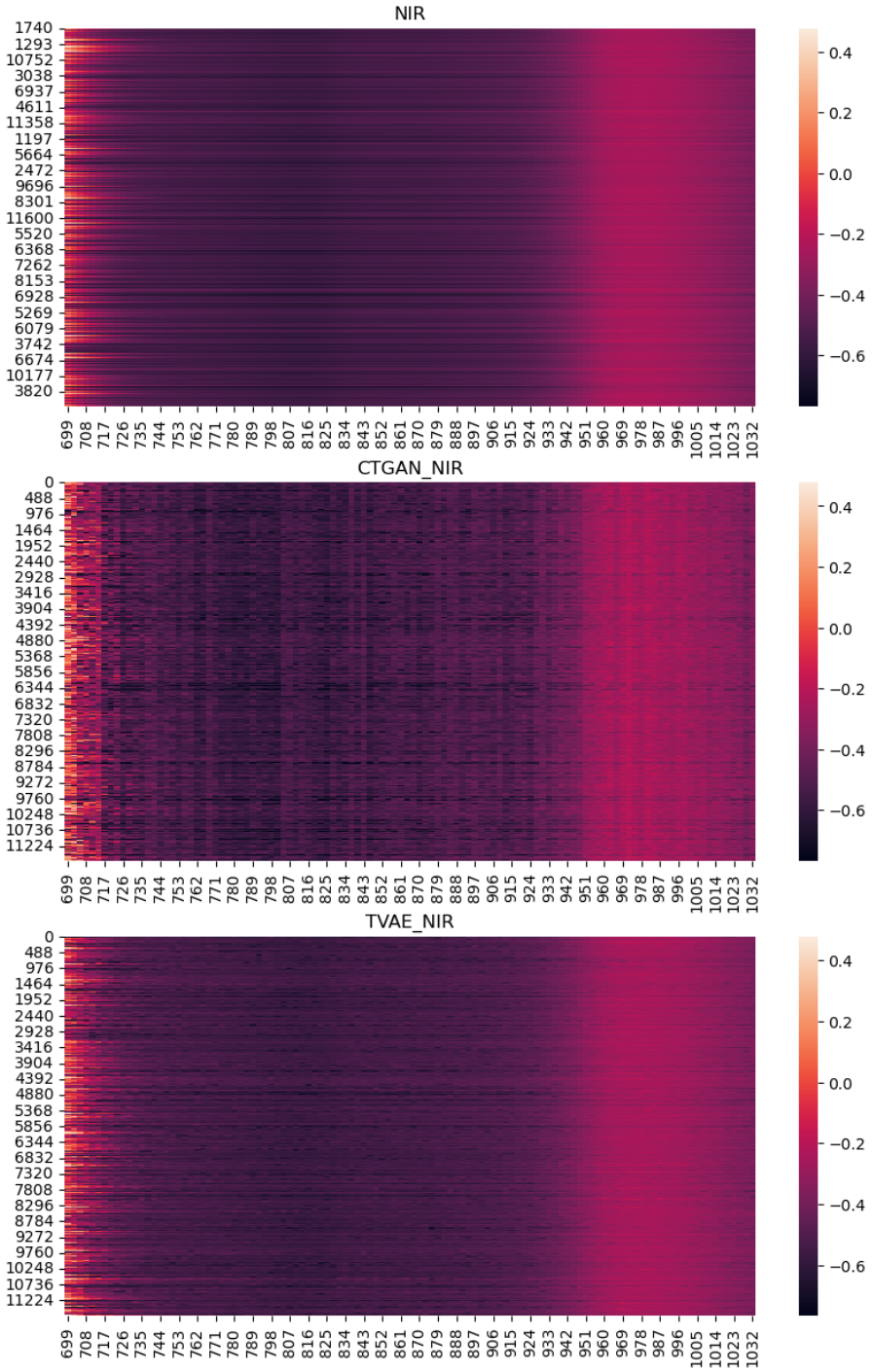


Figure 5.3: Heatmap of spectra from real and raw synthetic datasets

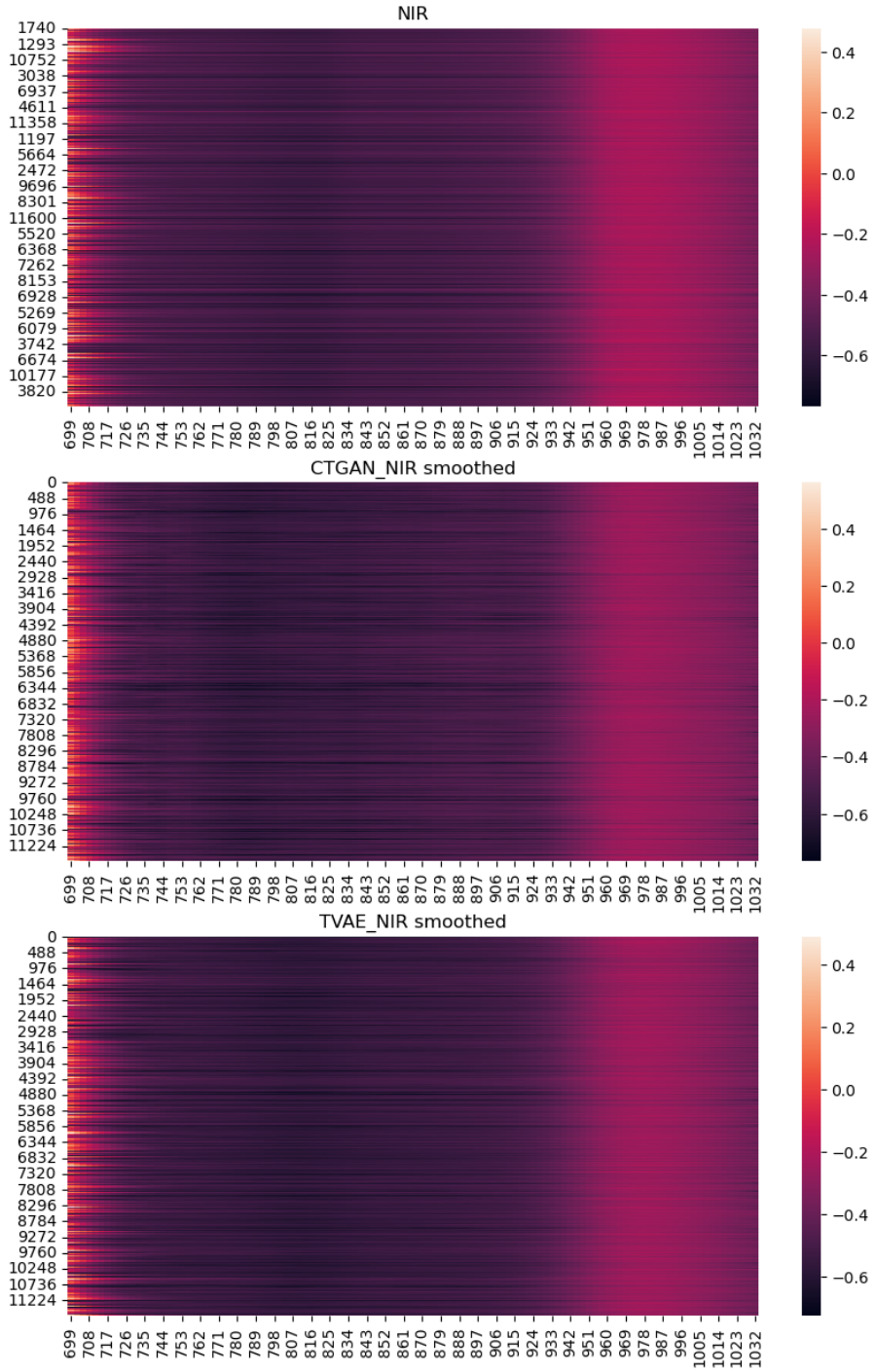


Figure 5.4: Heatmap of spectra from real and smoothed synthetic datasets

Lineplots for Individual Samples

Similarly, Figure 5.5 reveals that the raw data produced by CTGAN is considerably rougher than the real data, despite capturing the overall trend accurately. In contrast, TVAE-generated spectra display a smoothness more akin to the real data when compared to CTGAN. Figure 5.5 also shows that the line plots of the smoothed spectra exhibit greater similarity to the real data than their raw counterparts, emphasizing the importance of smoothing in achieving a more accurate representation of the real data.

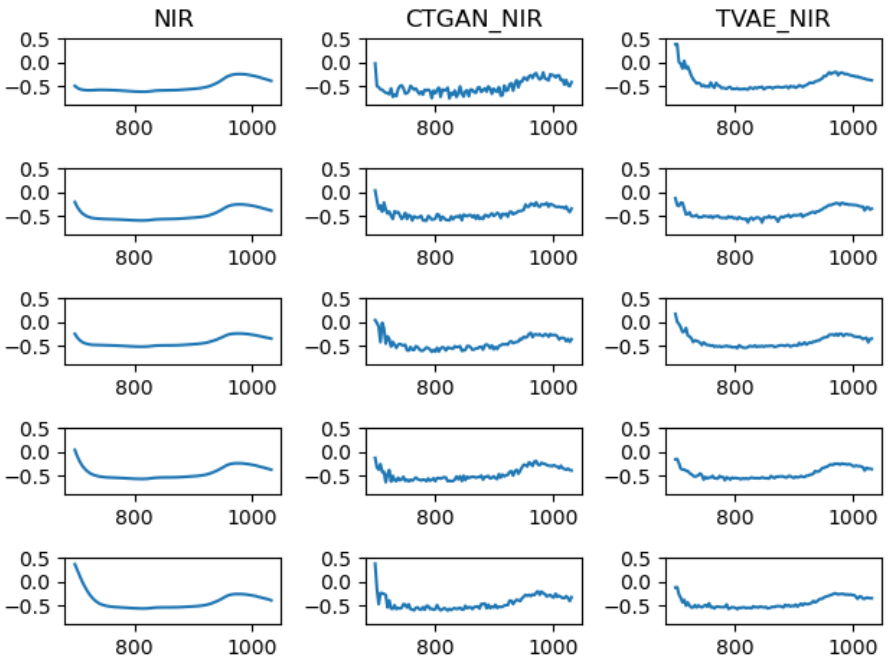


Figure 5.5: Multiple lineplots that show multiple spectra individually from real and smoothed synthetic datasets

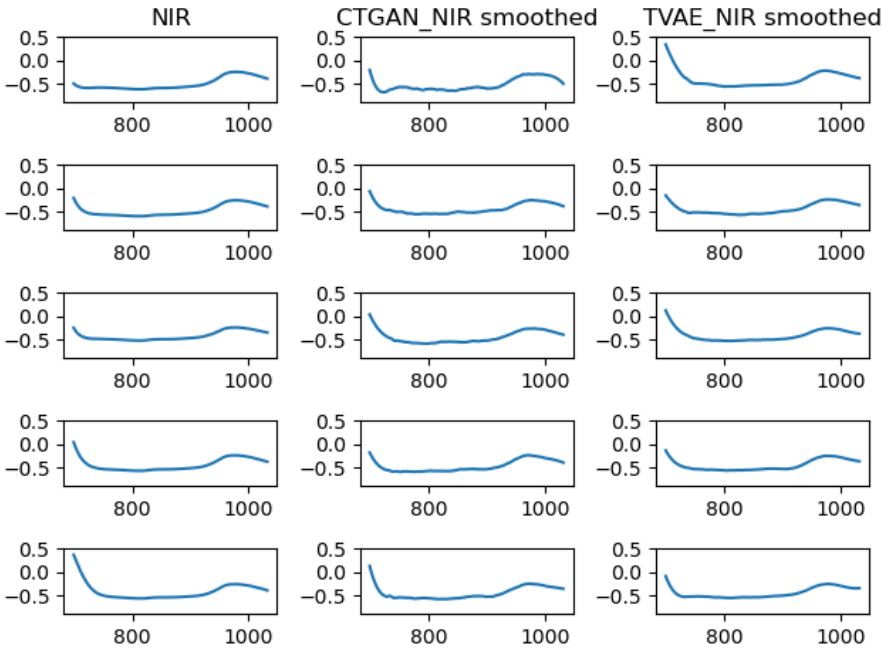


Figure 5.6: Multiple lineplots that show multiple spectra individually from real and smoothed synthetic datasets

5.1.2 Univariate Resemblance Analysis

Statistical Tests

Figure 5.7 illustrates that the majority of columns in the CTGAN-generated spectra have a p value lower than the significance level (<0.05), indicating that these columns are statistically

different from the real data. Interestingly, the statistical tests seem to consistently identify which columns are sufficiently similar and which are not, with the same pattern observed across all figures.

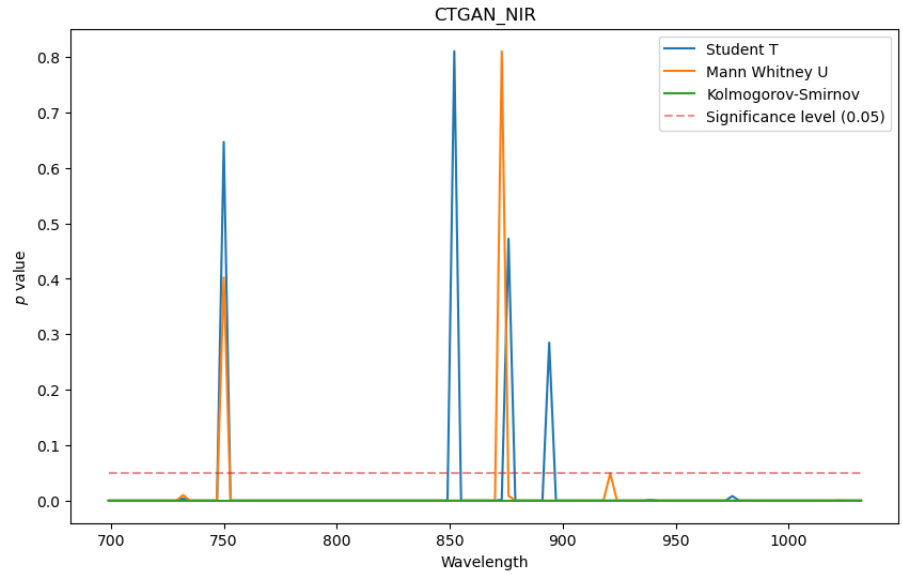


Figure 5.7: Lineplot that shows p values from different univariate statistical tests when comparing real data with raw CTGAN-generated data

In comparison, Figure 5.8 reveals that more raw spectra from TVAE have p values higher than the significance level (≥ 0.05) compared to those produced by CTGAN. Nevertheless, many spectra still exhibit p values below the significance level (<0.05), following the same pattern of similarity identification.

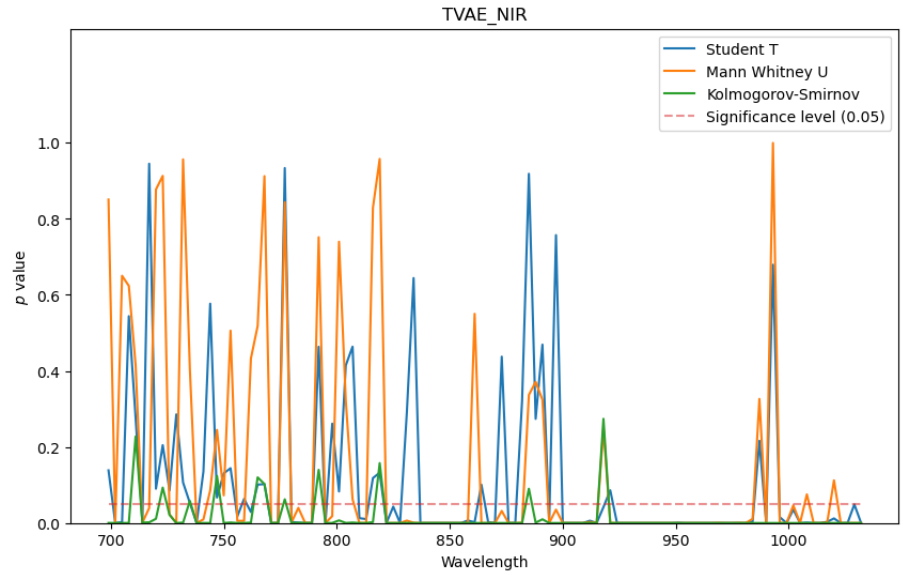


Figure 5.8: Lineplot that shows p values from different univariate statistical tests when comparing real data with raw TVAE-generated data

Upon examining the smoothed spectra, Figure 5.9 shows that CTGAN-generated spectra have slightly more columns with p values above the significance level (≥ 0.05). However, the majority

of columns still fall below the significance value, leading us to reject the null hypothesis that they are similar to the real data.

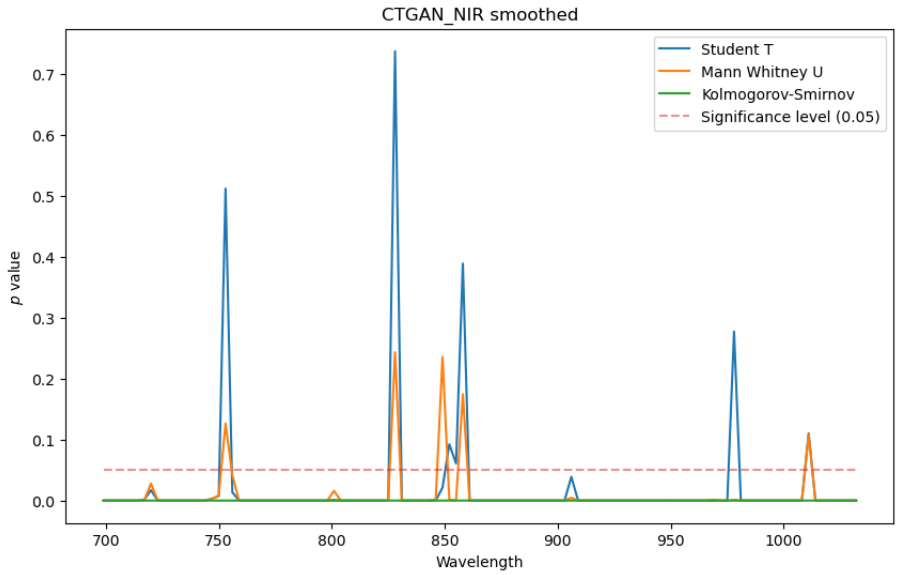


Figure 5.9: Lineplot that shows p values from univariate statistical tests when comparing real data with smoothed CTGAN generated data. The tests' null hypothesis is that the data is similar for attribute tested for, and we accept it if p value is higher than the significance level, which is typically set at 0.05

This pattern is also observed in Figure 5.10, where the smoothed spectra from TVAE display more columns with p values higher than those of the smoothed spectra from CTGAN. Furthermore, the pattern observed in the smoothed TVAE spectra differs from that of the raw TVAE spectra, with more columns clustering together and exhibiting p values (≥ 0.05).

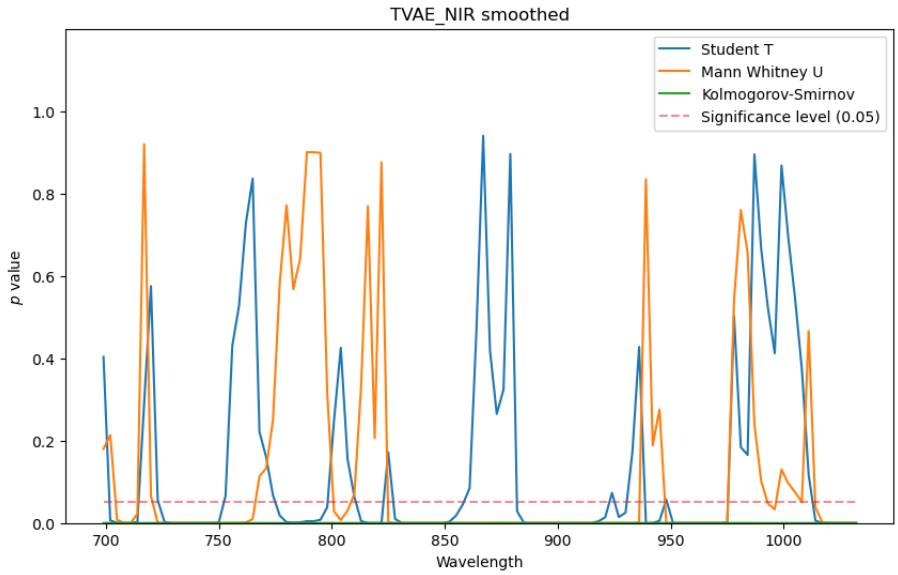


Figure 5.10: Lineplot that shows p values from univariate statistical tests when comparing real data with smoothed TVAE generated data. The tests' null hypothesis is that the data is similar for attribute tested for, and we accept it if p value is higher than the significance level, which is typically set at 0.05

Table 5.2 and table 5.3 show the mean test-value and percentage of them that are significant, respectively.

	Raw		Smooth	
	CTGAN_NIR	**TVAE_NIR**	**CTGAN_NIR**	**TVAE_NIR**
Kolmogorov-Smirnov	0.000	0.013	0.000	0.000
Student T	0.020	0.111	0.020	0.141
Mann-Whitney U	0.011	0.148	0.009	0.128

Table 5.2: The mean p values when comparing raw and smoothed synthetic spectra with real data using different statistical tests

Table 5.3: Percentage of p values that are above the significance level of 0.05 for the different statistical tests for raw and smoothed synthetic spectra

	Raw		Smooth	
	CTGAN_NIR	**TVAE_NIR**	**CTGAN_NIR**	**TVAE_NIR**
Kolmogorov-Smirnov	0.000	9.821	0.000	0.000
Student T	3.571	33.929	6.250	35.714
Mann-Whitney U	1.786	27.679	4.464	29.464

When we take a look at the mean p values and percentages of p values above the significance level, we can see that TVAE seems to outperform CTGAN in terms of generating spectra that are more statistically similar to the real data. This holds true for both raw and smoothed spectra. It's also interesting to note that smoothing the spectra leads to an improvement in statistical similarity across the board. This suggests that the smoothing process is helpful in bridging the gap between synthetic and real data.

Among the three statistical tests, the KS test appears to be the most sensitive to differences in the distributions. This test consistently results in the lowest mean p values and percentages of p values above the significance level. On the other hand, the Student T and MWU tests seem to be less sensitive to such differences, as they generally produce higher mean p values and percentages of p values above the significance level.

The results indicates that TVAE-generated spectra, particularly when smoothed, demonstrate a higher degree of statistical similarity to the real data compared to CTGAN-generated spectra.

Cosine Similarity

Because the data worked with is very high-dimensional, it is not practical to showcase univariate statistics for each feature. As this section tries to establish degree of similarity between the datasets, the similarity between the statistical properties of raw spectra generated by CTGAN and TVAE and the real was determined using cosine similarity.

As shown in Table 5.4, the cosine similarity between the aggregate statistics of the features of the raw spectra generated by CTGAN and TVAE is quite high, indicating that they closely resemble the real data, which suggests that both CTGAN and TVAE are successful in capturing the overall statistical properties of the real spectra. Most of the scores are near 1, which suggests identical directions in the multi-dimensional space of the vectors. The exceptions to this are skew and kurtosis, which score slightly lower, though they are still relatively high.

Table 5.4: Cosine similarity between summary statistics of NIR spectra from real and raw CTGAN and TVAE spectra

	CTGAN_NIR	TVAE_NIR
std	0.986167	0.998840
min	0.999995	0.999880
25%	0.998402	0.999977
50%	0.998050	0.999982
75%	0.997646	0.999971
max	1.000000	0.999998
skew	0.931373	0.967953
kurtosis	0.832141	0.949244

A similar pattern is observed in the smoothed datasets, as shown in Table 5.5. The majority of the aggregate metrics are nearly identical, with only skew and kurtosis showing slightly more dissimilarity, though they still remain fairly similar to the real data.

Table 5.5: Cosine similarity between summary statistics of NIR spectra from real and smoothed CTGAN and TVAE spectra

	CTGAN_NIR	TVAE_NIR
std	0.986786	0.996622
min	0.999738	0.999879
25%	0.998814	0.999779
50%	0.999363	0.999667
75%	0.999506	0.999626
max	0.998156	0.998648
skew	0.958181	0.936569
kurtosis	0.887498	0.892053

The overall high cosine similarity clearly indicates that CTGAN and TVAE are capable of generating synthetic spectra that closely mimic the aggregate statistics found in the real data. They achieve near-perfect cosine similarity scores for all metrics, except skew and kurtosis, where they still score quite high. This suggests that CTGAN and TVAE show promise in synthesizing NIR spectroscopy data while maintaining the essential statistical properties of the real dataset.

One potential area to explore further is the reason behind the lower cosine similarity scores for skew and kurtosis, as compared to other metrics. It would be beneficial to investigate whether this discrepancy is due to the nature of the synthetic data generation process or if it arises from other factors. By gaining a better understanding of this aspect, it may be possible to improve the performance of CTGAN and TVAE in generating synthetic spectra that closely resemble the real data.

5.1.3 Multivariate Relationships Analysis

In this subsection, we will discuss the multivariate relationships in the results, focusing on the relationship between the target variable (Dry Matter Content) and the features (NIR spectra), as well as the pairwise Pearson correlations between the features. Analyzing these relationships is crucial for understanding how well the synthetic data generated by CTGAN and TVAE preserve the real data's structure and relationships.

Correlations between Target and Features

To assess the relationship between the features and target variable, we plotted the correlation between each feature and the target variable, with the correlation on the y-axis and wavelength on the x-axis. This is shown in Figures 5.11 and 5.12. CTGAN was not able to effectively preserve the target-feature relationship, whereas TVAE managed to maintain the trend, although it was somewhat rough. The smoothing process did not improve the relationship for CTGAN, but smoothed TVAE spectra got quite close to the real pattern.

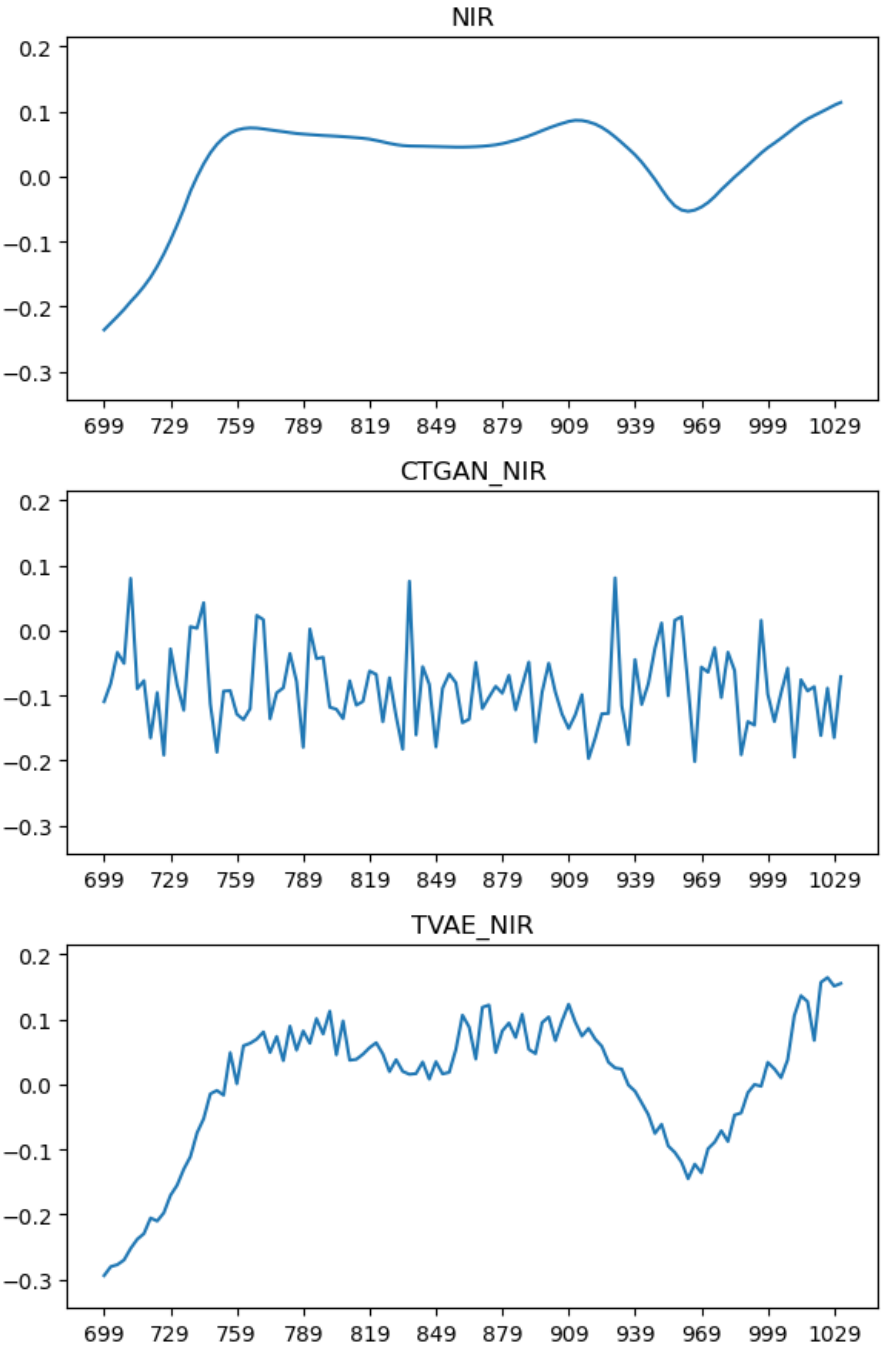


Figure 5.11: Lineplots that shows the correlations between spectra and target for real and raw synthetic datasets

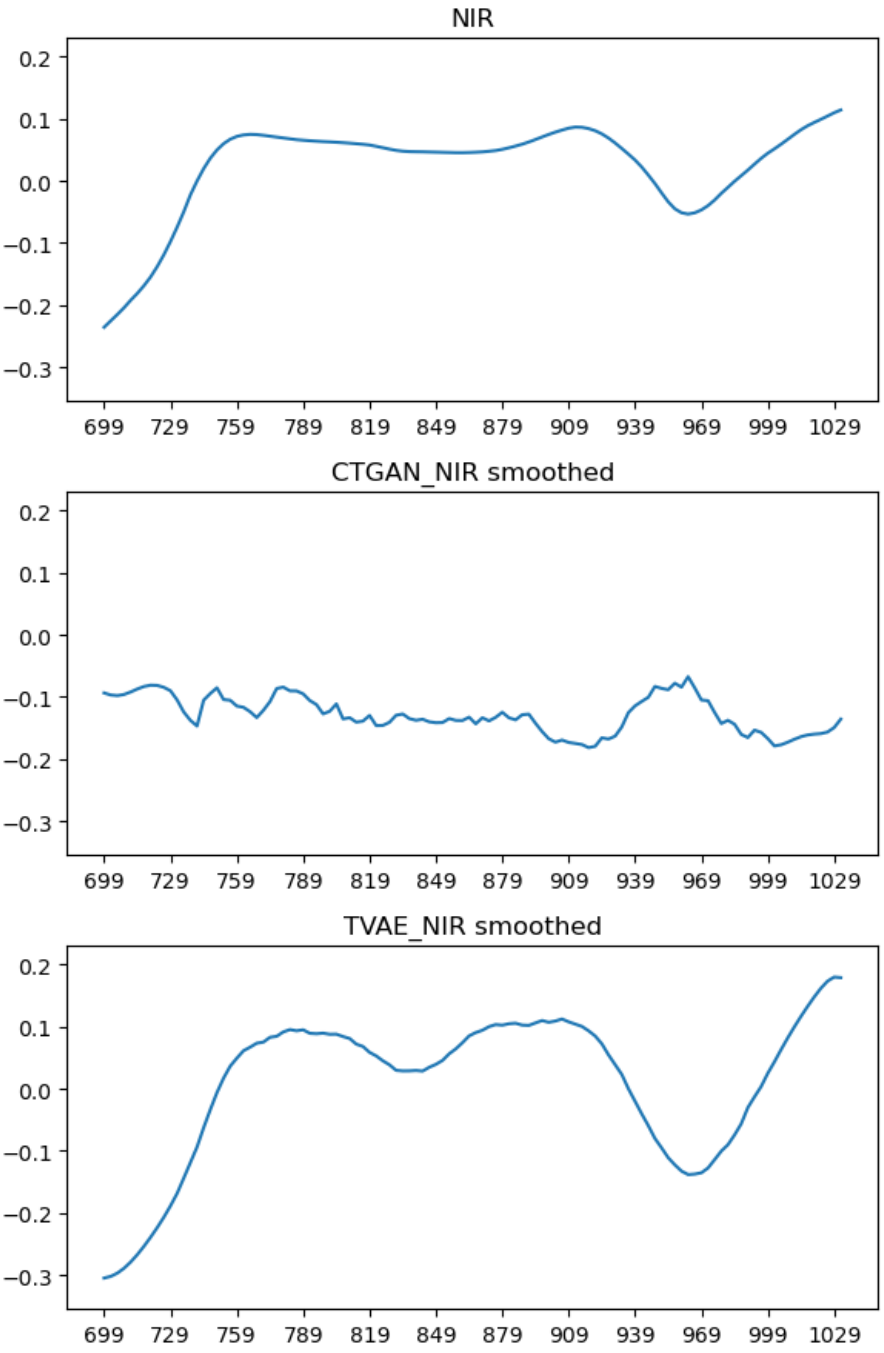


Figure 5.12: Lineplots that shows the correlations between spectra and target for real and smoothed synthetic datasets

Subsequently, we computed the cosine similarity between the target-feature correlations of the real data and the synthetic data generated by both CTGAN and TVAE, in both raw and smoothed conditions. The cosine similarity scores are presented in Table 5.6.

Table 5.6: Comparison of Cosine Similarity for CTGAN_NIR and TVAE_NIR in Two Datasets

	Raw		Smooth	
	CTGAN_NIR	**TVAE_NIR**	**CTGAN_NIR**	**TVAE_NIR**
Cos Sim	-0.336	0.92	-0.401	0.936

As seen in table 5.6, the cosine similarity scores reveal how well the synthetic data preserves the target-feature relationships compared to the real data. As mentioned previously, higher cosine similarity scores indicate better preservation of these relationships.

Correlations within spectra

Next, we investigated the pairwise correlations between the spectra. We created heatmaps of the Pearson correlation matrices for the real spectra, CTGAN spectra, and TVAE spectra. This was done for both raw synthetic spectra (Figure 5.13) and smoothed synthetic spectra (Figure 5.14). We observed that both CTGAN and TVAE captured the pattern of the relationships, although the intensity did not match perfectly. TVAE seemed to capture the pattern and intensity better than CTGAN, and smoothing improved both measures for both generative models. The correlations between neighboring spectra are understandably high in the real data, particularly in the middle portion, due to the high degree of smoothness exhibited. The less smooth spectra produced by TVAE and especially CTGAN are obviously less likely to capture the same level of correlation between neighboring spectra.

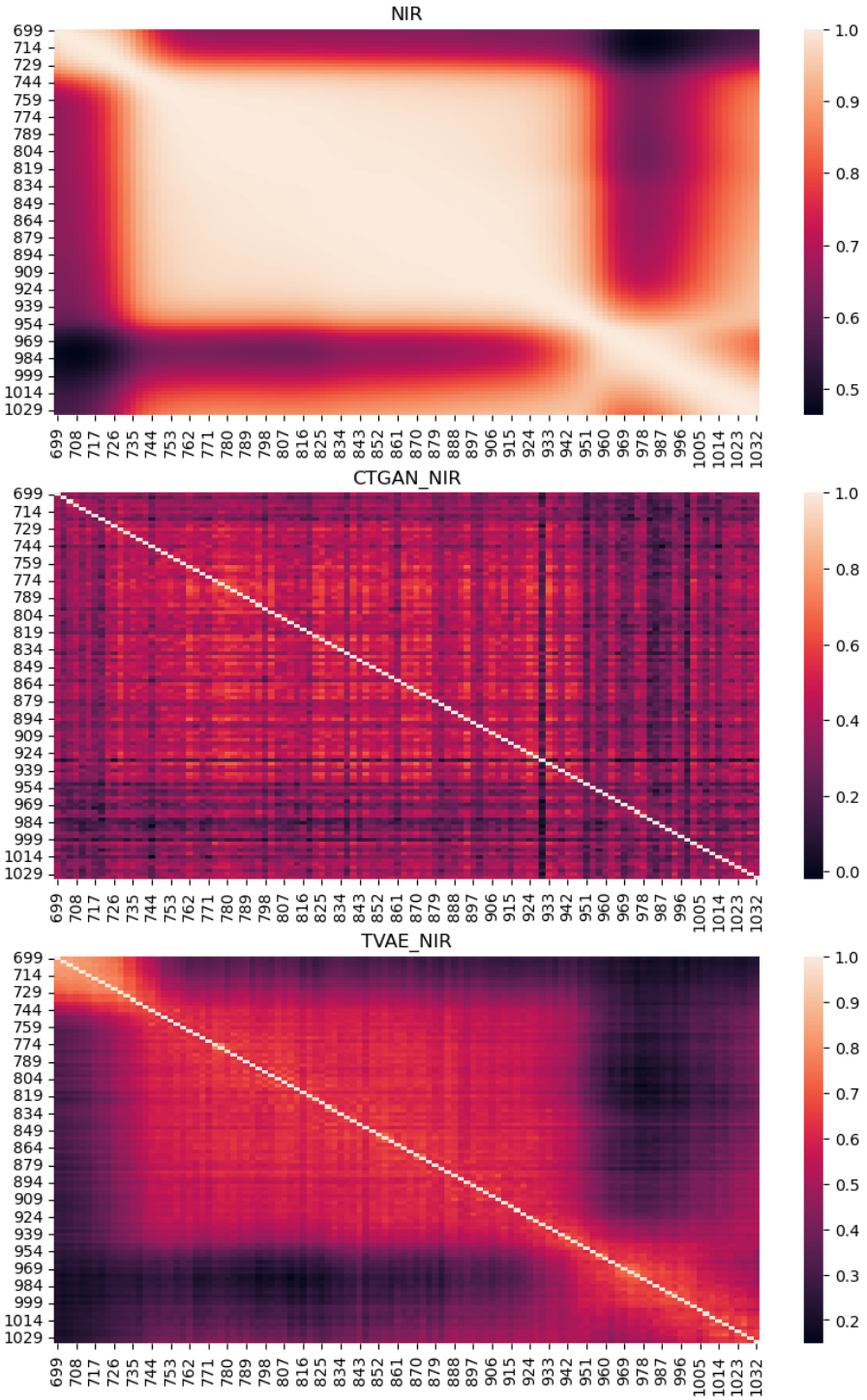


Figure 5.13: Heatmaps that visualize the correlations between the various wavelengths. We can see that TVAE have been able to capture the pattern of correlations between the wavelengths, but not match the intensity. CTGAN have been poor at maintaining the integrity of the multivariate relationships from the real data

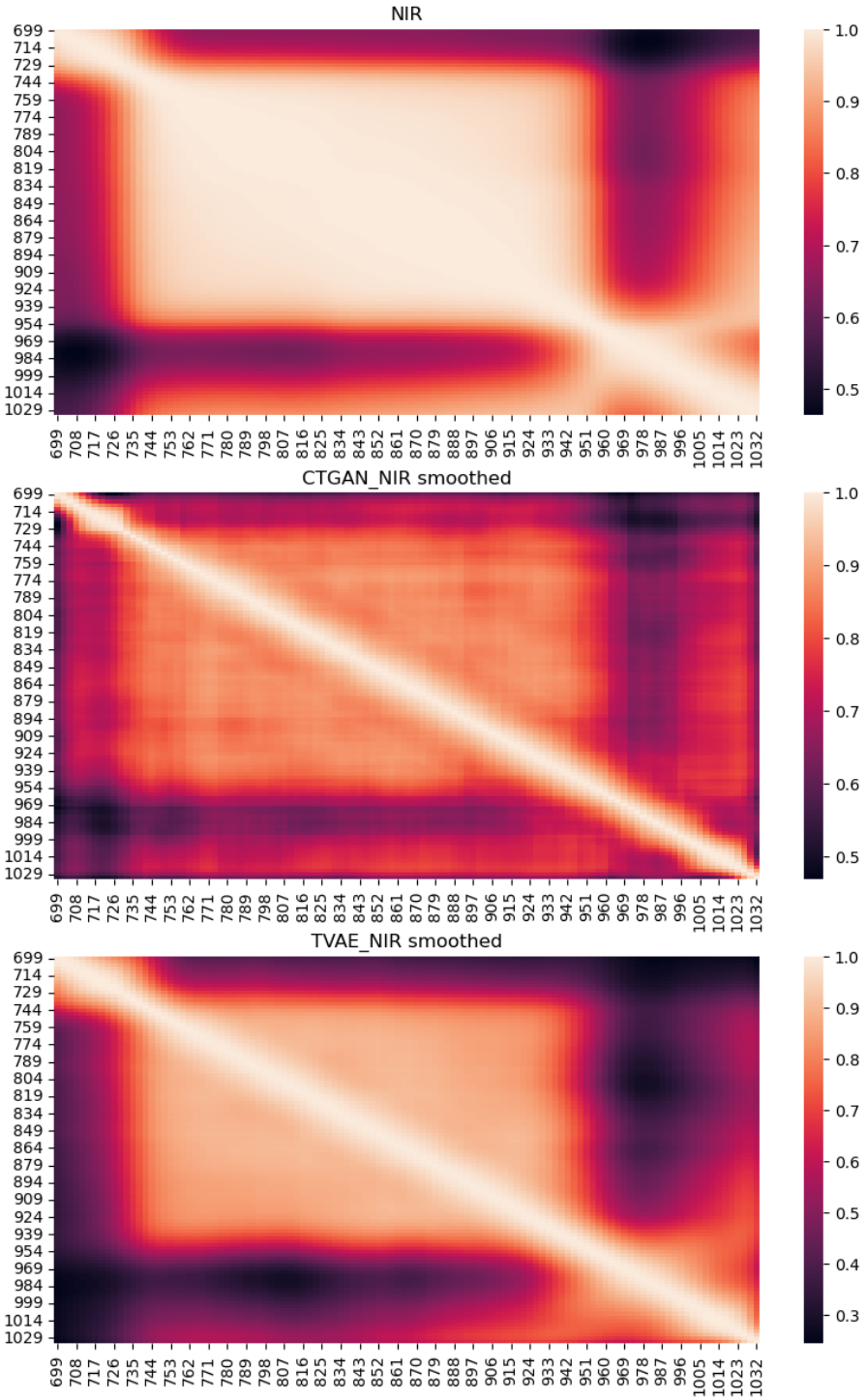


Figure 5.14: Heatmaps of pearson correlation matrix for real and smoothed synthetic datasets

We also computed the cosine similarity between pairwise Pearson correlations for all datasets, with the results presented in Table 5.7.

	Raw		Smooth	
	CTGAN_NIR	**TVAE_NIR**	**CTGAN_NIR**	**TVAE_NIR**
Mean	0.471212	0.384182	0.079860	0.174313
Min	0.066943	0.154251	0.000083	0.000038
Max	0.935732	0.533228	0.471514	0.349565
Cos Sim	0.967834	0.989142	0.997282	0.988119

Table 5.7: Summary statistics of differences in pairwise pearson correlations between spectra from real and synthetic data (raw and smoothed), as well as the cosine similarity between them

The table shows the summary statistics of the differences in pairwise Pearson correlations between the spectra from the real and synthetic datasets, as well as the cosine similarity between them. These statistics help assess the preservation of the pairwise relationships between the features in the synthetic data generated by CTGAN and TVAE.
Overall, the cosine similarity scores in Table 5.7 demonstrate that both CTGAN and TVAE are able to maintain the pairwise relationships between the spectra in the synthetic data to a considerable extent. Notably, the cosine similarity scores are slightly higher for the smoothed CTGAN spectra, but a tiny bit lower for the smoothed TVAE spectra, when comparing with the raw counterpart.
The cosine similarity scores in Table 5.7 reveal that both CTGAN and TVAE can maintain the pairwise relationships between the spectra in the synthetic data to a considerable extent. Interestingly, the scores are slightly higher for the smoothed CTGAN spectra but marginally lower for the smoothed TVAE spectra compared to their raw counterparts. Our analysis of multivariate relationships highlights the potential of CTGAN and TVAE in synthesizing NIR spectroscopy data that retains the structure and relationships present in the real data. The cosine similarity scores for target-feature relationships and pairwise Pearson correlations between the features demonstrate the synthetic data's ability to largely preserve the relationships found in the real data. This finding is particularly encouraging for future applications of these techniques in synthesizing and analyzing NIR spectroscopy data.

5.1.4 Dimensional Resemblance Analysis

To investigate the similarity between the real and synthetic datasets, we applied dimensional analysis using both linear (PCA) and non-linear (Isomap) dimensionality reduction techniques. The PCA score plots for the raw and smoothed datasets are shown in Figures 5.15 and 5.16, respectively, using the first two principal components as the axes. Figure 5.15 shows that the raw CTGAN spectra have a similar oval shape as the real data, but are more compact. The raw TVAE spectra have a similar size, but are not as elongated as the real. Figure 5.16 shows that the smoothed CTGAN spectra are more similar in size to the real data and retain a recognizable, but similar oval shape. The smoothed TVAE spectra are similar to the raw TVAE spectra, but are more oval-like like the real data.

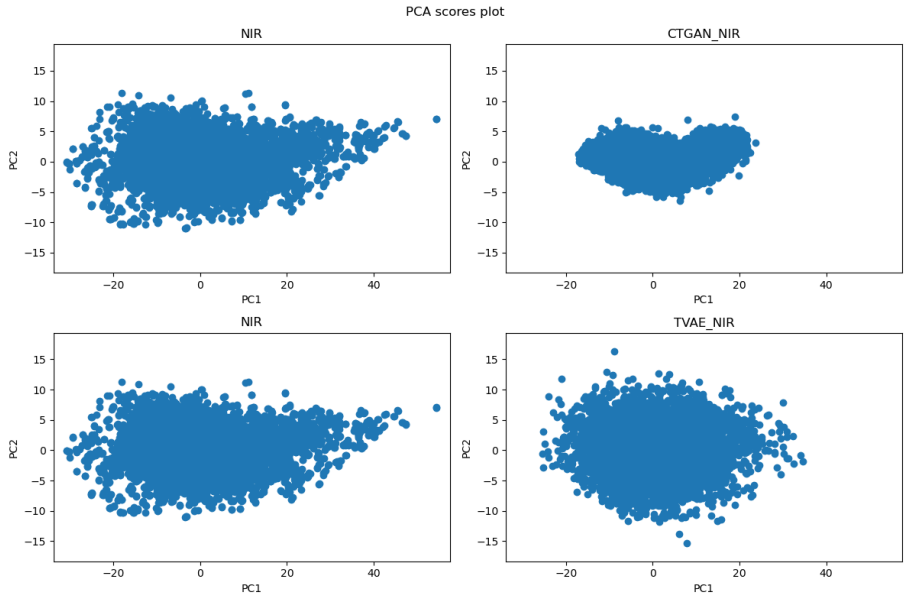


Figure 5.15: PCA scores plot for all three datasets with two first principal components as axes

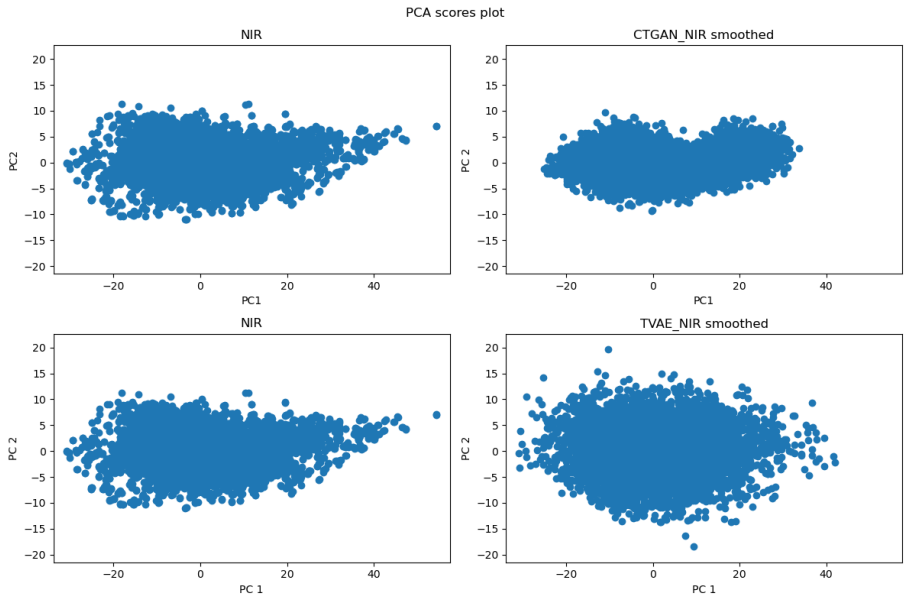


Figure 5.16: PCA score plot of real spectra and smoothed synthetic spectra with two first principal components as axes

The projection of the raw spectra from CTGAN and TVAE is displayed in Figure 5.17. We observe that the projected spectra produced by CTGAN closely resemble the raw spectra from CTGAN in Figure 5.15; they exhibit a somewhat similar oval shape as the real but can be easily distinguished by their size. The projected spectra generated by TVAE also bear a resemblance to the raw TVAE spectra shown in Figure 5.15, closer in size to the real than CTGAN spectra, but they do not quite capture the elongated shape.

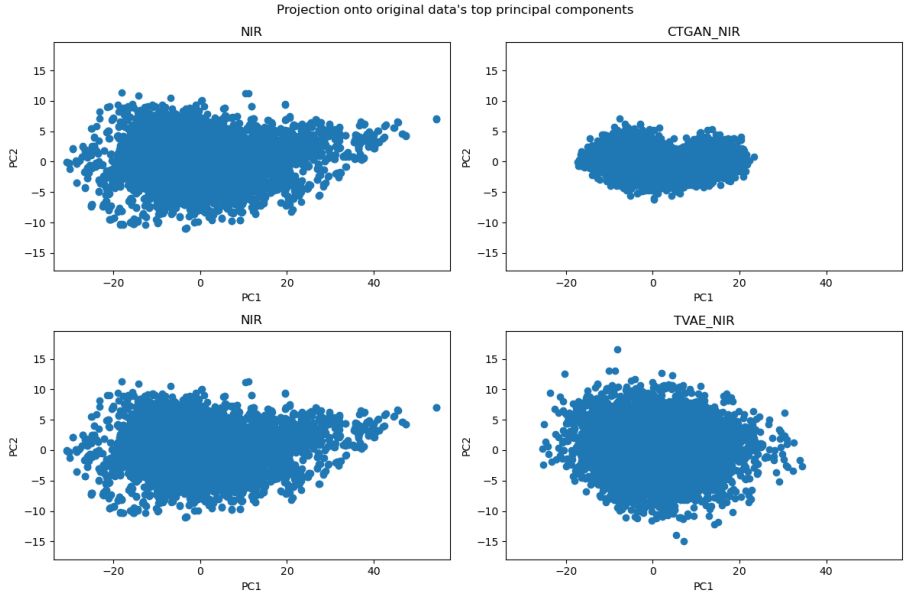


Figure 5.17: Raw synthetic data projected onto the top 2 principal components of the real data

The projection of the corresponding smoothed spectra, displayed in Figure 5.18, performs better. The size and shape of the smoothed CTGAN spectra become much closer to the real, and the smoothed TVAE spectra become more elongated along principal component 1 like the real spectra, but they are stretched further along second principal component than the real spectra.

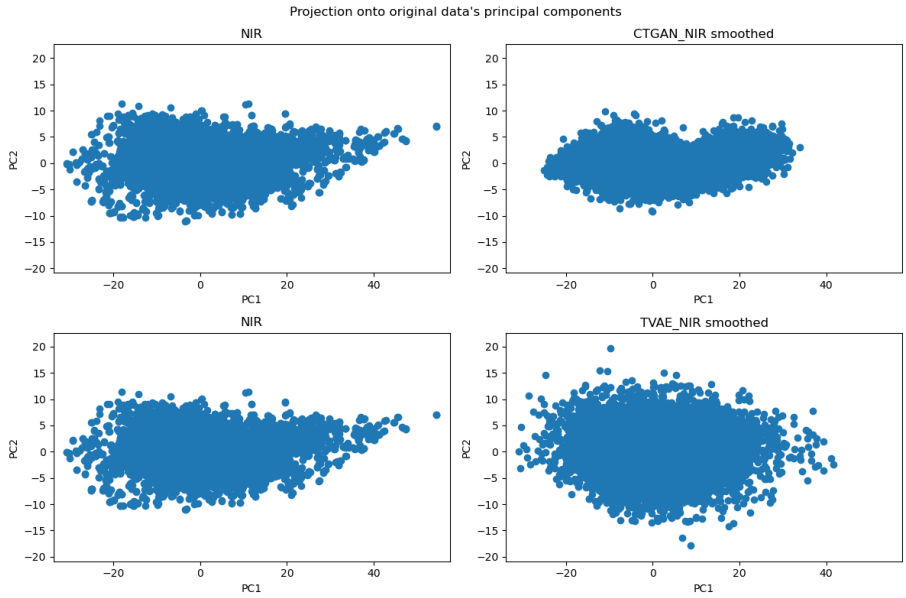


Figure 5.18: Smoothed synthetic data projected onto the top 2 principal components of the real data

Figure 5.19 displays a scree plot that explains 95% of the variance after the PCA decomposition of the raw spectra generated by CTGAN and TVAE. We observe that only three principal components are required to achieve this for the real data, while the raw synthetic spectra from CTGAN and TVAE demand significantly more (84 and 87 principal components are needed to explain at least 95% of the variance in raw CTGAN and TVAE spectra, respectively).

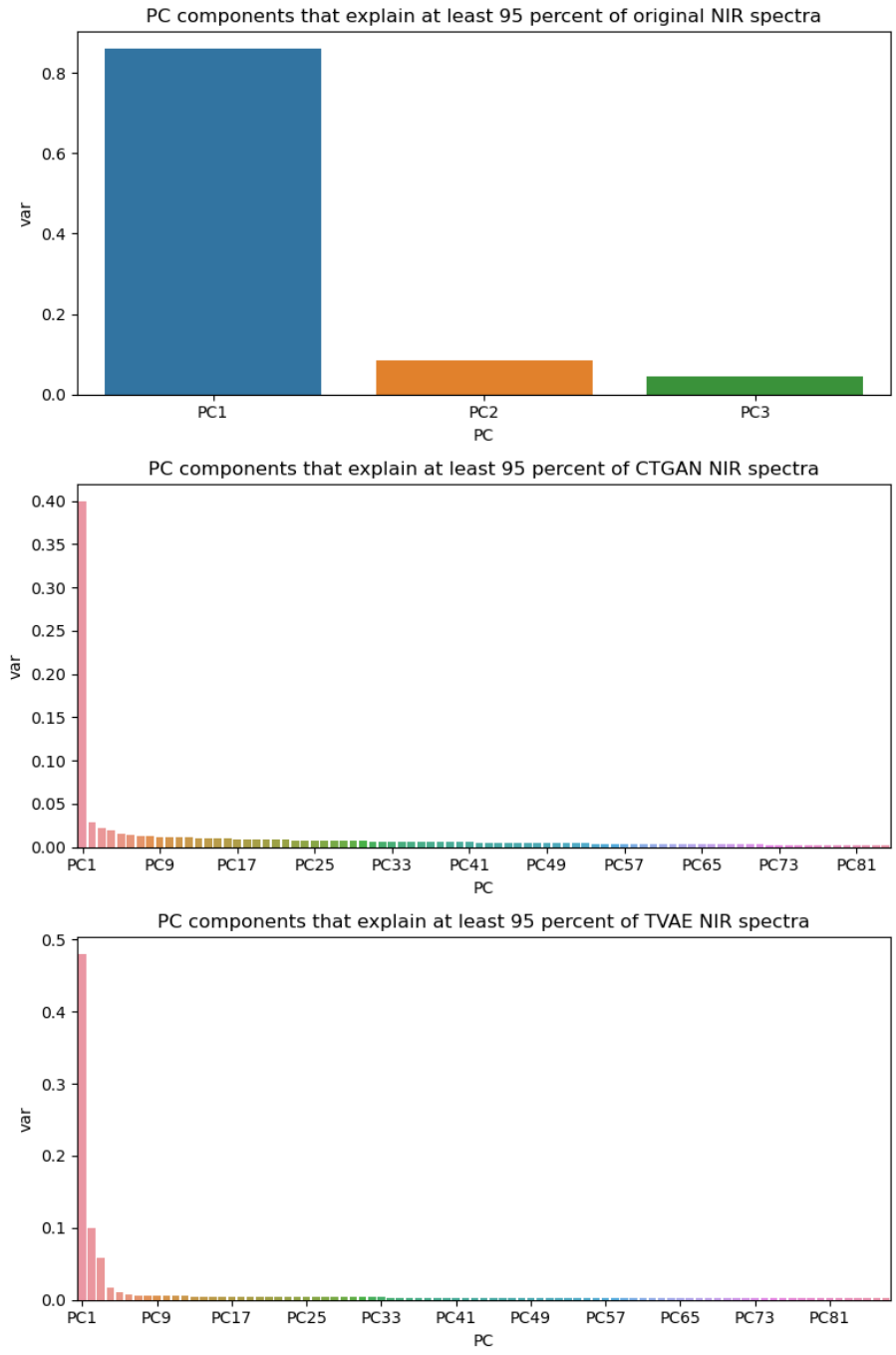


Figure 5.19: Scree plot of principal components that explains at least 95% of variance in the real and raw synthetic data

Figure 5.20 presents the corresponding scree plot for the smoothed synthetic data. Smoothing the data leads to a substantial reduction in the number of principal components needed to explain at least 95% of the variance (8 for smoothed CTGAN spectra and 5 for smoothed TVAE spectra).

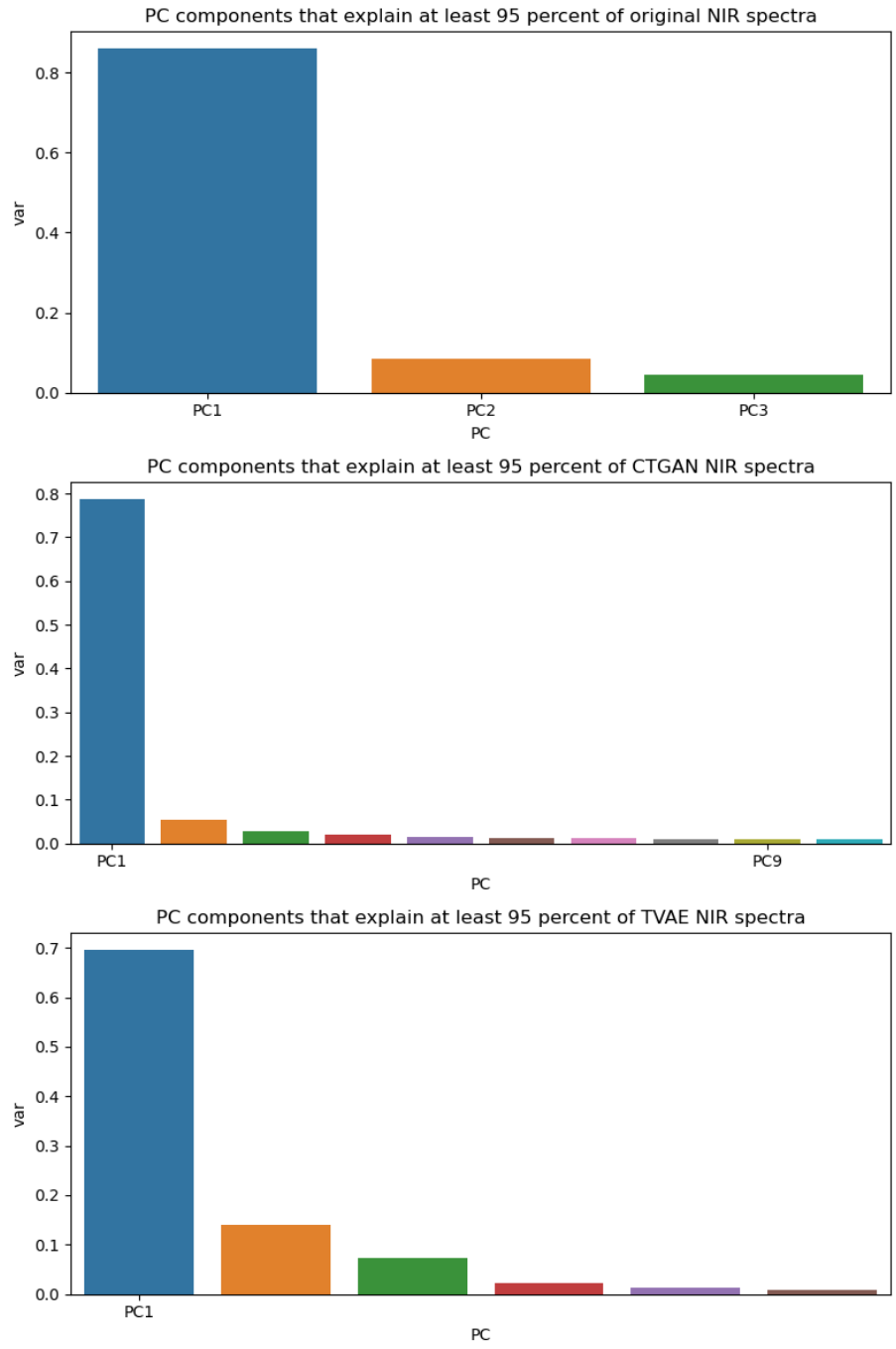


Figure 5.20: Scree plot of principal components that explains at least 95% of variance in the real and smoothed synthetic data

The observed relationship between the amount of variance explained by the principal components in the scree plots and the extent to which the data is spread along these principal components in the PCA score plots is consistent with our expectations. As the principal components capture the directions of maximum variance in the data, it is natural to see a correspondence between the explained variance and the spread of data points along these components in the score plots.
We plotted the synthetic and real spectra on their corresponding lower-dimensional Isomap manifold, with raw spectra displayed in Figure 5.21 and smoothed spectra in Figure 5.22. Neither of the raw spectra captured the shape of the real spectra well, but CTGAN appeared slightly more faithful. Smoothing the CTGAN spectra improved its dimensional resemblance

to the real, but this improvement was not observed for the TVAE spectra. Interestingly, both the raw and smoothed TVAE spectra were quite similar to each other.

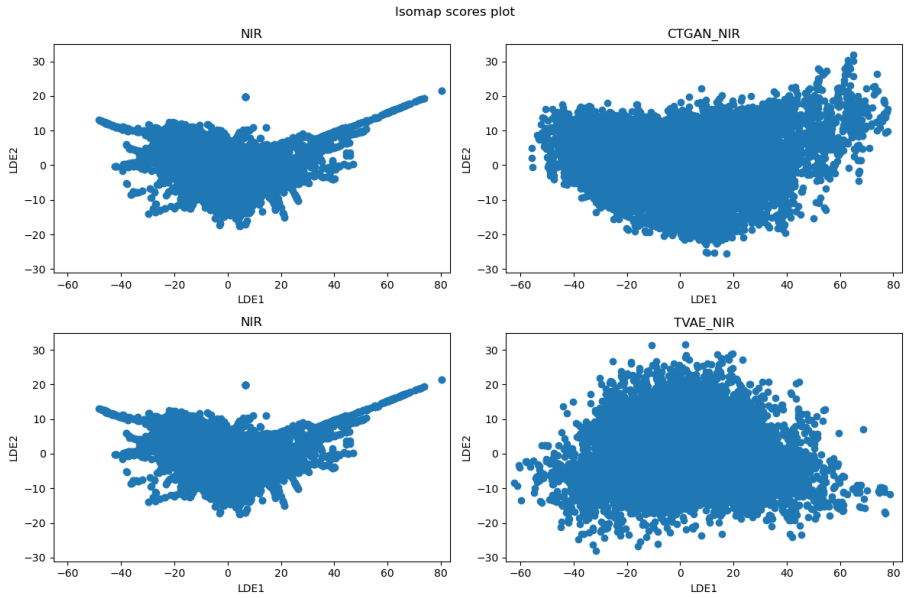


Figure 5.21: Isomap score plot of raw and real spectra against their own top lower-dimensional embeddings as axes

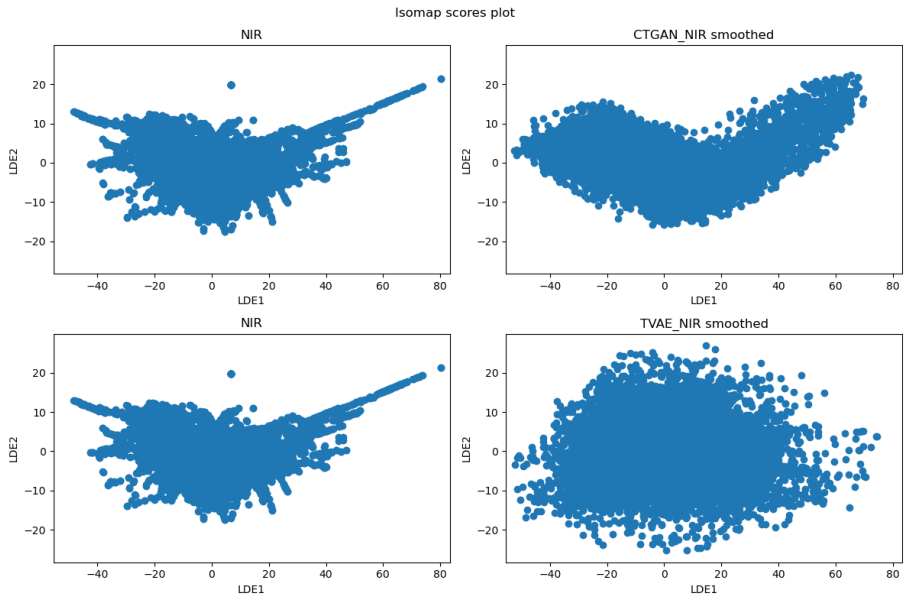


Figure 5.22: Isomap score plot of smooth and real spectra against their own top lower-dimensional embeddings as axes

Additionally, we projected the synthetic spectra onto the lower-dimensional embeddings obtained from the Isomap reduction of the real data. The Isomap projection of the raw spectra can be seen in Figure 5.23, while the Isomap projections of the smoothed spectra are displayed in Figure 5.24. The projections of both CTGAN and TVAE successfully captured the distinct characteristics of the real data's shape. However, both models displayed a similar rotation relative to the real data and did not exhibit the same level of spread along the top two lower-dimensional embeddings. The corresponding raw and synthetic spectra shared similar shape

and rotation, but the projected data points from the smoothed data were more dispersed than their raw counterparts. This indicates that the top two lower-dimensional embeddings account for a greater proportion of the nonlinear variance in the real data compared to the synthetic data, although smoothing assists in capturing more of this variance.

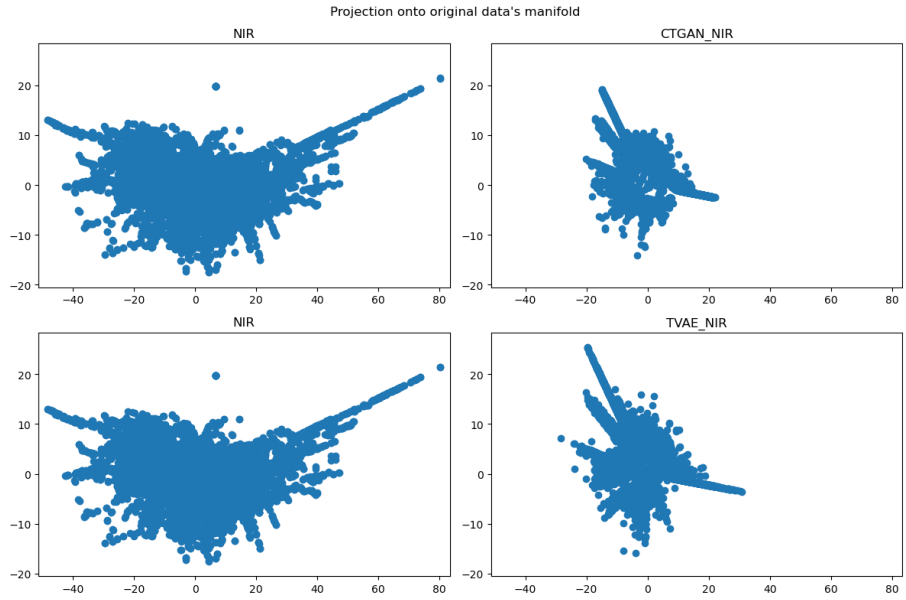


Figure 5.23: Isomap score plot of real spectra and raw spectra against top lower-dimensional embeddings from real spectra as axes

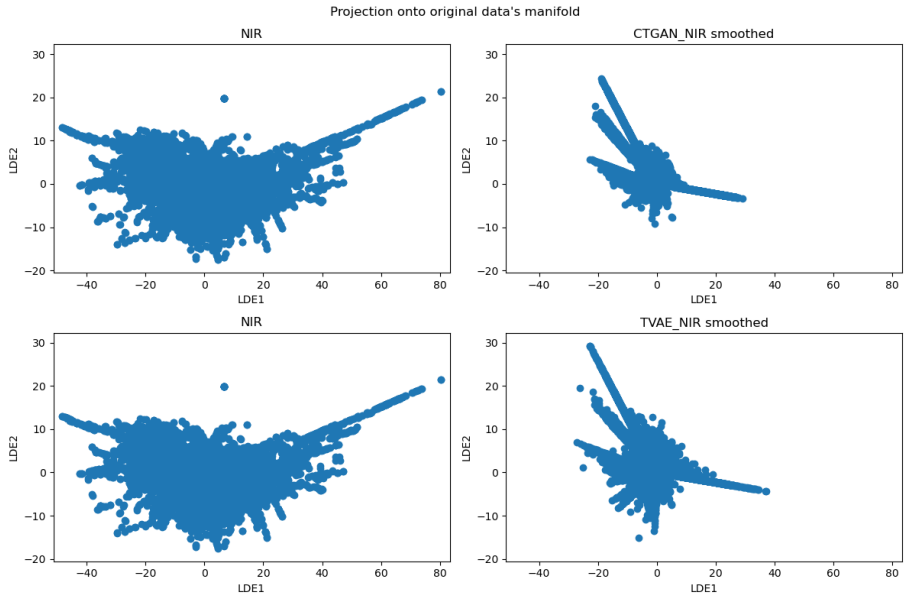


Figure 5.24: Isomap score plot of real spectra and smoothed spectra against top lower-dimensional embeddings from real spectra as axes

5.1.5 Data Labeling Analysis

In this section, we present the results of the Data Labeling Analysis performed on the synthetic spectra generated by CTGAN and TVAE, both raw and smoothed. This analysis evaluates the

semantic resemblance between the real and synthetic data by assessing the performance of various ML classifiers in distinguishing between the real and synthetic records. Lower classification accuracy implies higher fidelity of the synthetic data, as it indicates that the classifiers struggle to differentiate between the real and synthetic records.

Tables 5.8, 5.9, 5.10, and 5.11 present the performance metrics (Accuracy, F1, Precision, and Recall) of five different ML classifiers (RF, KNN, DT, (SVM), and MLP when applied to the raw and smoothed synthetic spectra generated by CTGAN and TVAE, respectively.

For the raw synthetic spectra generated by CTGAN (Table 5.8), RF and MLP achieved perfect classification scores, indicating strong classifiers are clearly able to distinguish between real and synthetic records. SVM and DT score overall high too. KNN, however, had a lower classification scores, suggesting some degree of similarity between the real and synthetic data.

	Accuracy	**F1**	**Precision**	**Recall**
RF	1.0	1.0	1.0	1.0
KNN	0.760154	0.684654	0.520513	1.0
DT	0.99658	0.996581	0.996581	0.996581
SVM	0.999145	0.999145	0.998291	1.0
MLP	1.0	1.0	1.0	1.0

Table 5.8: Evaluation of ML algorithms performance at discriminating between real and raw synthetic NIR Spectra generated by CTGAN

Regarding the raw synthetic spectra generated by TVAE (Table 5.9), RF and MLP exhibited near-perfect classification scores. KNN and SVM, on the other hand, demonstrated significantly lower scores, except for the again perfect recall for KNN and near perfect precision for SVM, which indicates raw TVAE spectra resemble real spectra more than the raw CTGAN spectra. It is intriguing that KNN again demonstrates perfect recall but otherwise overall poor classification performance. One can be tempted to believe that KNN classifies nearly all records as the same class.

	Accuracy	**F1**	**Precision**	**Recall**
RF	0.999572	0.999572	0.999145	1.0
KNN	0.501496	0.006814	0.003419	1.0
DT	0.969645	0.969251	0.95641	0.982441
SVM	0.501496	0.656857	0.953846	0.500898
MLP	1.0	1.0	1.0	1.0

Table 5.9: Evaluation of ML algorithms performance at discriminating between real and raw synthetic NIR Spectra generated by TVAE

When considering the smoothed synthetic spectra generated by CTGAN (Table 5.10), the classification scores were still generally high, with MLP achieving perfect accuracy and RF nearly so, indicating that smoothing did not help fool RF and MLP. DT and SVM also displayed strong classification metrics yet again. Even KNN improved its classification scores, suggesting that the resemblance between real and synthetic data was not improved by smoothing, rather the contrary.

	Accuracy	F1	Precision	Recall
RF	0.998717	0.998716	0.997436	1.0
KNN	0.886704	0.872289	0.773504	1.0
DT	0.990167	0.990133	0.986325	0.993971
SVM	0.992304	0.992353	0.998291	0.986486
MLP	1.0	1.0	1.0	1.0

Table 5.10: Evaluation of ML algorithms performance at discriminating between real and smoothed synthetic spectra generated by CTGAN

Lastly, for the smoothed synthetic spectra generated by TVAE (Table 5.11), the strong classifiers RF and MLP maintained high classification accuracies. DT also maintained a high overall scores. SVM and KNN showed a marked increase in classification scores compared to the raw TVAE spectra, implying that smoothing had a limited impact on the fidelity of the synthetic data in this case.

	Accuracy	F1	Precision	Recall
RF	0.988884	0.988851	0.98547	0.992255
KNN	0.683198	0.538318	0.369231	0.993103
DT	0.952116	0.951431	0.937607	0.965669
SVM	0.910646	0.915419	0.966667	0.869331
MLP	0.999572	0.999572	0.999145	1.0

Table 5.11: Evaluation of ML algorithms performance at discriminating between real and smoothed synthetic NIR Spectra generated by TVAE

In our data labeling analysis, we discovered that smoothing the synthetic data did not make it more difficult for classifiers to distinguish between real and synthetic data. In some instances, such as with KNN, the classifier's performance actually improved. Interestingly, KNN achieved a perfect recall but had relatively lower scores in other performance metrics. One potential explanation for this behavior could be that the KNN classifier is biased towards predicting one class over the other. This hypothesis is further supported by the fact that KNN achieved a classification accuracy close to 50% when classifying raw TVAE spectra in a balanced dataset. A 50% classification accuracy suggests that the classifier is unable to distinguish between the samples effectively. These findings highlight the importance of considering different classifiers when evaluating synthetic data fidelity, as well as the potential limitations of using synthetic data in real-world applications. Further research could investigate techniques to mitigate classifier bias or improve the overall performance of generative models, ensuring that the synthetic data is more representative of the real data across various performance metrics.

5.2 Utility

In this section, we assess the utility of the synthetic data generated by CTGAN and TVAE by evaluating the performance of ML algorithms trained on both real and synthetic data. Using the TRTR and TSTR methodology, we compared the performance of five regressor variants of the classifiers employed in the data labeling analysis: RF, KNN, DT, SVM, and MLP. The coefficient of determination (R^2), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE) were calculated for each classifier trained on the real, raw synthetic, and smoothed synthetic data.
Table 5.12 shows the R^2 values for the ML algorithms trained on the real spectra and the raw spectra produced by CTGAN and TVAE. The results demonstrate that the models trained on

the real data generally outperform those trained on the synthetic data. Notably, the RF classifier trained on TVAE-generated data achieved a moderately high R^2 value of 0.180346, while the same classifier trained on CTGAN-generated data yielded a negative R^2 value, indicating that the model is not able to predict the target variable accurately. Our SVM regressor performed poorly at predicting DMC from all spectra, and intriguingly performed worst with the real NIR spectra.

	NIR	**CTGAN_NIR**	**TVAE_NIR**
RF	0.667707	-0.006353	0.180346
KNN	0.372196	-0.146006	0.133436
DT	0.292846	-1.322496	-0.255557
SVM	-8.574527	-3.601274	-2.544479
MLP	0.708510	-1.269254	0.002788

Table 5.12: Comparing ML algorithms trained on real spectra against raw spectra produced by CTGAN and TVAE on performance on same holdout subset of real data

Table 5.13 presents the R^2 values for the ML algorithms trained on the real spectra and the smoothed spectra produced by CTGAN and TVAE. Similar to the results obtained with the raw synthetic data, the models trained on the real data outperform those trained on the smoothed synthetic data. However, the RF classifier trained on TVAE-generated smoothed data achieved a higher R^2 value of 0.189407 compared to the same classifier trained on the raw TVAE-generated data. The SVM regressor still performed quite poorly at predicting the DMC when trained on smoothed spectra, performing worse than a baseline model.

	NIR	**CTGAN_NIR**	**TVAE_NIR**
RF	0.667707	-0.209106	0.189407
KNN	0.372196	-0.294911	0.150815
DT	0.292846	-2.450614	-0.388260
SVM	-8.574527	-14.083039	-2.277547
MLP	0.708510	-0.409346	-0.041049

Table 5.13: Comparing ML algorithms trained on real spectra against smoothed spectra produced by CTGAN and TVAE on performance on same holdout subset of real data

The utility assessment indicates varying results for the synthetic data generated by TVAE and CTGAN. For TVAE, some promise was shown, particularly for RF and KNN. However, the synthetic data did not yet reach the level of utility necessary to replace real data for training ML models. In contrast, CTGAN consistently achieved negative scores. This result suggests that models trained on synthetic data generated by CTGAN perform worse than a simple baseline model, such as predicting the mean of the target variable for all instances.

Chapter 6

Discussion

Our discussion centers around understanding and analyzing the outcomes of our research, in which we scrutinized the fidelity and utility of synthetic data generated by CTGAN and TVAE for potential use in ML applications within NIR spectroscopy. We leveraged a multitude of techniques to evaluate the synthetic data's quality and its viability as a substitute for real data in the training and assessment of ML models.

We examined various facets of fidelity in our analysis, including univariate resemblance, multivariate relationships, dimensional resemblance analysis, and data labeling analysis. The goal of these assessments was to conduct a comprehensive exploration of the similarity between synthetic and real data, considering statistical properties, inter-variable relationships, and overall data structure. Besides fidelity, we also appraised the utility of the synthetic data by gauging its suitability as an alternative to real data in the training of a range of ML models.

As we move forward, we will delve into the implications of our results, ponder over potential reasons behind the observed findings, and propose further work to enhance the quality and utility of synthetic data for NIR spectroscopy applications. We will also confront the constraints of our study and evaluate how these limitations could influence our conclusions.

6.1 Our Findings

This study investigated the capabilities of CTGAN and TVAE from SDV in generating synthetic data that closely resembled real NIR spectral data. Both models successfully produced synthetic data with similar statistical properties, but TVAE demonstrated superior preservation of correlations between variables and maintained the relationship between target and features more effectively than CTGAN, which performed poorly in this regard. TVAE performed significantly better in preserving a subset of flat trend NIR spectra with a similar quantity to the real data, which CTGAN was unable to achieve.

During the data labeling analysis, it was observed that classifiers could easily distinguish between real and CTGAN-generated data, while having a harder time differentiating between real and TVAE-generated data. This finding suggests that the synthetic data produced by TVAE is more similar to the real data.

In the utility assessment, it was revealed that although TVAE-generated synthetic data shows potential, particularly for RF and KNN classifiers, it has not yet reached the level of utility necessary to replace real data for training ML models. Conversely, CTGAN-generated synthetic data performed poorly, with models trained on this data faring worse than a simple baseline

model.

The poor performance by CTGAN could be linked to struggle to replicate the correlation between the spectra and DMC found in the real data. It is intriguing to note this difficulty, given that CTGAN managed to largely replicate the relationships between the spectra. In contrast, TVAE captured all the correlations in the real data quite well, suggesting a higher capacity to model complex relationships in the data.

6.2 Interpretation and Implication of Results

The results demonstrated promise, particularly in more superficial comparisons, such as plots and statistical similarities between real and synthetic data. Although it is possible to distinguish between them, especially between real and raw synthetic spectra which appeared notably rougher, the synthetic data still captured the overall pattern and trend. Moreover, TVAE effectively preserved the Pearson pairwise correlations between features and between features and the target.

However, when using dimensionality reduction and ML techniques to examine the resemblance between real and synthetic data, it became apparent that they were easily distinguishable. It is worth noting that fooling ML classifiers remains a challenge when generating synthetic tabular data [87].

The synthetic spectra did not show significant potential as a replacement for real spectra when training ML regressors, as they yielded much lower scores compared to the real data. This was particularly true for CTGAN, which performed worse than a baseline model for all ML models. Upon examining the correlation between the target and features in the CTGAN dataset, we found that it did not capture the relationship between target and features well, despite doing a decent job at preserving the correlation between features. However, it did not preserve this correlation as well as TVAE, resulting in CTGAN producing rougher spectra than TVAE, indicating that it generally struggled more than TVAE to capture the correlations between columns. We also noted that CTGAN suffered from unstable loss values, indicating poor performance on this data.

The correlations between the target and features are not particularly high even in the real data, which might partially explain why achieving a high degree of utility is challenging in this case. It is a possibility that regression problems require better data than classification, making it inherently more challenging working with a continuous target. It is possible that a higher degree of utility could be achieved if a different target were chosen.

6.3 Remaining Challenges

The tabular generative models used in this thesis were successful in capturing some important aspects of fidelity but failed significantly in others. While they managed to produce data that appeared promising on a superficial level, ML models could easily distinguish them from each other. Strong classifiers typically perform well in distinguishing real and synthetic tabular records [87], indicating that it is a generally difficult problem yet to be solved.

As for utility, the synthetic data did not seem like a viable substitute for real data. The scores were far worse than those obtained with real data, but this might improve with a different target, such as one that would lead to a classification problem instead of regression. Another option

could be binning the target to transform it into a classification problem, potentially simplifying the task. Additionally, the training of the models might have performed better with different hyperparameters. This thesis did not employ automatic and systematic hyperparameter tuning, as manual experimental adjustments did not appear to yield promising results.

Another challenge faced in our study relates to the inherent nature of NIR spectra, which exhibit diffuse signals and high correlations between features. It is possible that the tabular variants of generative models, such as CTGAN and TVAE, may struggle to fully capture these correlations. This could potentially explain the limitations we observed in terms of data fidelity and utility. Future work could explore alternative generative models that are better suited to capturing these complex, highly correlated features present in NIR spectra.

In addition to NIR spectroscopy, there are other spectroscopic techniques that exhibit sharper peaks and higher spectral resolutions. It is possible that these characteristics might be easier for generative models to capture and reproduce. Future research could investigate the applicability of generative models to other spectroscopic techniques, which may lead to improved fidelity and utility in these domains.

6.4 Future Work

The results of this study illuminate the shortcomings NIR spectroscopic data using state of the art generative models. However, there are still potential areas that warrant further exploration that could possibly enhance the quality and utility of the synthetic data. In this context, we propose two main ways forward for future research: Data Handling and Generative models.

6.4.1 Data Handling

A potential approach to improve the results may involve investigating alternative preprocessing techniques to better align the synthetic data with the real data. In our study, we did not perform hyperparameter tuning for SG smoothing, suggesting that there might be more suitable hyperparameters for the data generated by CTGAN and TVAE. In terms of fidelity, the synthetic data, especially from TVAE, closely resembled the real data. However, the data generated by CTGAN appeared somewhat more irregular. Adjusting the hyperparameters in SG smoothing or employing alternative smoothing techniques, such as moving average, standard normal variate, and multiplicative scatter correction (either individually or in combination), could further enhance the quality of the synthetic data. These methods have been used to preprocess NIR spectra in previous studies [97].

Another approach we could consider is treating the NIR spectra as 1D images, which might yield more desirable outcomes. In our study, we opted to represent NIR spectra samples as rows in a table due to their structured format. This choice made the use of CTGAN and TVAE particularly sensible. However, we observed that the synthetic data tended to be less regular than the real data. A GAN model that interprets NIR spectra as images might be better equipped to account for and maintain the local relationships between neighboring features, which appears to be a primary challenge with our results. This could be achieved through the use of a GAN with CNN, unlike the fully connected networks employed by CTGAN. Additionally, we could explore the possibility of treating a small selection of randomly chosen spectra as a 2D image. This approach could potentially enhance the smoothness of the real data by leveraging multiple spectra to illustrate this characteristic.

Another interesting idea if considering NIR spectra as image data would be to use a CNN to

predict the target. Since the synthetic spectra in many ways visually resemble the real spectra, treating the synthetic spectra as 1D-images could yield better performance with respect to utility, even if the data was produced by a model that considers the data tabular.

It might also be helpful to choose a different subset of the NIR spectra. The wavelengths in the range of approximately 700-750 nm display quite a different intensity from the subsequent wavelengths. This discrepancy could make it more challenging for the generative models to accurately model the spectra and preserve the relationship between them. Removing the first few columns could make a difference.

Binning the target could transform the regression problem into a classification problem, potentially making it easier to preserve the relationship between the target and features. Additionally, predicting bins might be more forgiving of the rougher patterns of the synthetic spectra, thus making it easier to predict the target.

Spectral data, especially NIR spectra, is characterized by a high degree of correlation between features [76]. This presents a unique challenge for synthetic data generation. Our current preprocessing methods and generative models, while effective to an extent, were not fully able to capture these inter-feature correlations. To address this, future work should consider exploring data handling techniques and generative models that can explicitly model these correlations. Methods such as PLSR [19] could be employed in preprocessing to emphasize the structure of correlation in the data. Alternatively, more advanced generative models that explicitly model correlations could be utilized.

Lastly, an alternative approach worth exploring involves creating hybrid datasets by combining real and synthetic data to potentially enhance the predictions of ML models. In this study, we focused on training models exclusively on synthetic data; however, augmenting real data with synthetic data could lead to improved prediction performance. This idea addresses the limitations observed when using purely synthetic data and introduces a new direction for investigating the potential advantages of leveraging hybrid datasets in this domain.

6.4.2 Model Tuning

Despite initial attempts, manual tuning of hyperparameters for the CTGAN and TVAE models yielded negligible improvements in the generated synthetic NIR spectra. As a result, the prospect of systematic hyperparameter tuning, which includes techniques such as grid search, random search, and Bayesian optimization [98], was not pursued due to computational resource constraints. Future research with more computational resources could explore this aspect for potentially enhanced results.

6.4.3 CTAB-GAN

In addition to hyperparameter tuning, the consideration of more advanced GAN models could also be beneficial. In particular, CTAB-GAN is an intriguing candidate for further exploration [99]. CTAB-GAN, unlike CTGAN and TVAE, is specifically tailored to manage skewed continuous distributions, a characteristic seen in our NIR spectra. By employing diverse data preprocessing methods for different distributions, CTAB-GAN provides a versatile tool for handling various spectral shapes.

The structure of CTAB-GAN includes a generator and a discriminator, both of which are CNNs, along with an auxiliary multi-layer perceptron classifier. This additional classifier plays a pivotal role in preserving the semantic richness of the original data and estimating the classes

of synthetic data, a feature that could prove especially advantageous when dealing with the intricate associations found within NIR spectral data. To tackle any imbalances present in the training dataset, CTAB-GAN also integrates a conditional aspect, which serves to optimize the learning experience.

CTAB-GAN employs a training regimen using a cross-entropy loss function supplemented with information loss and classification loss. This multi-objective loss function could lead to a more balanced representation of the real NIR spectra in the synthetic data. In preliminary studies, CTAB-GAN has demonstrated superior performance in terms of ML efficacy and statistical similarity metrics compared to other state-of-the-art models, suggesting its potential to improve the results obtained in our study. The exploration of such advanced models in conjunction with hyperparameter tuning could lead to significant improvements in the generation of synthetic NIR spectra.

6.4.4 Other Generative Models

Generative AI has experienced rapid progress in recent years, driven by advancements in deep learning algorithms, computational power, and the availability of large-scale datasets. This growth has led to the development of sophisticated models capable of generating realistic and high-quality data across various domains, such as images, text, audio, and tabular data. Two types of generative models that have been on the rise recently is the adverserial autoencoders [100] and denoising diffusion probabilistic models [101].

Adversarial Autoencoders (AAEs) could be considered for further work on generating synthetic NIR spectra. As a type of generative model that combines the concepts of autoencoders and GANs, AAEs can potentially improve the generation of synthetic NIR spectra by leveraging their unique architecture and adversarial loss concept. As explained earlier in the text, the latent space in VAE is regularized by minimizing the KL-divergence between the distribution of the encoded samples and a predefined distribution, usually normal. In AAE, the latent space is instead regularized through adversarial loss. By using a discriminator to distinguish between random vectors drawn from a desired distribution (again, usually normal) and an encoded sample, it will through adversarial loss enforce the distribution of the encoded sample to approximate that distribution.

AAEs have shown promising results in generating clearer latent spaces and more consistent representations in other domains. In the context of generating synthetic NIR spectra, AAEs could potentially produce more realistic and accurate spectra by better capturing the underlying structure and distribution of the original data.

Denoising Diffusion Probabilistic Models (DDPMs) offer a promising avenue for generating high-quality synthetic NIR spectra. These models have recently gained traction in the generative modeling domain due to their ability to produce highly realistic synthetic samples by leveraging a diffusion process [102]. DDPMs have outperformed GAN and VAE in various important benchmarks in recent years [103, 104]

The diffusion process is a random process that models how a system changes over time due to random fluctuations or noise. In generative modeling, the diffusion process is used to simplify a sample from a complex data distribution into a simpler distribution, typically Gaussian noise, through a series of time steps that progressively adds a small amount of noise to the data. A neural network then tries to undo this diffusion process by denoising it to restore the data back to its original state. It starts with the last time step noise was introduced to the data and tries to output the less noisy version of the previous time step, repeating this process for every step

that introduced noise to the data. By iteratively performing the denoising process in reverse order of the diffusion process, the neural network learns the intricate distribution of the real data, enabling it to transform randomly sampled noise into meaningful synthetic samples that bears a strong resemblance to real data.

DDPM are becoming a leading paradigm in most domains, and tabular data is no exception; in a study by Kotelnikov et al. [105], the authors introduced TabDDPM, a DDPM specifically designed for tabular data, and compared its performance against leading variants of GAN (CTGAN, CTAB-GAN and CTAB-GAN+) and VAE (TVAE) in terms of quality, ML efficiency, and privacy. TabDDPM demonstrated superior performance in approximating individual feature distributions, particularly for numerical features with a uniform distribution, categorical features with numerous categories, and features that were both continuous and discrete. The ML efficiency of TabDDPM was evaluated using the average performance of various algorithms (Decision Tree, RF, Logistic Regression, and MLP models from the scikit-learn library) and the performance of CatBoost, a highly successful classifier of tabular data. TabDDPM outperformed all other generators in both evaluation methods, showcasing its high utility. When we treat NIR data as tabular, this approach could perhaps lead to better results than we acquired in this study.

Chapter 7

Conclusion

In this thesis, we scrutinized the capability of the Synthetic Data Vault's (SDV) models—CTGAN and TVAE—in synthesizing Near-Infrared (NIR) spectral data that closely mirrors real-life counterparts. The investigation revealed that both models could generate synthetic data with similar statistical properties. However, TVAE emerged as the superior model, demonstrating a higher proficiency in preserving correlations between variables and the relationship between target and features.

Despite the superficially promising nature of the synthetic data, as evidenced by plots and statistical similarities, ML classifiers managed to distinguish real from synthetic data effortlessly. This reveals the persistent challenge of creating synthetic tabular data that can convincingly imitate real data. Furthermore, when assessing utility, synthetic data generated by both CTGAN and TVAE fell short in replacing real data for training ML models aimed at predicting DMC.

In conclusion, the synthetic NIR spectral data crafted by CTGAN and TVAE display potential but also room for substantial enhancement, particularly in terms of fidelity and utility. This research broadens our understanding of synthetic data generation and its role in NIR spectroscopy. The study highlighted TVAE's superior performance in generating more realistic synthetic data, yet underlined that neither model could convincingly supplant real data in ML model training.

The findings of this thesis pave the way for intriguing future research opportunities. These include exploring diverse preprocessing techniques, experimenting with a wider range of generative models, investigating the treatment of NIR spectra as 1D images, and transitioning the regression problem into a classification one. The concept of creating hybrid datasets, which blend real data with synthetic data, also warrants further exploration. Delving into these directions could herald the advent of more adept generative models and amplify the utility of synthetic data in NIR spectroscopy and related domains.

In our pursuit of synthetic data generation techniques that can effectively stand in for real data in training ML models, further research is indispensable. Continued efforts in this area promise to bring about significant benefits across a multitude of applications within the field of NIR spectroscopy.

Bibliography

[1] M Blanco and INIR Villarroya. "NIR spectroscopy: a rapid-response analytical tool". In: *TrAC Trends in Analytical Chemistry* 21.4 (2002), pp. 240–250.

[2] Alexander Ratner et al. "Snorkel: Rapid training data creation with weak supervision". In: *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*. Vol. 11. 3. NIH Public Access. 2017, p. 269.

[3] MIT Data to AI Lab. *SDV: Synthetic Data Vault*. `https://github.com/sdv-dev/SDV`. 2021.

[4] Lei Xu et al. "Modeling tabular data using conditional gan". In: *Advances in Neural Information Processing Systems* 32 (2019).

[5] Anbing Zheng et al. "Identification of Multi-Class Drugs Based on Near Infrared Spectroscopy and Bidirectional Generative Adversarial Networks". In: *Sensors* 21.4 (2021), p. 1088.

[6] Man Wu et al. "Deep learning data augmentation for Raman spectroscopy cancer tissue classification". In: *Scientific reports* 11.1 (2021), p. 23842.

[7] Stefano Di Frischia et al. "Enhanced Data Augmentation using GANs for Raman Spectra Classification". In: *2020 IEEE International Conference on Big Data (Big Data)*. IEEE. 2020, pp. 2891–2898.

[8] Avital Oliver et al. "Realistic evaluation of deep semi-supervised learning algorithms". In: *Advances in neural information processing systems* 31 (2018).

[9] Connor Shorten and Taghi M Khoshgoftaar. "A survey on image data augmentation for deep learning". In: *Journal of big data* 6.1 (2019), pp. 1–48.

[10] Celio Pasquini. "Near infrared spectroscopy: A mature analytical technique with new perspectives–A review". In: *Analytica chimica acta* 1026 (2018), pp. 8–36.

[11] NT Anderson et al. "Achieving robustness across season, location and cultivar for a NIRS model for intact mango fruit dry matter content". In: *Postharvest Biology and Technology* 168 (2020), p. 111202.

[12] Carl R. Noller and Melvyn C. Usselman. *Spectroscopy of organic compounds*. URL: `https://www.britannica.com/science/chemical-compound/Spectroscopy-of-organic-compounds`.

[13] Wilsonee. File:Spectroscopy pic.png. 2017. URL: `https://commons.wikimedia.org/wiki/File:Spectroscopy_pic.png`.

[14] Brian G Osborne. "Near-infrared spectroscopy in food analysis". In: *Encyclopedia of analytical chemistry: applications, theory and instrumentation* (2006).

[15] Donald F Swinehart. "The beer-lambert law". In: *Journal of chemical education* 39.7 (1962), p. 333.

[16] U Thissen et al. “Comparing support vector machines to PLS for spectral regression applications”. In: *Chemometrics and Intelligent Laboratory Systems* 73.2 (2004), pp. 169–179.

[17] Nabil Benoudjit et al. “Chemometric calibration of infrared spectrometers: selection and validation of variables by non-linear models”. In: *Chemometrics and intelligent laboratory systems* 70.1 (2004), pp. 47–53.

[18] Svante Wold, Kim Esbensen, and Paul Geladi. “Principal component analysis”. In: *Chemometrics and intelligent laboratory systems* 2.1-3 (1987), pp. 37–52.

[19] Hervé Abdi. “Partial least square regression (PLS regression)”. In: *Encyclopedia for research methods for the social sciences* 6.4 (2003), pp. 792–795.

[20] LML Laurens and EJ Wolfrum. “High-throughput quantitative biochemical characterization of algal biomass by NIR spectroscopy; multiple linear regression and multivariate linear regression analysis”. In: *Journal of agricultural and food chemistry* 61.50 (2013), pp. 12307–12314.

[21] Mark Tranmer and Mark Elliot. “Multiple linear regression”. In: *The Cathie Marsh Centre for Census and Survey Research (CCSR)* 5.5 (2008), pp. 1–5.

[22] Peter D Wentzell and Lorenzo Vega Montoto. “Comparison of principal components regression and partial least squares regression through generic simulations of complex mixtures”. In: *Chemometrics and intelligent laboratory systems* 65.2 (2003), pp. 257–279.

[23] Jie Yang et al. “Deep learning for vibrational spectral analysis: Recent progress and a practical guide”. In: *Analytica chimica acta* 1081 (2019), pp. 6–17.

[24] Sebastian Raschka. *Python machine learning*. Packt publishing ltd, 2015.

[25] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *nature* 521.7553 (2015), pp. 436–444.

[26] Michael I Jordan and Tom M Mitchell. “Machine learning: Trends, perspectives, and prospects”. In: *Science* 349.6245 (2015), pp. 255–260.

[27] Francois Chollet. *Deep learning with Python*. Simon and Schuster, 2021.

[28] Vladimir Zwass. *Neural network*. `https://www.britannica.com/technology/neural-network`. Accessed: 2023-04-06.

[29] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. “Learning representations by back-propagating errors”. In: *nature* 323.6088 (1986), pp. 533–536.

[30] Neena Aloysius and M Geetha. “A review on deep convolutional neural networks”. In: *2017 international conference on communication and signal processing (ICCSP)*. IEEE. 2017, pp. 0588–0592.

[31] Manjot Kaur and Aakash Mohta. “A review of deep learning with recurrent neural network”. In: *2019 International Conference on Smart Systems and Inventive Technology (ICSSIT)*. IEEE. 2019, pp. 460–465.

[32] M Sadegh Riazi, Bita Darvish Rouani, and Farinaz Koushanfar. “Deep learning on private data”. In: *IEEE Security & Privacy* 17.6 (2019), pp. 54–63.

[33] Davide Castelvecchi. “Can we open the black box of AI?” In: *Nature News* 538.7623 (2016), p. 20.

[34] Xiaoting Zhong et al. “Explainable machine learning in materials science”. In: *npj Computational Materials* 8.1 (2022), p. 204.

[35] Salvador Robles Herrera, Martine Ceberio, and Vladik Kreinovich. “When is deep learning better and when is shallow learning better: qualitative analysis”. In: *International Journal of Parallel, Emergent and Distributed Systems* 37.5 (2022), pp. 589–595.

[36] Olivier Devos et al. “Support vector machines (SVM) in near infrared (NIR) spectroscopy: Focus on parameters optimization and model interpretation”. In: *Chemometrics and Intelligent Laboratory Systems* 96.1 (2009), pp. 27–33.

[37] Sanguk Lee et al. “Random forest as a potential multivariate method for near-infrared (NIR) spectroscopic analysis of complex mixture samples: Gasoline and naphtha”. In: *Microchemical Journal* 110 (2013), pp. 739–748.

[38] Anil K Jain. “Data clustering: 50 years beyond K-means”. In: *Pattern recognition letters* 31.8 (2010), pp. 651–666.

[39] Mingxing Xu et al. “Improving the accuracy of soil organic carbon content prediction based on visible and near-infrared spectroscopy and machine learning”. In: *Environmental Earth Sciences* 80.8 (2021), p. 326.

[40] Abraham Savitzky and Marcel JE Golay. “Smoothing and differentiation of data by simplified least squares procedures.” In: *Analytical chemistry* 36.8 (1964), pp. 1627–1639.

[41] Felipe Bachion de Santana, André Marcelo de Souza, and Ronei Jesus Poppi. “Visible and near infrared spectroscopy coupled to random forest to quantify some soil quality parameters”. In: *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy* 191 (2018), pp. 454–462.

[42] Supei Zhang et al. “Determination of the food dye indigotine in cream by near-infrared spectroscopy technology combined with random forest model”. In: *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy* 227 (2020), p. 117551.

[43] Roman M Balabin, Ravilya Z Safieva, and Ekaterina I Lomakina. “Near-infrared (NIR) spectroscopy for motor oil classification: From discriminant analysis to support vector machines”. In: *Microchemical Journal* 98.1 (2011), pp. 121–128.

[44] Guikun Chen et al. “An efficient tea quality classification algorithm based on near infrared spectroscopy and random Forest”. In: *Journal of Food Process Engineering* 44.1 (2021), e13604.

[45] Roman M Balabin and Ekaterina I Lomakina. “Support vector machine regression (SVR/LS-SVM)—an alternative to neural networks (ANN) for analytical chemistry? Comparison of nonlinear methods on near infrared (NIR) spectroscopy data”. In: *Analyst* 136.8 (2011), pp. 1703–1712.

[46] Julio Cesar L Alves and Ronei J Poppi. “Biodiesel content determination in diesel fuel blends using near infrared (NIR) spectroscopy and support vector machines (SVM)”. In: *Talanta* 104 (2013), pp. 155–161.

[47] Li-Guo Zhang et al. “Rapid identification of adulterated cow milk by non-linear pattern recognition methods based on near infrared spectroscopy”. In: *Food chemistry* 145 (2014), pp. 342–348.

[48] Roman M Balabin, Ravilya Z Safieva, and Ekaterina I Lomakina. “Gasoline classification using near infrared (NIR) spectroscopy data: Comparison of multivariate techniques”. In: *Analytica Chimica Acta* 671.1-2 (2010), pp. 27–35.

[49] Li-Jun Ni et al. “Pattern recognition of Chinese flue-cured tobaccos by an improved and simplified K-nearest neighbors classification algorithm on near infrared spectra”. In: *Analytica chimica acta* 633.1 (2009), pp. 43–50.

[50] Matthieu Lesnoff, Maxime Metz, and Jean-Michel Roger. "Comparison of locally weighted PLS strategies for regression and discrimination on agronomic NIR data". In: *Journal of Chemometrics* 34.5 (2020), e3209.

[51] Aphex34. *Fully connected convolutional neural network*. File:Typical cnn.png. 2015. URL: https://commons.wikimedia.org/wiki/File:Typical_cnn.png.

[52] Nitin Kumar Chauhan and Krishna Singh. "A review on conventional machine learning vs deep learning". In: *2018 International conference on computing, power and communication technologies (GUCON)*. IEEE. 2018, pp. 347–352.

[53] Jiechao Yang et al. "Combination of convolutional neural networks and recurrent neural networks for predicting soil properties using Vis–NIR spectroscopy". In: *Geoderma* 380 (2020), p. 114616.

[54] Ailing Tan et al. "Near infrared spectroscopy quantification based on Bi-LSTM and transfer learning for new scenarios". In: *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy* 283 (2022), p. 121759.

[55] MingxianLin. File:RNN.png. 2018. URL: https://commons.wikimedia.org/wiki/File:RNN.png.

[56] David Foster. *Generative deep learning: teaching machines to paint, write, compose, and play*. O'Reilly Media, 2019.

[57] Tony Jebara. *Machine learning: discriminative and generative*. Vol. 755. Springer Science & Business Media, 2012.

[58] Jiayu Wang et al. "Transformation gan for unsupervised image synthesis and representation learning". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 472–481.

[59] Jordi Esteve Sorribas. File:1 ajkhD8gbCBwVFOjNfEAzw.png. 2023. URL: https://commons.wikimedia.org/wiki/File:1_ajkhD8gbCBwVFOjNfEAzw.png.

[60] Yoshua Bengio, Aaron Courville, and Pascal Vincent. "Representation learning: A review and new perspectives". In: *IEEE transactions on pattern analysis and machine intelligence* 35.8 (2013), pp. 1798–1828.

[61] GM Harshvardhan et al. "A comprehensive survey and analysis of generative models in machine learning". In: *Computer Science Review* 38 (2020), p. 100285.

[62] Ian J. Goodfellow et al. *Generative Adversarial Networks*. 2014. arXiv: 1406.2661 [stat.ML].

[63] Diederik P Kingma and Max Welling. *Auto-Encoding Variational Bayes*. 2022. arXiv: 1312.6114 [stat.ML].

[64] MA Clyde et al. "Current challenges in Bayesian model choice". In: *Statistical challenges in modern astronomy IV*. Vol. 371. 2007, p. 224.

[65] Ruslan Salakhutdinov. "Learning deep generative models". In: *Annual Review of Statistics and Its Application* 2 (2015), pp. 361–385.

[66] Tero Karras et al. "Training generative adversarial networks with limited data". In: *Advances in neural information processing systems* 33 (2020), pp. 12104–12114.

[67] Hamed Alqahtani, Manolya Kavakli-Thorne, and Gulshan Kumar. "Applications of generative adversarial networks (gans): An updated review". In: *Archives of Computational Methods in Engineering* 28 (2021), pp. 525–552.

[68] Eitan Richardson and Yair Weiss. "On gans and gmms". In: *Advances in Neural Information Processing Systems* 31 (2018).

[69] Naveen Kodali et al. “On convergence and stability of gans”. In: *arXiv preprint arXiv:1705.07215* (2017).

[70] Martin Arjovsky and Léon Bottou. “Towards principled methods for training generative adversarial networks”. In: *arXiv preprint arXiv:1701.04862* (2017).

[71] Dor Bank, Noam Koenigstein, and Raja Giryes. “Autoencoders”. In: *arXiv preprint arXiv:2003.05991* (2020).

[72] EugenioTL. *Variational Autoencoder structure*. File:VAE Basic.png. 2021. URL: `https://commons.wikimedia.org/wiki/File:VAE_Basic.png`.

[73] Unai Garay-Maestre, Antonio-Javier Gallego, and Jorge Calvo-Zaragoza. “Data augmentation via variational auto-encoders”. In: *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications: 23rd Iberoamerican Congress, CIARP 2018, Madrid, Spain, November 19-22, 2018, Proceedings 23*. Springer. 2019, pp. 29–37.

[74] Mehdi Mirza and Simon Osindero. “Conditional generative adversarial nets”. In: *arXiv preprint arXiv:1411.1784* (2014).

[75] Gael Lederrey, Tim Hillel, and Michel Bierlaire. “DATGAN: Integrating expert knowledge into deep learning for synthetic tabular data”. In: *arXiv preprint arXiv:2203.03489* (2022).

[76] PW Goedhart. “Comparison of multivariate calibration methods for prediction of feeding value by near infrared reflectance spectroscopy.” In: *Netherlands Journal of Agricultural Science* 38.3B (1990), pp. 449–460.

[77] Tomoyuki Nagasawa et al. “fNIRS-GANs: data augmentation using generative adversarial networks for classifying motor tasks from functional near-infrared spectroscopy”. In: *Journal of Neural Engineering* 17.1 (2020), p. 016068.

[78] Dehua Zhu et al. “Synthetic spectra generated by boundary equilibrium generative adversarial networks and their applications with consensus algorithms”. In: *Optics Express* 28.12 (2020), pp. 17196–17208.

[79] Kaixun He, Jingjing Liu, and Zhi Li. “Application of Generative Adversarial Network for the Prediction of Gasoline Properties”. In: *Chemical Engineering Transactions* 81 (2020), pp. 907–912.

[80] James Jordon et al. *Synthetic Data – what, why and how?* 2022. arXiv: `2205.03257 [cs.LG]`.

[81] Mikel Hernandez et al. “Standardised metrics and methods for synthetic tabular data evaluation”. In: *Preprint at https://doi. org/10.36227/techrxiv* 16610896 (2021), p. v1.

[82] Richard Taylor. “Interpretation of the correlation coefficient: a basic review”. In: *Journal of diagnostic medical sonography* 6.1 (1990), pp. 35–39.

[83] Vance W Berger and YanYan Zhou. “Kolmogorov–smirnov test: Overview”. In: *Wiley statsref: Statistics reference online* (2014).

[84] Edward H Livingston. “Who was student and why do we care so much about his t-test? 1”. In: *Journal of Surgical Research* 118.1 (2004), pp. 58–65.

[85] Nadim Nachar et al. “The Mann-Whitney U: A test for assessing whether two independent samples come from the same distribution”. In: *Tutorials in quantitative Methods for Psychology* 4.1 (2008), pp. 13–20.

[86] Joshua B Tenenbaum, Vin de Silva, and John C Langford. “A global geometric framework for nonlinear dimensionality reduction”. In: *science* 290.5500 (2000), pp. 2319–2323.

[87] EL Hacen Zein and Tanguy Urvoy. “Tabular Data Generation: Can We Fool XGBoost?” In: *NeurIPS 2022 First Table Representation Workshop*.

[88] Kei Long Wong et al. “A Novel Fusion Approach Consisting of GAN and State-of-Charge Estimator for Synthetic Battery Operation Data Generation”. In: *Electronics* 12.3 (2023), p. 657.

[89] Michael Platzer and Thomas Reutterer. “Holdout-based empirical assessment of mixed-type synthetic data”. In: *Frontiers in big Data* 4 (2021), p. 679939.

[90] Lei Xu et al. “Modeling Tabular data using Conditional GAN”. In: *CoRR* abs/1907.00503 (2019). arXiv: 1907.00503. URL: http://arxiv.org/abs/1907.00503.

[91] Puneet Mishra and Dário Passos. “Deep multiblock predictive modelling using parallel input convolutional neural networks”. In: *Analytica chimica acta* 1163 (2021), p. 338520.

[92] John W Palmer et al. “Fruit dry matter concentration: a new quality metric for apples”. In: *Journal of the Science of Food and Agriculture* 90.15 (2010), pp. 2586–2594.

[93] Nicholas T Anderson, Phul P Subedi, and Kerry B Walsh. “Manipulation of mango fruit dry matter content to improve eating quality”. In: *Scientia Horticulturae* 226 (2017), pp. 316–321.

[94] Hailong Wang et al. “Fruit quality evaluation using spectroscopy technology: a review”. In: *Sensors* 15.5 (2015), pp. 11889–11927.

[95] Md Asadur Rahman, Mohd Abdur Rashid, and Mohiuddin Ahmad. “Selecting the optimal conditions of Savitzky–Golay filter for fNIRS signal”. In: *Biocybernetics and Biomedical Engineering* 39.3 (2019), pp. 624–637.

[96] Pasquale Zingo and Andrew Novocin. “Introducing the TSTR Metric to Improve Network Traffic GANs”. In: *Advances in Information and Communication: Proceedings of the 2021 Future of Information and Communication Conference (FICC), Volume 1*. Springer. 2021, pp. 643–650.

[97] Bahareh Jamshidi et al. “Reflectance Vis/NIR spectroscopy for nondestructive taste characterization of Valencia oranges”. In: *Computers and Electronics in Agriculture* 85 (2012), pp. 64–69.

[98] Li Yang and Abdallah Shami. “On hyperparameter optimization of machine learning algorithms: Theory and practice”. In: *Neurocomputing* 415 (2020), pp. 295–316.

[99] Zilong Zhao et al. *CTAB-GAN: Effective Table Data Synthesizing*. 2021. arXiv: 2102.08369 [cs.LG].

[100] Alireza Makhzani et al. “Adversarial autoencoders”. In: *arXiv preprint arXiv:1511.05644* (2015).

[101] Jascha Sohl-Dickstein et al. “Deep unsupervised learning using nonequilibrium thermodynamics”. In: *International Conference on Machine Learning*. PMLR. 2015, pp. 2256–2265.

[102] Jonathan Ho, Ajay Jain, and Pieter Abbeel. “Denoising diffusion probabilistic models”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 6840–6851.

[103] Gustav Müller-Franzes et al. “Diffusion Probabilistic Models beat GANs on Medical Images”. In: *arXiv preprint arXiv:2212.07501* (2022).

[104] Prafulla Dhariwal and Alex Nichol. *Diffusion Models Beat GANs on Image Synthesis*. 2021. arXiv: 2105.05233 [cs.LG].

[105] Akim Kotelnikov et al. “TabDDPM: Modelling Tabular Data with Diffusion Models”. In: *arXiv preprint arXiv:2209.15421* (2022).

Norges miljø- og biovitenskapelige universitet
Noregs miljø- og biovitskapelege universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway