Uncovering Behavioural Patterns in Multivariate Time Series through Latent Space Analysis

Utrecht University

Beatriz Zamith, 2843927

Faculty of Science,
at Utrecht University

Supervisors:
Dr. M. Behrisch
Prof. dr. ir. A.C. Telea

External Supervisor:
S. Pelka

A thesis submitted for the degree of Master of Science
(MSc.) in Artificial Intelligence

July 14, 2023

Acknowledgements

I would like to show my sincere appreciation to my first supervisor, Michael Behrisch, and my external supervisor, Sabine Pelka, for their continuous support, guidance, and valuable insights that have been instrumental throughout the entire process.

I also want to express my appreciation to my second supervisor, Alex Telea, for his valuable feedback and examination of my project.

Finally, I would like to express my gratitude to my family, friends and boyfriend who have shown me unwavering love and support throughout this journey.

Abstract

Multivariate time series (MTS) data, consisting of multiple time series observed concurrently across multiple variables, has become increasingly common in a variety of fields ranging from finance to healthcare. However, due to its complex nature, analysing and extracting meaningful insights from it remains a challenging task. This project aims to address the challenge of extracting valuable insights from high-complexity data by combining machine learning with visualisation tools, with a particular focus on latent space analysis. The framework is applied to a general energy sustainability project to understand how behavioural interventions affect energy consumption. We evaluate desirable features of the framework by conducting testing and comparing the model's performance using various combinations of parameters. Additionally, we apply the framework to real use case scenarios, demonstrating its practical applicability and effectiveness. In conclusion, our findings indicate that the framework effectively facilitates the exploration of MTS data. Latent space analysis proves valuable in providing deeper insights into the data, allowing us to investigate the effects of sociodemographic factors, detect anomalies, and uncover behavioural patterns.

Contents

List of Figures

List of Tables

1 Introduction

The electricity and heat production sector, used to regulate residential energy usage, is the largest single source of global greenhouse gas emissions, amounting to a total of 25% of all global emissions, and consequently a meaningful contributor to climate change [23]. One of the current solutions to this urgent problem is the use of smart buildings and grids, which optimize and reduce energy consumption and encourage the primary use of renewable energy sources [42]. This shift in energy production is a much-needed contribution to addressing climate change. However, traditional approaches made in an effort to persuade end users to adopt energy-efficient behaviour are often characterized by poor outcomes [47].

Recent solutions aim to address the issue of previous failed smart grid interventions by grounding its treatments in fundamental principles of behavioural science [19], using behavioural interventions. However, the nature of such projects often implies the extensive collection of data across different types of devices, sectors, and backgrounds, encompassing a vast array of data types and sources. Furthermore, smart meter data enables the collection of highly detailed measurements over time, resulting in the generation of Multivariate Time Series (MTS) data. This type of data is recognized for its high complexity, both in size and nature, making it a challenging task to analyse it and gather insights from it. This is a common problem for researchers dealing with this type of data and occurs, among other reasons, due to a large number of variables and features present which often result in high data dimensionality, making it difficult to understand the relationships between these variables and effectively communicating insights to stakeholders.

Therefore, the intended impact of this project is to provide a solution to the challenge of extracting insights from multivariate time series data by combining machine learning and visualisation tools. While this solution will be implemented on a specific sustainability research project it can and is meant to be extended to other high-complexity data sets. Furthermore, this project also aims to understand whether and in what way behavioural intervention can successfully influence consumers and lead to more sustainable energy consumption. Given that work has been done in this domain, existing tools will be used to explore and tackle the data, and a combination of deep learning and data visualisation techniques will be applied, with a focus on latent space analysis.

The thesis starts by providing background information on Chapter 2. This will introduce Multivariate Time Series (MTS) and their challenges to the reader, as well as the most common approaches, namely in the field of traditional statistics and machine learning, which will facilitate the understanding of the thesis. Furthermore, related MTS visualization techniques will be explored. In Chapter 3 the tasks of the project will be described fol-

lowed by the overall pipeline. Then, Chapter 4 will detail the implemented model and visualization techniques and introduce a pattern taxonomy guide. In Chapter 5 the design of the Visualisation tool interface is detailed and explored. Chapter 6 details the analysis and assessment of the project's model and examines real-world applications and outcomes. Finally, Chapter 7 answers the research questions and addresses limitations suggesting future research directions.

1.1 Research Objectives

The overall aim of this project is to provide a useful and efficient framework for analysing and processing MTS data in the context of behavioural interventions. In particular, to be able to determine and visualize the impact of such interventions on participants' energy consumption. This framework will be applied specifically to an energy sustainability project, given that it is a difficult use case involving a complex MTS dataset with extensive temporal and spatial resolutions. This leads us to the following research questions:

> *RQ1. Which interventions are most successful in positively influencing a consumer towards efficient energy consumption?*
>
> *RQ2. What effects do social-demographic and technical factors have on the effect of interventions on participants' consumption?*

Given the high complexity and dimensionality of the data at hand, the focus of the approach will lie on latent space exploration, which is able to reduce and project the data onto a lower-dimensional space, allowing for easier visualization and exploration of the data, without any loss of information. This leads to the following research questions:

> *RQ3. Which novel insights can be gained from analyzing the latent space representation of the sustainability research data that can not be obtained from the input model representation?*
>
> *RQ4. Can we distinguish a signal from noise in the extensive input space of the sustainability research project?*

Finally, this research will inquire into whether latent space analysis can uncover patterns in data that shed light on how participant behaviour changes when exposed to events, and whether such changes can be understood and interpreted. This leads to the following research question:

> *RQ5. Can latent space analysis reveal behaviour patterns?*

1.2 Contribution

This thesis aims to achieve the following contributions:

- Propose a data analysis methodology to effectively address the limitations of MTS data, unlocking its full potential and enabling comprehensive exploration of patterns in the data.
- Evaluate the effectiveness and applicability of latent space models as a tool for revealing behavioural patterns in data related to events, particularly in the energy consumption sector.
- Provide valuable insights to policymakers, informing the design of targeted interventions and policies that promote energy conservation, incentivize sustainable behaviour, and contribute to a more sustainable energy future.
- Establish a solid foundation for potential future predictive models in energy consumption, allowing improved forecasting and planning capabilities to optimize energy management.

2 Background & Related Work

The purpose of this chapter is to introduce Multivariate Time Series (MTS) and its associated challenges, as well as common approaches used in the field. Additionally, we will discuss related work on visualization techniques applied to Time Series (TS) data. By providing this overview, we aim to establish a foundation for addressing the research question at hand.

2.1 Multivariate Time Series

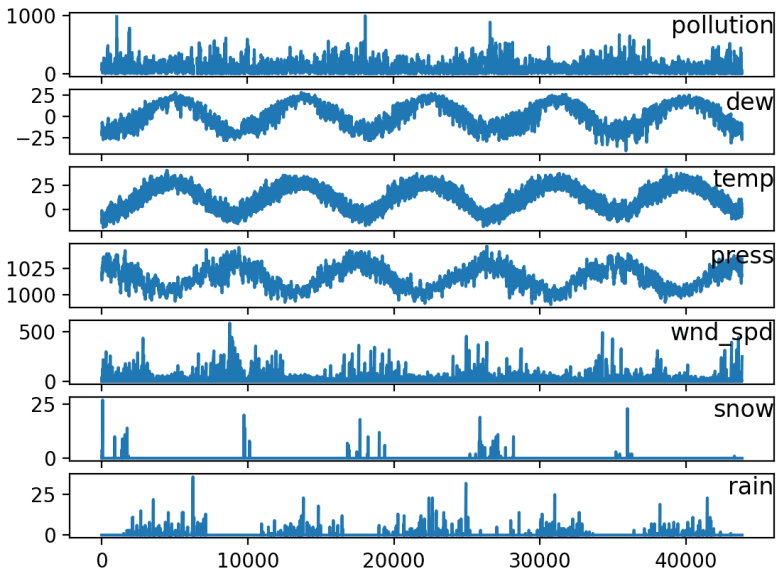


Figure 1: Beijing PM2.5 dataset. Reprinted from Brownlee, 2017 [33]

Multivariate series data (MTS), as illustrated in Figure 1, consists of multiple time series observed concurrently across multiple variables and has become increasingly common in a variety of fields ranging from finance to healthcare [38, 45]. As the name implies, MTS data consists of two or more interconnected variables (or dimensions) that vary over time. For example, the Beijing Pollution dataset [9], partially displayed in Figure 1, includes a number of environmental variables, such as pollution concentration, temperature, dew point, pressure, snow depth, wind speed, and many others, all of which are tracked over time at regular intervals. Each element in this situation can be thought of as a dimension, and analysis will show that these variables are interconnected. A few patterns can be easily detected by visually analysing the data; these refer to a recognisable and repeated

arrangement or sequence of events or values. For example, a clear pattern emerges in the temperature feature, showing a steady increase throughout the day, reaching a high peak and gradually decreasing to a lower peak. A similar pattern can be observed in the dew point variable, while the pressure variable demonstrates an inverse relationship. Another pattern can be seen in the snow depth variable, where the value is zero for most of the day and reaches a higher peak at regular intervals, which coincide with higher temperate peaks. These relationships provide valuable insights for weather and pollution forecasting, as the interactions between these features over time can offer predictive information about atmospheric conditions and pollutant levels. Therefore, the need to detect and understand patterns becomes a crucial requirement for dealing with MTS data and comes from the fact that they hold information on how features relate to each other and offer predictive insights. Due to its information-dense nature, MTS has proven itself advantageous for an array of tasks, such as forecasting, anomaly detection and classification [8, 27, 59].

However, due to its complex nature and numerous inherent difficulties, interpreting the data and extracting insights and value from it is often considered a challenging task. First, and most noticeably, due to the presence of a large number of variables and features that often results in a high data dimensionality, which hinders analysis [8]. This difficulty is further amplified by the non-stationarity of the data, which makes identifying underlying patterns and relationships between variables over time difficult [8, 55]. Ultimately, effectively visualizing and communicating insights to stakeholders ends up being just as challenging as the relationships between variables may shift over time and patterns observed in the data may be nested and overlapping [28].

Although TS and MTS analysis was conducted predominantly using classical statistical analysis techniques [11], in recent years, several new approaches have been proposed to address the challenges of MTS data [4, 14, 35, 36, 51]. Some of the most popular methods for analysing and simplifying MTS data are based on dimensionality reduction techniques. These strategies seek to minimise data dimensionality while maintaining the most significant information and patterns in the data, making it easier to detect patterns and relationships amongst variables, and allowing for a smoother analysis and interpretation of the data. On the statistic side, techniques such as Principal Component Analysis (PCA) [4, 35, 36] are employed. In recent years, machine learning has gained significant attention in the analysis of MTS data. One notable approach that has emerged is the usage of autoencoder models [31, 39, 49, 54]. Chapters 2.2.1 and 2.3.1, explain these concepts further.

2.2 Statistical Approach to MTS

In the field of statistics, a number of well-known methods have been frequently used for MTS analysis because of their capacity to identify relationships and simulate temporal dynamics [44]. However, these statistical approaches come with some limitations. Traditional statistical approaches presuppose linearity and stationarity of data [2, 43], which may not be true in real-world MTS circumstances. Furthermore, statistical models may also have trouble capturing nonlinear interactions between variables and frequently need assumptions about the underlying distribution [21]. The most common statistical approach to MTS data is the use of projection algorithms, there are numerous techniques available, one such example being PCA, which is detailed in Chapter 2.2.1.

2.2.1 Principal component analysis (PCA)

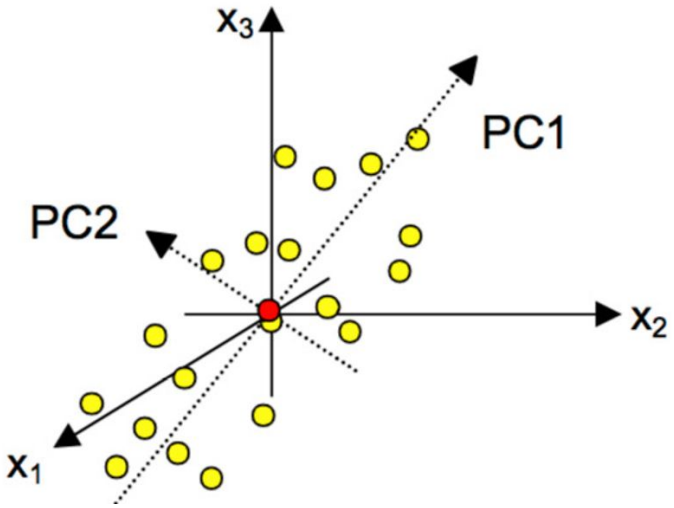


Figure 2: Principal Component Analysis (PCA). Reprinted from *Sartorius*, 2020 [1]

An illustration of a PCA projection can be found in Figure 2. PCA is a projection algorithm commonly used to analyse datasets with a large number of dimensions/features [35]. It is able to explore and identify patterns in data by examining the correlations between its variables. PCA solves the dimensionality challenge of MTS data by projecting the original data onto a smaller dimension, containing a new set of variables known as principal components, which capture the majority of the variance in the original data [35, 36, 51], these can be identified in Figure 2 as *PC1* and *PC2*. Despite being computationally inexpensive when compared to other projection algorithms, its main limitation lies in the fact that it is a linear projection,

which as seen in Chapter 2.1 may not be the most adequate method for dealing with MTS data.

2.3 Machine Learning Approach to MTS

Machine learning has emerged as a valuable tool for analysing MTS data due to its ability to detect detailed patterns and provide correct predictions [6]. These techniques provide flexible and data-driven approaches for capturing the interdependencies, temporal dynamics, and non-linear connections seen in MTS data [18, 50, 52]. By using machine learning approaches, researchers are given the ability to dive into and make use of the inherent complexity of MTS data, allowing advanced analysis, precise prediction, and well-informed decision-making across various fields. In addition to traditional machine learning techniques, one popular approach for MTS analysis is the use of autoencoders, which will be further discussed in Chapter 2.3.1.

2.3.1 Autoencoder Models

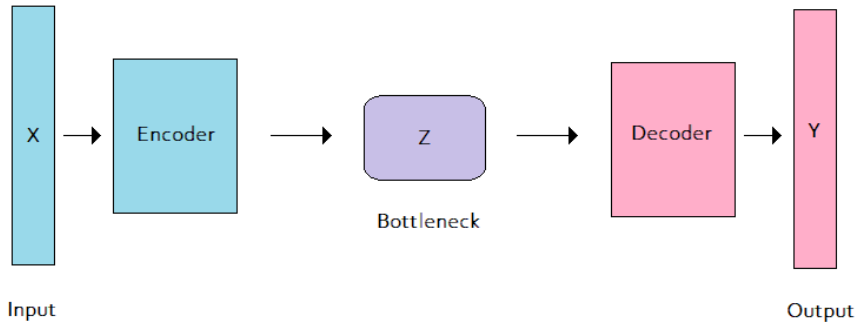


Figure 3: Structure of an Autoencoder model

Autoencoder models are another popular and efficient method for overcoming the difficulties presented by MTS data [31, 39, 46, 49, 54]. An autoencoder is a type of artificial network used in unsupervised learning and consists of two functions: An encoder function, which maps input data to a hidden layer, known as the bottleneck, and a decoder function, which maps the encoded representation to the output layer, recreating the input data. Therefore, autoencoders can learn concise representations of input data without supervision [13]. The model structure can be visualised in Figure 3.

This type of artificial neural network has considerable benefits in the context of MTS data analysis due to its ability to learn a condensed representation of the data in a lower-dimensional space that captures the key

characteristics and patterns of the data [29]. This is rather helpful for feature extraction and determining the key variables or combinations of factors influencing the time series behaviour. It is also able to handle missing or irregularly sampled data and is resistant to noise and variations in the data [5]. Furthermore, autoencoders possess a significant advantage over PCA in terms of being non-linear techniques, despite sharing a common goal of creating projections. While PCA is a linear technique, autoencoders can capture and model complex nonlinear relationships, making them more flexible and powerful in analyzing MTS data.

2.4 MTS Visualization

As evident in Chapter 2.3, machine learning techniques have been successful in simplifying complex data, however, its interpretation remains a challenge. In a research domain simplifying data is undoubtedly beneficial, but allowing this simplified information to be explored and interacted with becomes a vital step in understanding it.

In recent years, MTS visualisation has become increasingly popular due to its broad applicability in a range of fields [57]. Besides being fundamental in identifying relationships, patterns, and trends among variables, as well as correlations, dependencies, and anomalies in the data. It also facilitates exploratory data analysis by allowing users to interact with the data and explore various relationships and patterns. Furthermore, MTS visualisation allows data to be communicated to stakeholders who may not have a technical background, such as policymakers, resulting in more informed decision making.

Timecurves, by Bach et al. [3], is a recent and novel approach to visualising temporal data. *Timecurves* explores the temporal dynamics of a dataset by focusing on similarity information between individual data points. Specifically, it employs a novel folding operation that brings together similar time points in a timeline visualisation, thus enabling the detection of informative temporal patterns. Additionally, *Timecurves* is not designed for domain-specific datasets [3]. The technique has proven to be a particularly useful tool for simplifying highly dimensional datasets by reducing them to a planar curve. Furthermore, it stands out for the way it presents information in a visually intuitive manner that allows users to perceive the connections between data points. However, in addition to its scalability problems, it may fail in its ability to convey all relevant information. Users might not fully understand the data and its relationships as a result. Additionally, since it does not display numerous features at once, *Timecurves* is by nature not ideal for multidimensional temporal data [3].

While general graph exploration techniques such as line graphs [56], heat maps [58], bar plots [24], parallel coordinates [22] and horizon graphs [12] have long been used for time series visualization, domain-specific designs

have recently become an appealing alternative. The driving factor behind this can be attributed to the fact that these designs are customised for the particular characteristics of the domain under research, which provides further sophisticated visualisations of MTS data.

Regarding domain-specific visualization tools, it is noteworthy to mention a few examples in the literature, such as *Peax* and *PSEUDo* by Lekschas et al. [37] and Yu et al. [60], respectively. These techniques employ interactive pattern searches in time series data, with the goal of assisting users in finding interesting patterns in vast volumes of sequential data by using a user-friendly interface for searching, displaying, and analyzing patterns. Both tools use unsupervised deep representation learning to learn meaningful representations of the input data, and allow users to comment on relevancy to improve their search results. Additionally, they also employ sophisticated search methods to raise the effectiveness and precision of pattern searches. PEAX uses a deep autoencoder to learn informative representations of the data and locality-sensitive hashing to index the time series data and speed up the search process [37]. On the other hand, PSEUDo combines convolutional neural networks and long-term memory networks to learn both local and global patterns in the data and uses an index-based search algorithm that permits quick searches across vast volumes of data [60].

Two other relevant tools for domain-specific MTS data are *Compass* and *SeqCausal*, by Deng et al. [16] and Jin et al. [34], respectively. While these tools are specifically aimed at causal analysis, their visual analytics approach gives users several insights into the data and interactions between them, with a focus on visual and interactive components. These tools make use of user-friendly interfaces, which greatly allow data to be explored in dynamic and engaging ways, allowing users to spot patterns and trends with ease.

The idea behind this project is to develop a tool that strives to attain user-friendliness and interactivity, such as the projects mentioned. In our case, the tool will be tailored to latent space representations of the data.

2.4.1 Latent Space Exploration

As already mentioned in Chapter 2.3.1, autoencoders and projection algorithms are recognized for their ability to tackle common and complex data problems and are able to yield impressive outcomes in a multitude of areas. The fundamental idea of an autoencoder goes through the process of uncovering a latent space, consisting of the encoded representation of the high-dimensional data. This space's compressed nature allows the data to be analysed with significantly more efficiency [48].

The latent space allows observations to be mapped and clustered together based on how similar they are. An illustration of the latent space representation of the MNIST dataset [15] can be found in Figure 4, where each data point is represented by the image it represents. Once the data

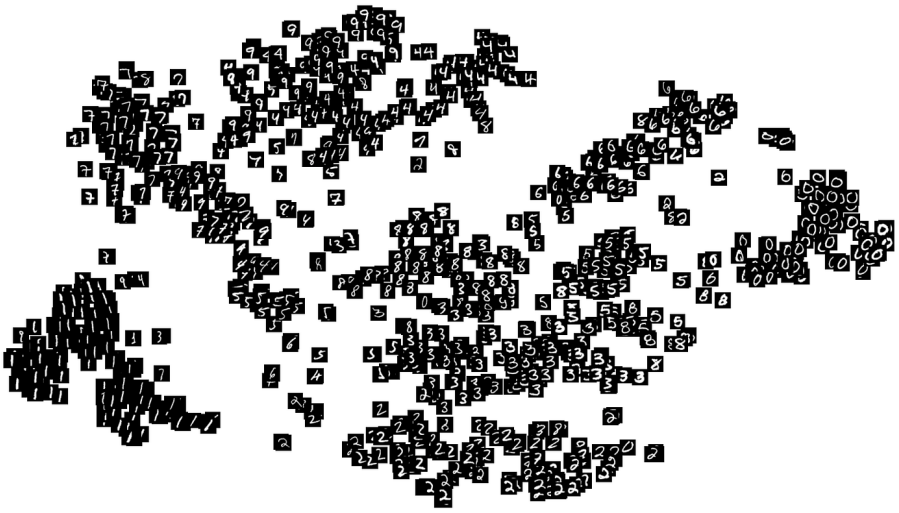


Figure 4: Latent Space of MNIST dataset. Reprinted from Depois, 2017 [17]

is projected onto the latent space, these similarities — which are often not noticeable when the data is presented in its high-dimensional form — become apparent. This has rendered latent space exploration a powerful tool for a variety of tasks. Ranging from broad scenarios such as classification tasks [20], and anomaly detection [26], to more specific applications such as drug discovery [10], speech recognition [30], image recognition [40], object detection [25], and biosignal detection [53].

One limitation of this technique lies within interpretability, since when reducing the dimensionality of the original data, what each encoded dimension represents in the latent space might not be so obvious. Furthermore, despite its success in various fields, limited research has been conducted to uncover behavioural patterns within latent space representations, in particular, in the context of energy consumption.

3 Data Analysis Approach

As observed in the previous Chapter, latent space analysis has been successful in simplifying MTS data. The aim of this project is to explore the usability of latent space exploration for uncovering behavioural patterns, particularly in the scope of energy sustainability research, which has not been done before. Furthermore, we aim to add a visually interactive component to the project to allow for the exploration and understanding of the data. In this chapter, we provide a comprehensive overview of the project's concept. We begin by outlining the project's functional tasks and following that, we provide a detailed breakdown of the project pipeline.

3.1 Dataset

This Chapter aims to provide an understanding of the dataset used in terms of terminology, interventions, pilot trials, and the types of data collected. The dataset used in this study belongs to a sustainability research project that aims to test the effectiveness of behavioural interventions in the context of household energy consumption. The project conducted five pilot trials in different European countries [19]. However, this study focuses specifically on two pilots: one that took place in German households, referred to as the German Pilot, and another in Croatian households referred to as the Croatian Pilot.

The interventions also referred to as *Nudges*, were introduced to participants in three sequential waves by means of a web portal, referred to as the 1st Intervention, 2nd Intervention, and 3rd Intervention. Each pilot and wave featured different interventions. The intervention phases were categorized as Pre-Intervention, Intervention, and Post-intervention.

The dataset comprised environmental, sensor, demographic, and technical data from participants in various households. Demographic variables included age, family type, children count, square metres of living space, house type, special devices, days per week spent at home, and usage of the web portal where interventions take place.

In the German pilot, the sensor data included Self Consumption, Overall Consumption, and Autarky rate, while in the Croatian pilot, it included Consumption and Production. Additionally, environmental data such as time, radiation, and temperature were also included in the dataset. More information about the sustainability project, the pilots, interventions and collected data can be found in the work by Anagnostopoulos et al. [19].

3.2 Task Analysis

Considering the context of the project, this tool is specifically designed for researchers who work with MTS data and seek to derive valuable insights

from it. Therefore, all of these tasks were developed with the consideration that users may not have prior knowledge of machine learning or visualisation techniques. However, it was assumed that users would have a good understanding of the domain being explored. In order to answer the research questions and identify the important crucial requirements for obtaining such targeted outcomes, the following tasks have been identified:

T1. Get an overview of the data

T2. Allow differentiating on different facets (pilot, demographics, etc)

T3. Model and visualize user behavior in order to track user behaviour changes over time

T4. Discern working from non-working nudges

T5. Explore contributing factors for nudge "performance"

behaviour

T6. Compare pilots to find similarities vs dissimilarities

3.3 Pipeline

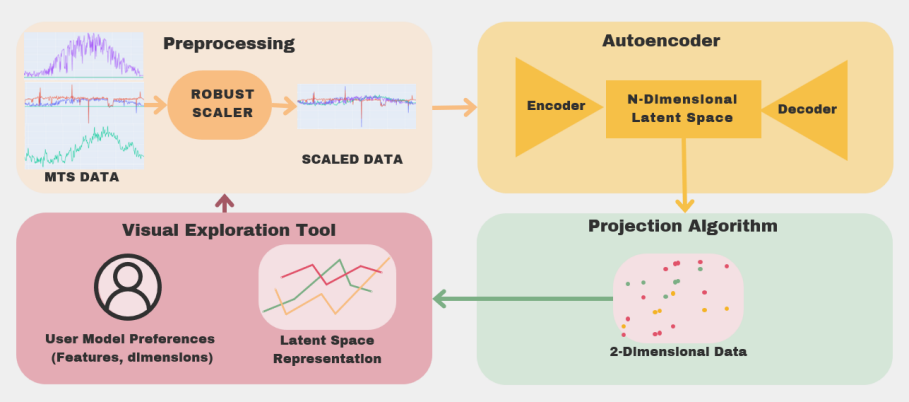


Figure 5: Workflow Diagram of the Proposed Pipeline

In order to complete the tasks detailed in Chapter 3.2 and ultimately to answer the research objectives detailed in Chapter 1.1, a four-step process will be employed. This framework is depicted in Figure 5, in the form of a flowchart. This project has two main focuses, the first one, consisting of the *Preprocessing*, *Autoencoder* and *Projection Algorithm* phases of

the flowchart, regard the processing and exploration of the data from a machine learning point of view, while the remaining phase regards the visual exploration of the data.

As the Figure suggests, this pipeline is circular. This highlights the interactive nature of the process and the fact that users have the ability to interact with a visual exploration tool, described in detail in Sections 4.3 and 5, to make decisions that directly impact the model. These decisions include selecting the pilot to be analysed, determining the features to be included in the model (as outlined in Section 4.4.1), and specifying the number of latent dimensions for the autoencoder.

The second step consists of preprocessing the data and scaling it to similar values. Given that MTS data by nature encompass data of various dimensions and values, it is important to ensure that it is scaled to the same range. As a result, the whole dataset taken from the user-selected pilot is scaled to ensure consistency within the same range. This scaling procedure only takes into account the features that the user has selected in the visual exploration tool.

As mentioned in Chapter 2, when dealing with MTS data, several techniques are available to process and simplify it, with projection algorithms and autoencoder models being notable options. While straightforward dimensionality reduction techniques such as PCA can be directly applied to project the data onto a lower-dimensional space, the use of autoencoders offers additional advantages. Autoencoders stand out for their ability to learn complex non-linear representations, enabling them to capture intricate patterns that may be overlooked by simple projection algorithms. Moreover, incorporating an intermediate step before projection, where the latent dimensions of the model can be expanded and explored, might bring further benefits such as more expressive feature spaces and increased interpretability. Despite typical ranges for latent dimensions ranging from 4 to 512 dimensions [41], for exploratory reasons, we do not impose any limitations here. With that said, the third step consists of encoding the scaled data into an N-Dimensional Latent Space by means of an autoencoder. Once the autoencoder captures the most relevant features of the data and encodes it, a projection algorithm can be used to reduce the dimensionality of the information so it can be plotted and visualised. This is a fundamental step in enhancing the interpretation of the data. Therefore, the fourth step consists of applying a projection algorithm to project the N-Dimensional Latent Space into a 2-Dimensional Space that can be plotted and interpreted.

While it may initially appear redundant to use both an autoencoder and a dimensionality reduction algorithm, their combination aims to strike a balance between capturing complex relationships and achieving interpretability. The autoencoder stands out for capturing subtle relationships in the data by making use of its capacity to reveal complex patterns. On the other hand, the dimensionality reduction process helps to make the data more

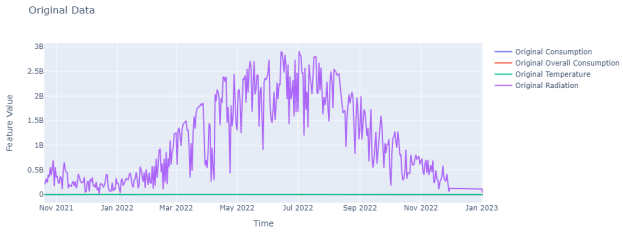
understandable overall.

After the data has been processed and simplified, it becomes essential to visualise it. Taking into account the exploratory context of the project, it is important to provide users with an interactive means to explore and analyse the data. For this reason, the final step of the pipeline consists of leading this data into the visual exploration tool, where all the exploration and visualisation take place.

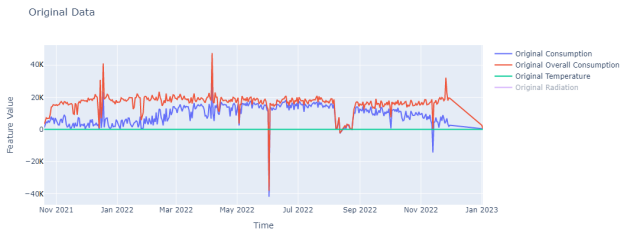
4 Implementation

In this chapter, we will delve into the techniques utilized in the model, encompassing data preprocessing and the implementation of the autoencoder model. Additionally, we will outline the implementation of the visual aspect of the model. Furthermore, we conclude the chapter by presenting an assessment of the project's ethical and privacy implications.

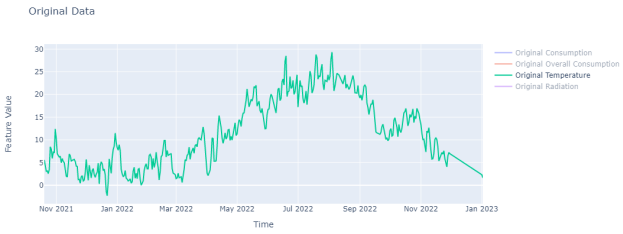
4.1 Preprocessing



(a) Unscaled Model Features



(b) Zoomed-In View of Unscaled Model Features (A)



(c) Zoomed-In View of Unscaled Model Features B

Figure 6: Visual Representation of Unscaled Features

Scaling the data into the same range becomes crucial when dealing with multidimensional data, with each dimension having different metrics and scales. As we can see in Figure 6, such is our case, for example, radiation has much larger values than consumption metrics, which in turn have much larger consumption values than temperature. By scaling the input into the

same range, we make sure the model can handle the data efficiently and compare features in a meaningful way. Additionally, it makes it easier for the model to be trained because it prevents any one feature from dominating the learning process and enables the model to recognise significant patterns across the scaled dimensions.

With that said, the goal of the normalizing step of this pipeline is to scale the data into a more comparable range while preserving its original structure. Furthermore, given the existence of outliers, it is important that the model is robust enough to not let them distort the data. For this purpose, 3 techniques were considered: Standard Scaling, Min-Max Scaling, and Robust Scaling.

The first scaling technique, Standard Scaling, is defined as $z = \frac{x-\mu}{\sigma}$ and consists of rescaling features such that they have zero mean and unit variance. This means that each feature scaled by this technique has a mean of 0 and a standard deviation of 1. Although it is easy to implement, this technique is sensitive to outliers and assumes that the features are normally distributed.

The Min-Max Scaler, defined as $z = \frac{x-x_{min}}{x_{max}-x_{min}}$, consists of rescaling the features to a specific range, usually between 0 and 1, such that the minimum value of each feature is 0 and the maximum value is 1. This technique is advantageous due to its ability to scale features to a specific range. However, it is important to note that this scaler is more sensitive to outliers compared to the previous technique since the presence of an outlier can significantly alter the minimum and maximum values used for scaling, thus affecting the entire process.

Lastly, the Robust Scaler, defined as $z = \frac{x- \, median}{IQR}$, removes the median to each feature and scales the data according to the IQR (difference between the 1st quartile and the 3rd quartile); this allows it to absorb the effects of outliers while scaling, making it a more robust tool than the two previous techniques.

These techniques are evaluated in Chapter 6.1.1, where Robust Scaler emerged as the best option and therefore the employed Scaling technique.

4.2 Autoencoder

The autoencoder model chosen for the project was a Variational Autoencoder (VAE). This decision was made due to its architecture having more parameters to tune which give significantly more control over the model, in comparison with traditional encoder models. Although the purpose of this thesis is not to find the most optimal model, improving the model could be a possible future direction, which makes this a good investment. Moreover, in line with the project's objective of utilizing the autoencoder solely for information encoding and not decoding, the focus will be exclusively on the encoder function, while the decoder function will not be utilized. The main language used for this project was *Python*, and an existing VAE model,

available online on *GitHub* [7], was used as a base to ensure the model fits the requirements of the project, a few changes were made to the original model. These include exploring and customising parameters such as the model's loss function, activation function, optimiser, batch size, epoch size, and learning rate. Additionally, changes were made to allow the latent dimension of the model to be customized and passed onto the model instead of static 2 dimensions. Acceptable values for the number of latent dimensions could range from a minimum of 2 to a maximum of 256. However, for exploratory purposes, there is not a significant constraint. Furthermore, the original Sigmoid activation function on the network output layer was removed. This activation proves useful for classification tasks where there is a need to ensure that the output values are between 0 and 1, representing probabilities for each class. However, since the data will not be scaled between 0 and 1, it is preferable not to confine the output to a specific range by applying an activation function. Additionally, it's worth noting that in Chapter 6, the tuning of parameters for the current model focused solely on the encoding aspect and did not involve decoding. Since the model is utilized exclusively for data encoding and not for decoding or reconstruction purposes, the parameter tuning was carried out using the entire dataset without splitting it into separate train and test sets.

Activation Function	Loss Function	Optimizer	Learning Rate	Batch Size	Epochs
Tanh	MAE	Adam	0.001	16	2

Table 1: Features of the Model

The details of the exploration of the different parameters of the VAE model indicated above can be found in Chapter 6.1.2. The resulting parameters from this investigation are displayed in Table 1.

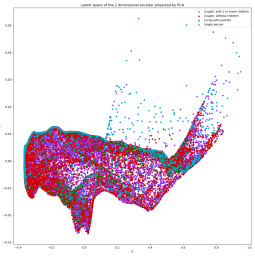
4.3 Visualization

In order to visualise the information in the latent space, the final steps of the pipeline outlined in Chapter 3.3 are used. The first step involves applying a projection algorithm to project the high-dimensional data encoded by the VAE model onto a bidimensional latent space.

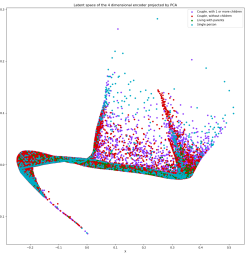
Several projection algorithms were considered, including Principal Component Analysis (PCA), which attempts to identify the directions of maximum variance in data and project it onto a lower-dimensional space. Additionally, non-linear algorithms were also considered, namely t-distributed Stochastic Neighbour Embedding (t-SNE) and Uniform Manifold Approximation and Projection (UMAP), which aim to preserve the local structure of the data in the projection. These experiments were performed in *Python*,

utilising the *sklearn.decomposition.PCA* and *sklearn.manifold.TSNE* modules from the *scikit-learn* library, as well as the *umap-learn* library for UMAP. Each of these experiments was carried out on the entire dataset of the German pilot, including both environmental and consumption metrics, and is coloured by the intervention phase, detailed in Chapter 3.1. Therefore, each point in the projections corresponds to the encoded consumption and environmental information of a particular participant on a particular date. The results for the execution time comparison of each algorithm can be found in Chapter 6.1.3.

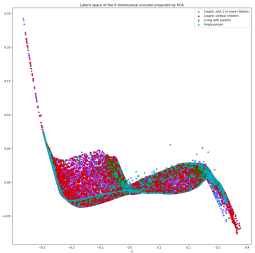
Figure 7: PCA Latent Space



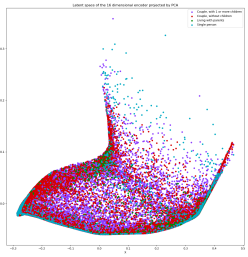
(a) PCA with 2 latent dimensions



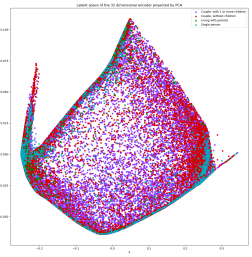
(b) PCA with 4 latent dimensions



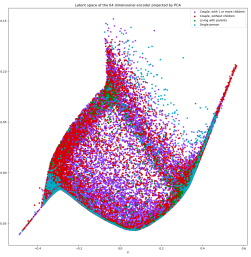
(c) PCA with 8 latent dimensions



(d) PCA with 16 latent dimensions



(e) PCA with 32 latent dimensions

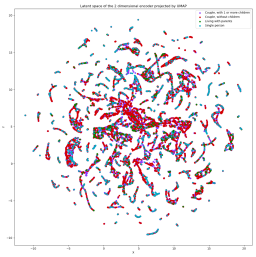


(f) PCA with 64 latent dimensions

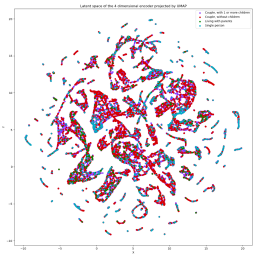
Figures 7, 8 and 9 show the results of the experimented plots. Each projection algorithm was explored for 2, 4, 8, 16, 32 and 64 dimensions. Furthermore, the experiments carried out on t-SNE also explored different learning rates (50, 200 and 500) and perplexity values (5, 40, 100).

Since the autoencoder already provides a scatterplot when the selected latent dimension is 2, it may seem redundant to run a projection algorithm on top of it. However, projecting the latent space using other techniques allows for comparison and assessment. PCA, t-SNE, and UMAP differ in how they capture and emphasize the data's structure, resulting in variations in the visualizations. While the 2D latent space of the autoencoder is specific

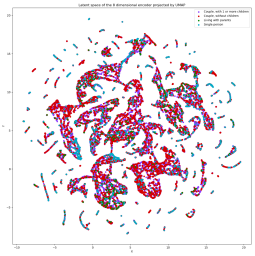
Figure 8: UMAP Latent Space



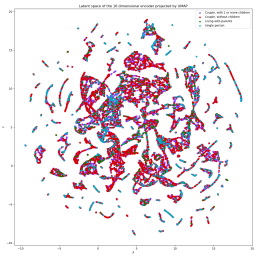
(a) UMAP with 2 latent dimensions



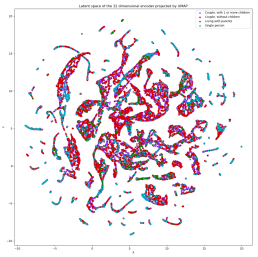
(b) UMAP with 4 latent dimensions



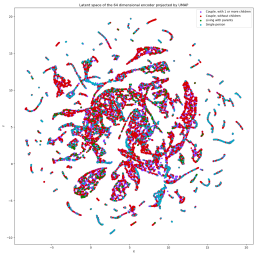
(c) UMAP with 8 latent dimensions



(d) UMAP with 16 latent dimensions

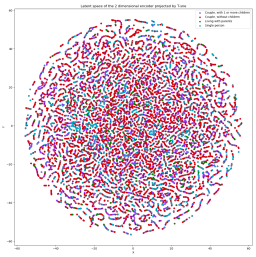


(e) UMAP with 32 latent dimensions

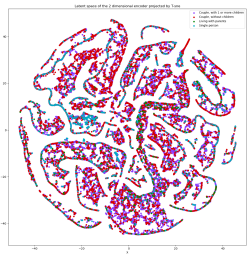


(f) UMAP with 64 latent dimensions

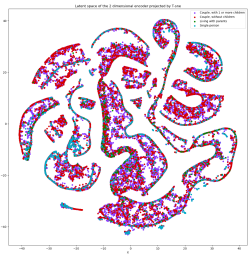
Figure 9: t-SNE Latent Space of 2 dimension



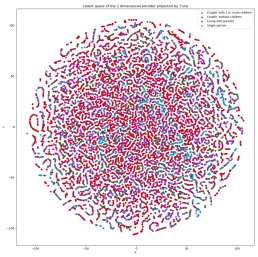
(a) t-SNE with perplexity 5, learning rate 50 and 2 latent dimensions



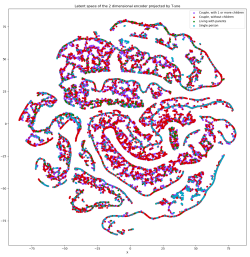
(b) t-SNE with perplexity 40, learning rate 50 and 2 latent dimensions



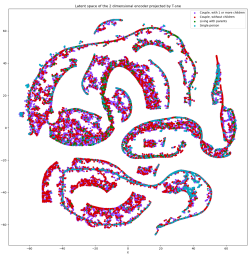
(c) t-SNE with perplexity 100, learning rate 50 and 2 latent dimensions



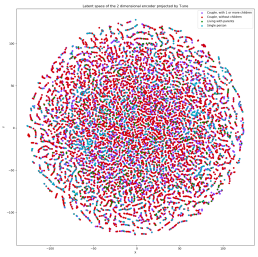
(d) t-SNE with perplexity 5, learning rate 200 and 2 latent dimensions



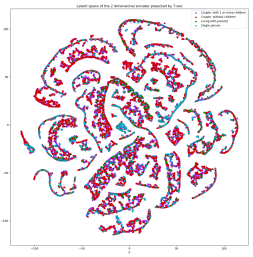
(e) t-SNE with perplexity 40, learning rate 200 and 2 latent dimensions



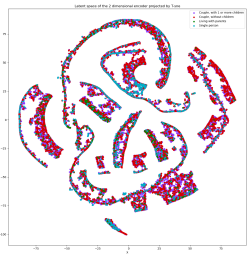
(f) t-SNE with perplexity 100, learning rate 200 and 2 latent dimensions



(g) t-SNE with perplexity 5, learning rate 500 and 2 latent dimensions

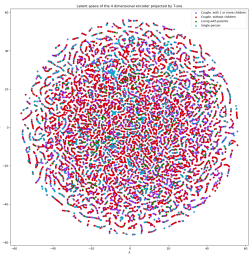


(h) t-SNE with perplexity 40, learning rate 500 and 2 latent dimensions

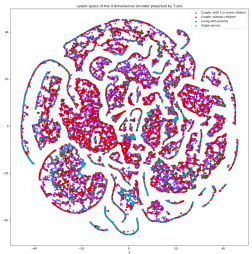


(i) t-SNE with perplexity 100, learning rate 500 and 2 latent dimensions

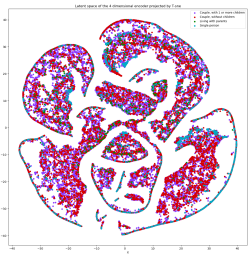
Figure 10: t-SNE Latent Space of 4 dimensions



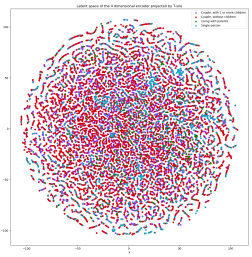
(a) t-SNE with perplexity 5, learning rate 50 and 4 latent dimensions



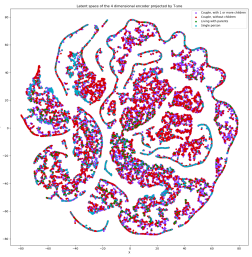
(b) t-SNE with perplexity 40, learning rate 50 and 4 latent dimensions



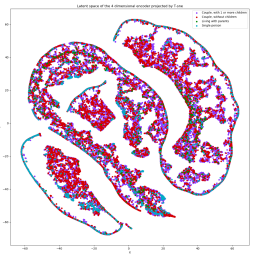
(c) t-SNE with perplexity 100, learning rate 50 and 4 latent dimensions



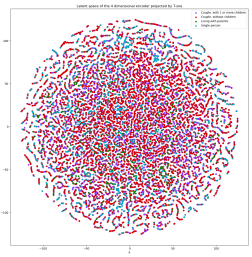
(d) t-SNE with perplexity 5, learning rate 200 and 4 latent dimensions



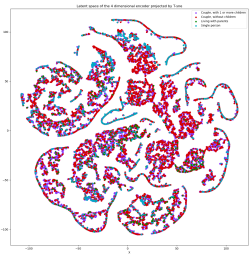
(e) t-SNE with perplexity 40, learning rate 200 and 4 latent dimensions



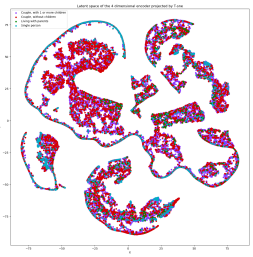
(f) t-SNE with perplexity 100, learning rate 200 and 4 latent dimensions



(g) t-SNE with perplexity 5, learning rate 500 and 4 latent dimensions

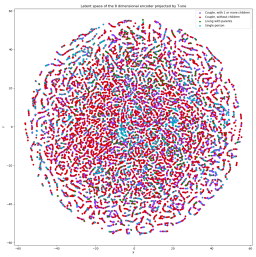


(h) t-SNE with perplexity 40, learning rate 500 and 4 latent dimensions

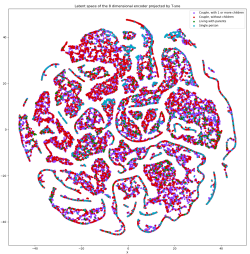


(i) t-SNE with perplexity 100, learning rate 500 and 4 latent dimensions

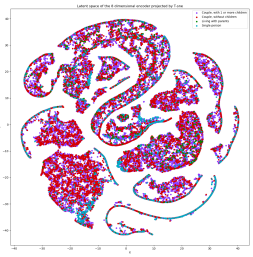
Figure 11: t-SNE Latent Space of 8 dimensions



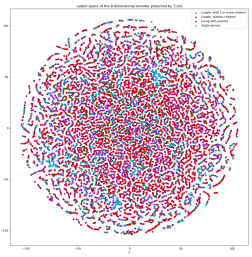
(a) t-SNE with perplexity 5, learning rate 50 and 8 latent dimensions



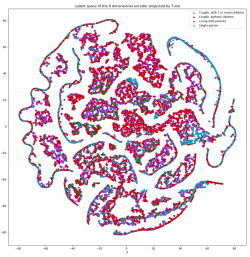
(b) t-SNE with perplexity 40, learning rate 50 and 8 latent dimensions



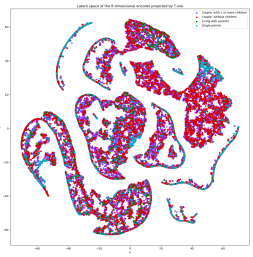
(c) t-SNE with perplexity 100, learning rate 50 and 8 latent dimensions



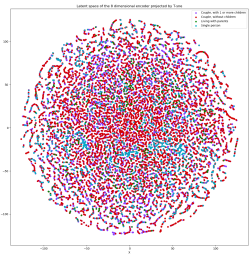
(d) t-SNE with perplexity 5, learning rate 200 and 8 latent dimensions



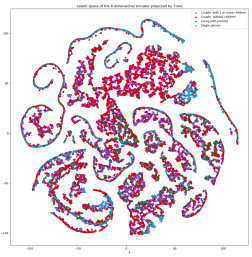
(e) t-SNE with perplexity 40, learning rate 200 and 8 latent dimensions



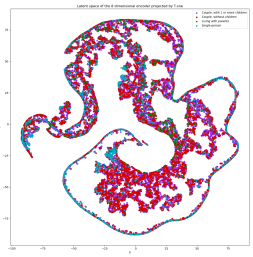
(f) t-SNE with perplexity 100, learning rate 200 and 8 latent dimensions



(g) t-SNE with perplexity 5, learning rate 500 and 8 latent dimensions

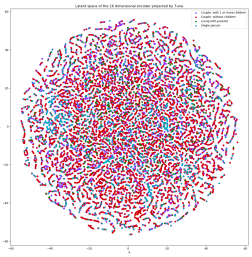


(h) t-SNE with perplexity 40, learning rate 500 and 8 latent dimensions

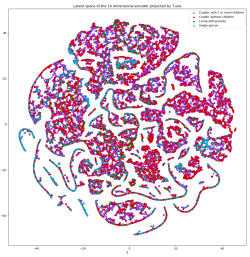


(i) t-SNE with perplexity 100, learning rate 500 and 8 latent dimensions

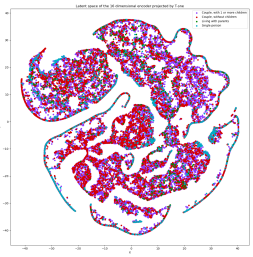
Figure 12: t-SNE Latent Space of 16 dimensions



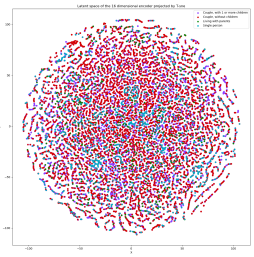
(a) t-SNE with perplexity 5, learning rate 50 and 16 latent dimensions



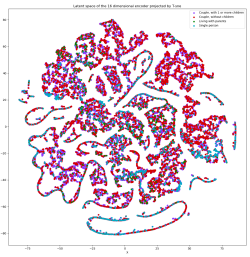
(b) t-SNE with perplexity 40, learning rate 50 and 16 latent dimensions



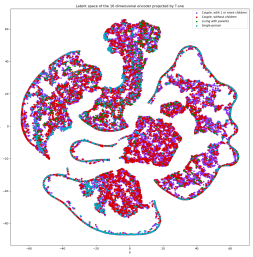
(c) t-SNE with perplexity 100, learning rate 50 and 16 latent dimensions



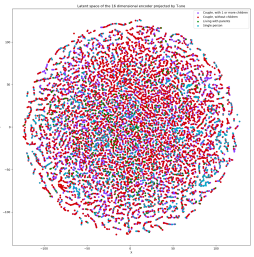
(d) t-SNE with perplexity 5, learning rate 200 and 16 latent dimensions



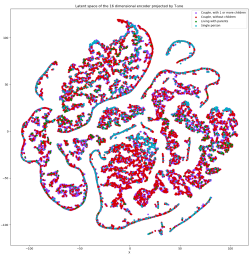
(e) t-SNE with perplexity 40, learning rate 200 and 16 latent dimensions



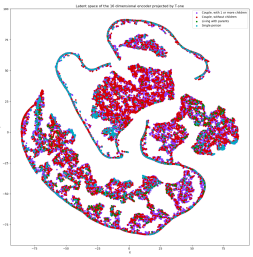
(f) t-SNE with perplexity 100, learning rate 200 and 16 latent dimensions



(g) t-SNE with perplexity 5, learning rate 500 and 16 latent dimensions

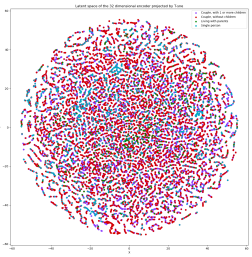


(h) t-SNE with perplexity 40, learning rate 500 and 16 latent dimensions

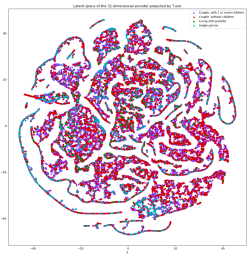


(i) t-SNE with perplexity 100, learning rate 500 and 16 latent dimensions

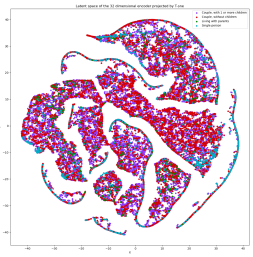
Figure 13: t-SNE Latent Space of 32 dimensions



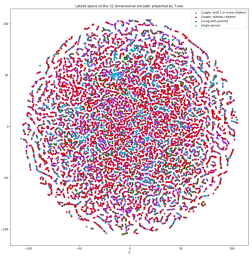
(a) t-SNE with perplexity 5, learning rate 50 and 32 latent dimensions



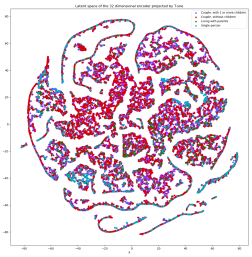
(b) t-SNE with perplexity 40, learning rate 50 and 32 latent dimensions



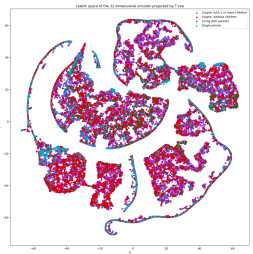
(c) t-SNE with perplexity 100, learning rate 50 and 32 latent dimensions



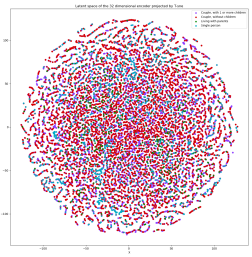
(d) t-SNE with perplexity 5, learning rate 200 and 32 latent dimensions



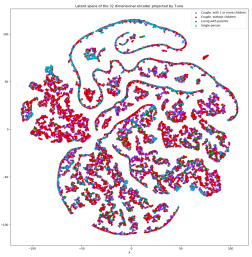
(e) t-SNE with perplexity 40, learning rate 200 and 32 latent dimensions



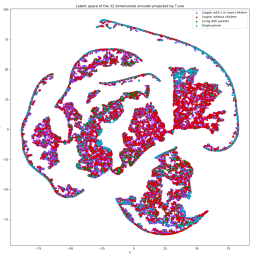
(f) t-SNE with perplexity 100, learning rate 200 and 32 latent dimensions



(g) t-SNE with perplexity 5, learning rate 500 and 32 latent dimensions

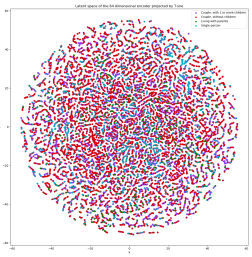


(h) t-SNE with perplexity 40, learning rate 500 and 32 latent dimensions

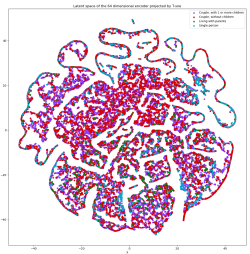


(i) t-SNE with perplexity 100, learning rate 500 and 32 latent dimensions

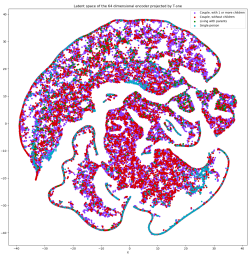
Figure 14: t-SNE Latent Space of 64 dimensions



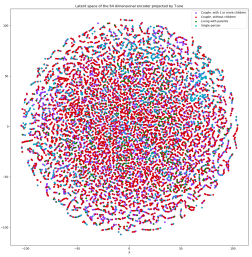
(a) t-SNE with perplexity 5, learning rate 50 and 64 latent dimensions



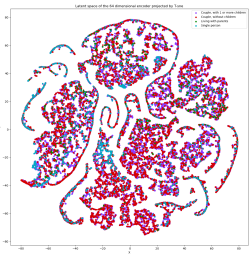
(b) t-SNE with perplexity 40, learning rate 50 and 64 latent dimensions



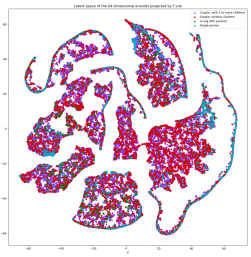
(c) t-SNE with perplexity 100, learning rate 50 and 64 latent dimensions



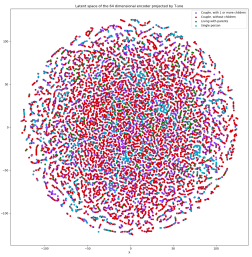
(d) t-SNE with perplexity 5, learning rate 200 and 64 latent dimensions



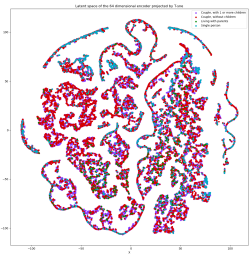
(e) t-SNE with perplexity 40, learning rate 200 and 64 latent dimensions



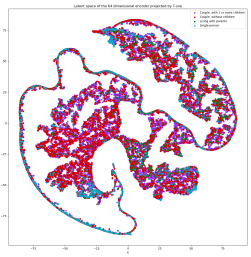
(f) t-SNE with perplexity 100, learning rate 200 and 64 latent dimensions



(g) t-SNE with perplexity 5, learning rate 500 and 64 latent dimensions



(h) t-SNE with perplexity 40, learning rate 500 and 64 latent dimensions



(i) t-SNE with perplexity 100, learning rate 500 and 64 latent dimensions

to the dataset and objective, other techniques highlight different aspects and patterns. Comparing the visualisations sheds light on the advantages and disadvantages of each approach, providing thorough knowledge of the dataset from many angles.

When analysing the t-SNE projections, a few observations are worthy of mention. Firstly, different learning rates seem to also have different effects on the latent projection. Higher learning rates reflect a higher density of data points while lower learning rates reflect the opposite. Such differences can be seen when comparing Figures 10c to Figure 10i. This happens because increasing the learning rate, leads the optimization process in the latent space to become more aggressive, resulting in a denser concentration of data points. On the other hand, a reduced learning rate slows down the optimisation process, leading to a larger spread of data points. Another subtle change can be seen in the latent projection when changing the number of dimensions of the encoded data. When the dimension of the encoded data is low, the points in the t-SNE plot appear to be closer together, indicating a higher density of data points in certain regions of the latent space. In contrast, when the dimension of the encoded data is high, the points in the t-SNE plot are more spread out, suggesting a lower density of data points in the latent space. This difference is hard to visualize when comparing projections of lower perplexity rates, such as Figure 14a and Figure 13d, as well as when comparing projections with a close number of dimensions, for instance, Figures 14i and 13i. However, the difference becomes clearer when comparing low-dimensional encoded models with higher-dimensional ones, both using high-perplexity values. One such example is to compare Figure 14i with Figure 9i. This shift in the latent projection occurs because of the relationship between perplexity and dimensionality. Higher perplexity values in the t-SNE emphasise global relationships, causing neighbouring points in the latent space to look closer together. Increasing the dimensionality of the encoded data allows for greater room for point spread. As a result, larger perplexity values and higher dimensionality provide unique visual patterns in the latent projection, indicating that local density fluctuations are preserved. Another reason why this increase in spread might occur is due to the inherent clustering tendency of autoencoders. When the pre-clustered data is passed to t-SNE, it can create additional gaps in the projection, resulting in a more dispersed visualisation. By a significant margin, the most substantial and noticeable effect can be observed when adjusting the perplexity values, which aligns with the previous observation. Low perplexity values, such as the one in Figure 12d, cause the data to look like a 'ball' with any point approximately equidistant from its nearest neighbours. Higher perplexity values, such as depicted in Figure 12f prove to preserve global structures instead of local structures, causing the data points to become densely packed. It is noteworthy to mention that given time constraints, it was not feasible to conduct additional experiments with different parameter values. There-

fore, the experiments were carried out on the basis of initial impressions. However, in retrospect, it becomes evident that a perplexity value of 5 was inadequate to effectively project the data.

Moving on to UMAP, shown in Figure 8, the experiments yielded projections similar to the higher perplexity values of t-SNE, causing the data points to become densely packed. This resemblance is due to the fact that both approaches prioritize retaining global structures while retaining some local aspects, which results in the capture of comparable underlying patterns in the data.

Lastly, PCA, shown in Figure 7, yielded plots that differed significantly from the other two projections. The resulting projections exhibit a surface-like appearance, with certain points exhibiting higher concentrations along specific lines within the projection. This can be observed by examining the concentration of "Single people" in the corners of the projection in Figure 7e, which is not as evident in other projection algorithms.

Although the ability to cluster data in the latent space may be able to provide some insight into the data, the objective of this project is to visualize consumption as a trajectory over time. While both UMAP and t-SNE are effective clustering techniques and are able to cluster information in the latent space somewhat better than PCA. The linear nature of PCA allows for points to be captured in a trajectory-like shape while also exhibiting some minimal clustering capability. With that said, and given that the technical evaluation described in Chapter 6 elects PCA as the most time-effective projection, PCA emerges as the optimal projection algorithm for this project.

The second step involves the development of a visualisation tool to visualise and explore the data in the latent space. The visualization tool complements the projection algorithm by providing a means to gain insights from the latent space representation. It was developed in *Python*, and using the *Streamlit*, *Pandas*, *Numpy* and *Malplotlib* libraries.

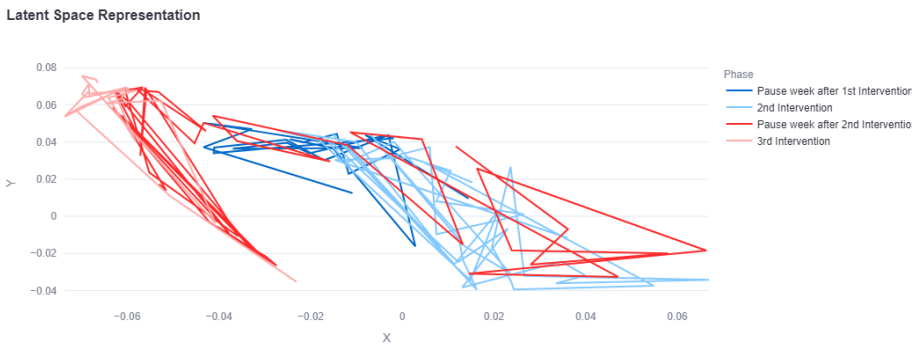


Figure 15: Latent Space Representation

Figure 15 exemplifies the latent space representation of the energy consumption of a single participant in the developed web application. The model incorporates consumption metrics specific to the Croatian pilot and employs colour-coded differentiation for intervention phases. Each data point within the figure possesses a tooltip, providing comprehensive details on the associated sensor and demographic information when hovering over it. Analysing these tooltips allows us to discern the representation of model features. In this case, the x-axis corresponds to energy consumption, while the y-axis denotes energy production.

More details on the interpretation of these graphs can be found in Chapter 4.4.6 and more information on the Streamlit web app can be found in Chapter 5.

4.4 Pattern Taxonomy

This Chapter details the patterns most commonly seen in the latent space representation followed by their interpretation. Before exploring these patterns, it is essential to identify the elements that have a major influence on the latent space representation, namely the model's features and dimensions.

4.4.1 Features

Energy Metrics	Environmental Metrics
Autarky Rate	Temperature
Self Consumption	Radiation
Overall Consumption	Time
Production	

Table 2: Energy and Environmental Features

Figure 2, displays the two main groups of features that pilots possess. The first group included measurements for the consumption or production of energy, such as the autarky rate, self-consumption, overall energy consumption, and production. It is noteworthy to mention that for the energy research project the combination of features is different for every pilot, given that each pilot has different sustainability goals. These metrics were deemed particularly relevant for the model as they contained all the information essential for our understanding. On the other hand, the second group consisted of environmental parameters, such as radiation, temperature, and time. These parameters were included with the intention of gaining deeper insights into participant behaviour and to understand how this group potentially influences the first group of metrics. We predicted that, for instance, higher radiation levels would be associated with increased energy production, or

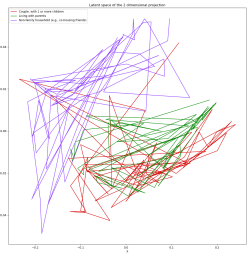
extreme temperature levels could be linked to elevated consumption levels. However, latent space analysis found no correlations between these features. Additionally, analyzing the data outside the latent space confirmed that this lack of connection was not a shortcoming of the latent space analysis. In other words, despite our efforts to thoroughly investigate it, the relationship between the environmental characteristics and the consumption/production metrics was not evident.

Furthermore, given the patterns found in Chapters 4.4.3 and 4.4.4, it is evident that environmental metrics are not conducive for meaningful analysis. With that being said, the recommended features to incorporate into the model are exclusively those from the metrics group, excluding any metrics derived from environmental parameters.

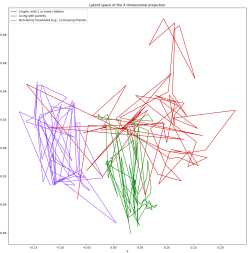
4.4.2 Latent Dimensions

Another essential component of the model and, subsequently the latent space representation, is the number of latent dimensions. Experiments were carried out to determine the optimal number of dimensions a model should have. In order to assess this, experiments were carried out for models with varying numbers of features, namely 2, 3, 4 and 5 features and 2, 4, 8, 16, 32, 64, 128 and 256 dimensions. The data used for this experiment concerns Group 6 of the Croatian pilot, with a total of 3 participants, each with different family type demographics, and the distribution of features goes as follows: In the models with 2 features, the primary inputs were self-consumption and overall-consumption metrics. For the models with 3 features, an additional time feature was included. In the 4-feature model, radiation was added as an extra input, and in the 5-feature model, temperature was included. All of these projections are coloured by the Family type demographic.

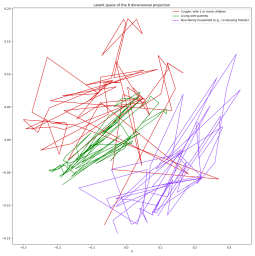
Figure 16: Latent Space of the VAE model with 2 features and (varying) latent dimensions



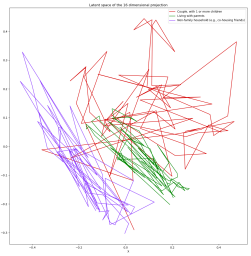
(a) Latent Space of model with 2 features and 2 latent dimensions



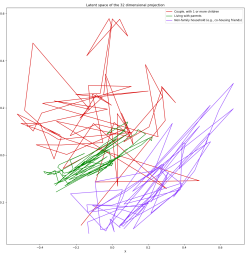
(b) Latent Space of model with 2 features and 4 latent dimensions



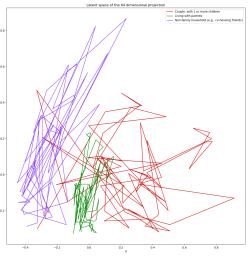
(c) Latent Space of model with 2 features and 8 latent dimensions



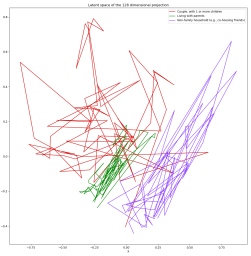
(d) Latent Space of model with 2 features and 16 latent dimensions



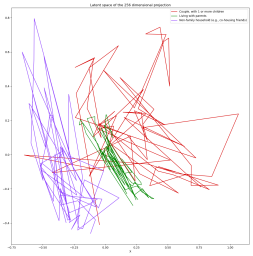
(e) Latent Space of model with 2 features and 32 latent dimensions



(f) Latent Space of model with 2 features and 64 latent dimensions

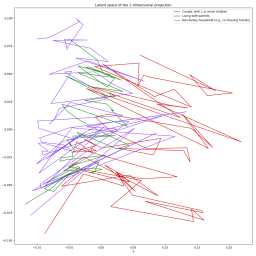


(g) Latent Space of model with 2 features and 128 latent dimensions

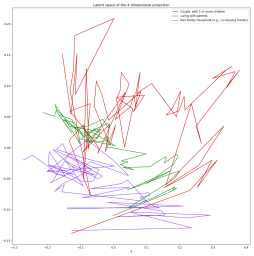


(h) Latent Space of model with 2 features and 256 latent dimensions

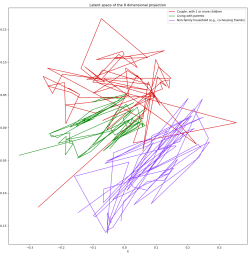
Figure 17: Latent Space of the VAE model with 3 features and (varying) latent dimensions



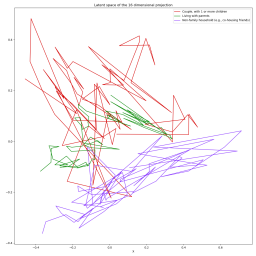
(a) Latent Space of model with 3 features and 2 latent dimensions



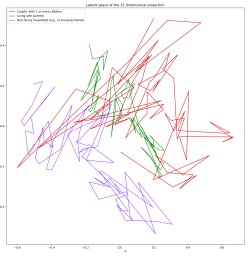
(b) Latent Space of model with 3 features and 4 latent dimensions



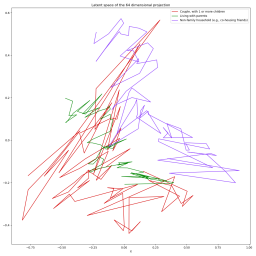
(c) Latent Space of model with 3 features and 8 latent dimensions



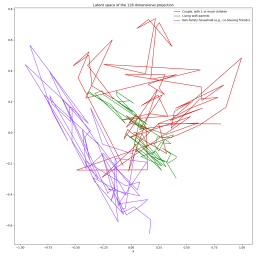
(d) Latent Space of model with 3 features and 16 latent dimensions



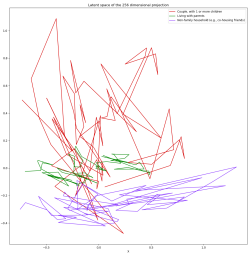
(e) Latent Space of model with 3 features and 32 latent dimensions



(f) Latent Space of model with 3 features and 64 latent dimensions

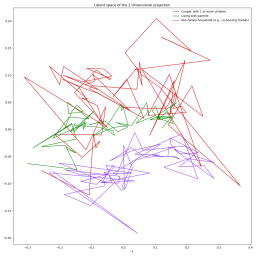


(g) Latent Space of model with 3 features and 128 latent dimensions

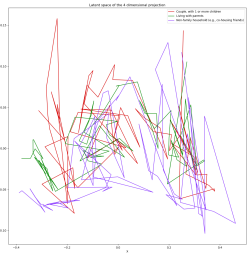


(h) Latent Space of model with 3 features and 256 latent dimensions

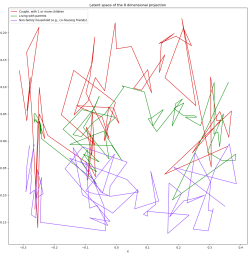
Figure 18: Latent Space of the VAE model with 4 features and varying dimensions



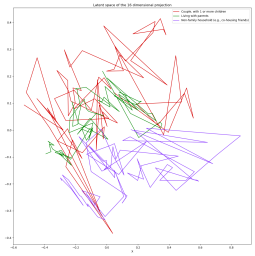
(a) Latent Space of model with 4 features and 2 latent dimensions



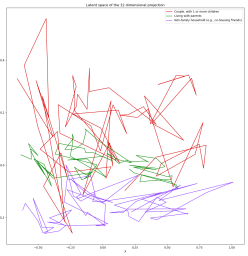
(b) Latent Space of model with 4 features and 4 latent dimensions



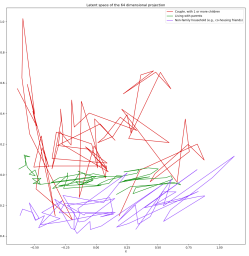
(c) Latent Space of model with 4 features and 8 latent dimensions



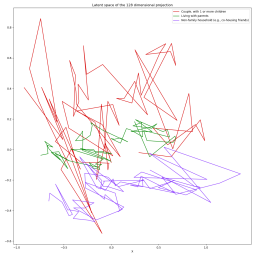
(d) Latent Space of model with 4 features and 16 latent dimensions



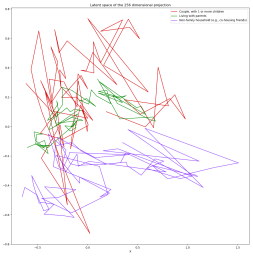
(e) Latent Space of model with 4 features and 32 latent dimensions



(f) Latent Space of model with 4 features and 64 latent dimensions



(g) Latent Space of model with 4 features and 128 latent dimensions



(h) Latent Space of model with 4 features and 256 latent dimensions

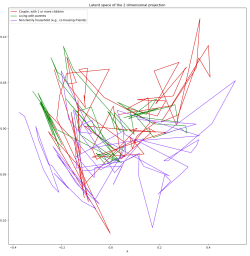
The results of these experiments can be found in Figures 16, 17, 18 and 19. Each line in the latent space represents the trajectory of a participant with respect to the features that were added to the model.

Considering Figure 16, which shows the Latent Space of a model with two features and different dimensions, as an illustration. A comparison of Figure 16a (with two dimensions) and Figure 16b (with four dimensions) clearly shows that the latter allows superior cluster distinction, making it the most suitable number of dimensions for this scenario. Similarly, in Figures 16c, 16d, 16e, 16f 16g and 16h, the addition of more dimensions results in an identical latent space, although with a small rotation. This pattern, where increasing the number of dimensions helps distinguish clusters easier is seen consistently across all models with varied features. Furthermore, there is a point beyond which the latent space exhibits no discernible changes, appearing identical except for potential rotations. This threshold that reflects the optimal number of dimensions appears to vary depending on the number of features in the model. Notably, as additional features are included, a greater number of dimensions are often required to provide optimal separation across clusters. For instance, the model with 3 features, displayed in Figure 17 reaches its threshold at 16 dimensions in Figure 17d, further increment on the latent dimensions yields the same latent space with a small rotation, displayed in Figures 17e, 17f, 17g and 17h. Similarly, for the model with 4 features, displayed in Figure 18, the latent space reached its threshold at 16 dimensions, seen in Figure 18d, making Figures 18e, 18f, 18g and 18h small rotations of the same plot. Finally, the model with 5 features, previewed in Figure 19 demands 32 dimensions to separate clusters efficiently, as Figure 19e suggests, making Figures 19f, 19g and 19h rotations of the best latent space dimension plot.

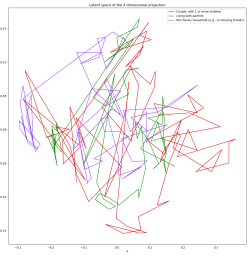
4.4.3 Latent Stripes

During earlier explorations of the latent space, an intriguing striped pattern was identified. Manual investigation and the use of tooltips to obtain insights into the underlying data points revealed that the presence of radiation or temperature in the latent space was responsible for the formation of these stripes. Despite the intriguing shape, this anti-pattern was found to obscure participant behaviour, hindering the ability to understand how participants reacted to interventions.

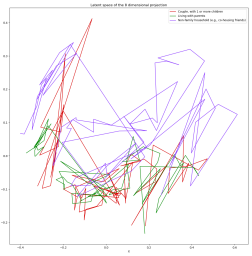
Figure 19: Latent Space of the VAE model with 5 features and varying dimensions



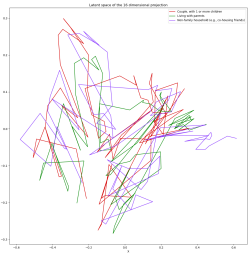
(a) Latent Space of model with 5 features and 2 latent dimensions



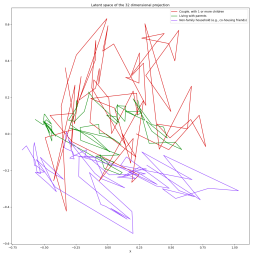
(b) Latent Space of model with 5 features and 4 latent dimensions



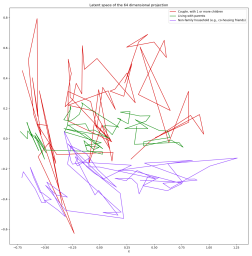
(c) Latent Space of model with 5 features and 8 latent dimensions



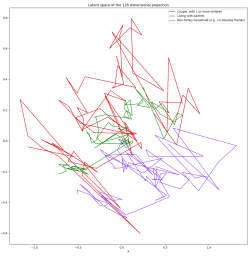
(d) Latent Space of model with 5 features and 16 latent dimensions



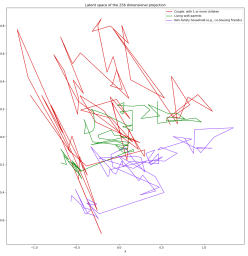
(e) Latent Space of model with 5 features and 32 latent dimensions



(f) Latent Space of model with 5 features and 64 latent dimensions

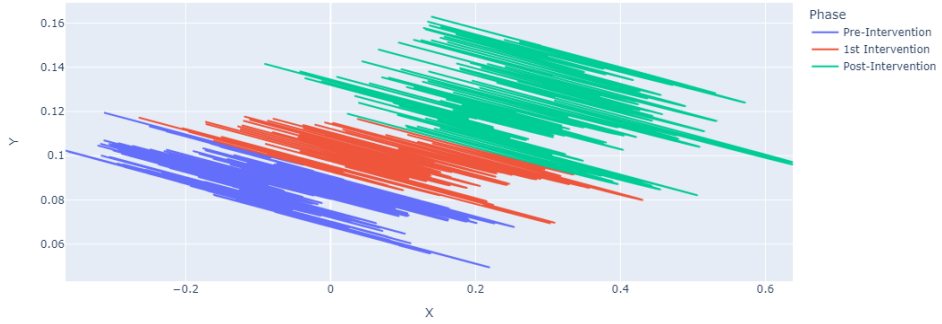


(g) Latent Space of model with 5 features and 128 latent dimensions

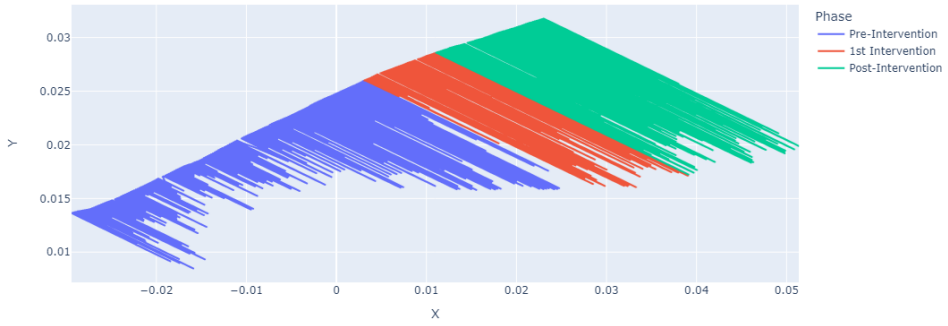


(h) Latent Space of model with 5 features and 256 latent dimensions

Figure 20: Latent Space with Environmental Metrics



(a) Latent Space with Temperature



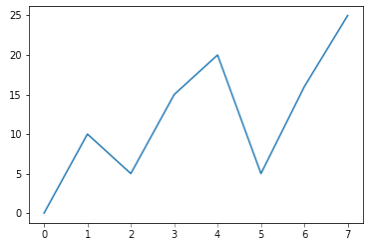
(b) Latent Space with Radiation

Figure 20 depicts the latent space projection of a partial section of the dataset representing the German pilot, coloured by the Intervention Phase. The model used to generate the projection included consumption metrics and environmental features, with hourly data. In Figure 20a, the introduction of temperature into the latent space, along with consumption metrics, produces intriguing stripe-like patterns. Initially, these patterns appear enigmatic, but using the tooltip as a lookup aid reveals that they are linked to fluctuations in temperature throughout the day. The stripes depict the progression of temperature levels as they rise steadily, peak, and then slowly decline. Similarly, in Figure 20b, the introduction of radiation in the model yields striped patterns that represent the rise and fall of radiation levels throughout the day. Additionally, a straight line forms in the latent space since radiation is exactly zero between sunset and morning. From these pictures, no insights can be gained regarding energy consumption or participants' behaviour, since all of the patterns and trends captured by the model are highly representa-

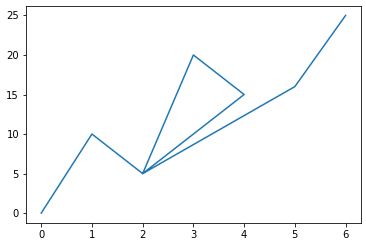
tive of these environmental metrics. Therefore, we can conclude that these distinctive patterns in temperature and radiation overwhelm the visibility of consumption metrics within the latent space, making it difficult to detect or analyse any behavioural patterns.

4.4.4 Time Partitioning

Figure 21: Effects of Time as a Feature of the Latent Space



(a) Latent Space with Time as a feature



(b) Latent Space without time as a feature

Another pattern was found by including environmental metrics, in this case, the Time feature. Given that we are dealing with MTS data it might seem contradicting that Time is added as a feature and not included in the model by default. However, information regarding time, as well as any other feature and demographic value, can be accessed through a tooltip in the latent space at all times. Furthermore, incorporating time as a feature ensures that points within the latent space are clearly separated. This means that when time is included, trajectories are consistently distinct. For instance, consider Figure 21, which serves as an illustrative example of a model that incorporates consumption, production, and time as features. If a participant has identical production and consumption values at two different time points, these points will never overlap in the latent space due to the inherent nature of the time dimension. Such behaviour can be visualized in Figure 21a. On the other hand, as we see in Figure 21b, if time is excluded, these points would overlap. As illustrated in the picture allowing points to overlap enables trajectories to be clearer. When two points in the latent space coincide, it becomes evident that the corresponding feature values for those points are identical. This characteristic simplifies the identification of similar data points, subsequently enhancing the analysis process. This becomes even more crucial for large datasets with much larger time periods, where more attention is needed for analysis. Therefore excluding Time as a direct feature of the model proves to be more beneficial than including it. Nonetheless, identifying the temporal location of the data points in the latent space

is always available no matter what features are included in the model, by means of the Visualization tool, described in Chapter 5.

4.4.5 Outliers in the Latent Space

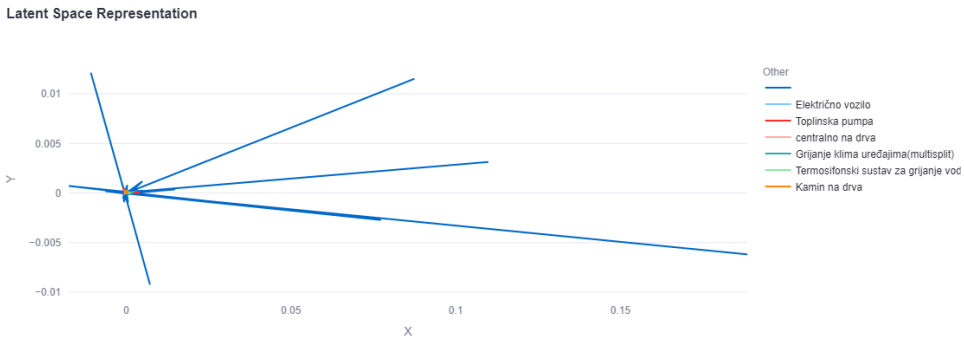


Figure 22: Outliers in the Latent Space

Another pattern discovered by interacting with the data was the existence of outliers in the latent space. An example is plotted in Figure 36. When projecting all of the data together, a few points that find themselves considerably distant from others in the latent space can be easily distinguished.

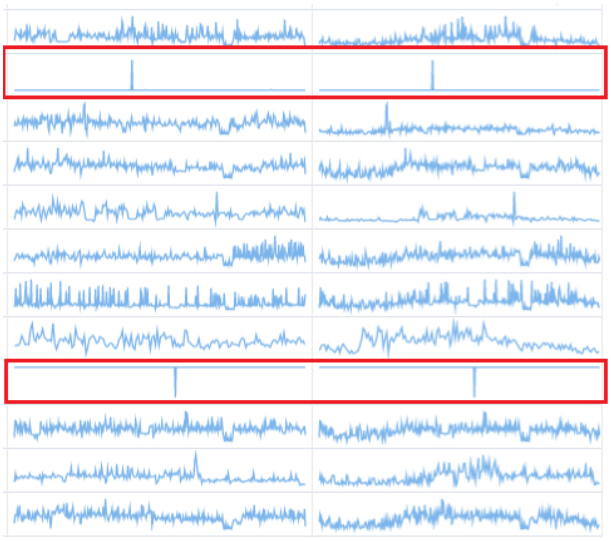


Figure 23: Corrupted Data

In order to understand where these outliers came from, the original data they represent was thoroughly examined. It was found that all of these latent

outliers represented corrupted data points, which were reflected in the latent space. In Figure 23, the consumption graphs of two outliers are highlighted in red. While it can be confirmed that these artefacts were not produced by the encoder, since they were present in the data before any encoding took place and are exclusively related to noisy data points, the exact source of these outliers can only be speculated upon. However, it is highly likely that they originated from malfunctioning sensors. Furthermore, while these outliers can be identified by observing the individual consumption graphs of each participant, using the latent space allows for this analysis to be much quicker.

4.4.6 Response Pattern

When projecting the encoded information onto the latent space, it is important to note that the alignment of features with the x or y axis may not always be perfect, especially in models with more than three features. In such cases, the orientation of features may deviate from the conventional axis alignment. However, examining the latent space with the use of the tooltip helps us identify these variations and gain insights into how the features provided to the model are represented by the axes.

Figure 24: Latent Space Representation of Simple Response Pattern

An example is illustrated in Figure 24. By analysing it with the tooltip, we discover that the x-axis corresponds to energy consumption, while the y-axis represents energy production. Armed with this knowledge, we are able to tell where in the latent space the participant behaved in a certain way. For instance, we recognize that the lower right part of the plot is represented by a high production rate and low consumption rate. Furthermore, colouring the data by categories allows for distinguishments to be made. In this case, the plot is coloured by the intervention phase.

If we take this into consideration, we can deduce that prior to the third

intervention, energy production levels were consistently low, and consumption levels varied from low to high. The third intervention stands out as it resulted in the highest energy production levels throughout the experiment, accompanied by lower overall consumption levels compared to other phases. Hence, we can conclude that the participant responded positively to the third intervention and unfavourably to the second intervention.

Figure 25: Latent Space Representation of Complex Response Pattern

A more challenging example is depicted in Figure 25, the axis follows the same conventions as the previous case, but the responses differ. It is evident in this case that the first intervention yielded the most favourable outcomes, with lower consumption and higher production values. Subsequent intervention phases resulted in significantly higher consumption levels.

4.5 Ethics and Privacy

The Utrecht University Research Institute of Information and Computing Sciences conducted an Ethics and Privacy Quick Scan (see Appendix A). It evaluated this research project as low-risk, requiring no additional ethics review or privacy assessment.

5 Interface

A web app was developed with *Streamlit* [32] to allow for an easy exploration and visualization of the data. For the purpose of this project, the application was developed and run locally on a local machine. However, *Streamlit* allows the application to be deployed to the cloud. With that said, three main pages were developed. The first page was dedicated to interactively exploring latent space analysis. The second page was designed to display horizon graphs, which allowed for a clear and concise representation of data trends over time. Finally, a third page was developed to display information for participants.

5.0.1 Latent Space Visualization

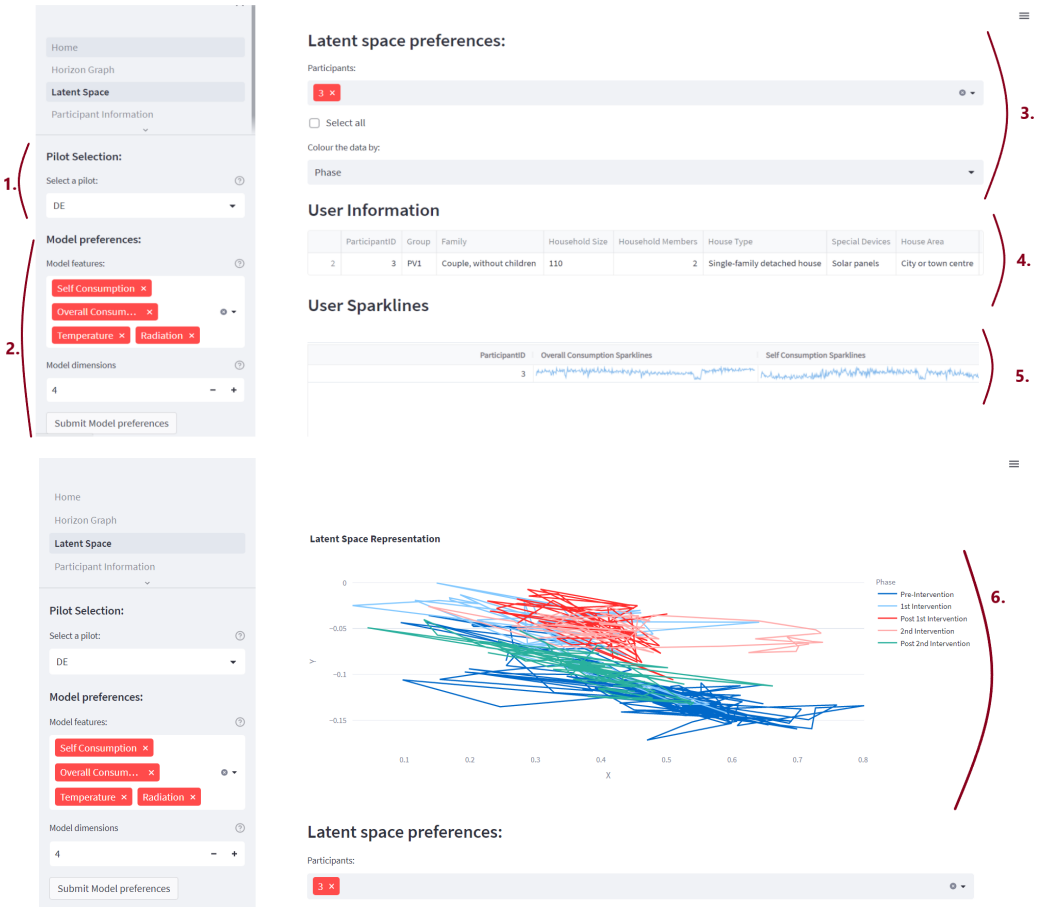


Figure 26: Latent Space Visualization Page

In order to provide a visual representation of the design of the latent

space page, a screenshot was captured and presented in Figure 26. At the top left corner of the page the pilot selection (1) section is displayed, it contains a selection box with the pilots implemented. Users can choose the pilot they are interested in exploring from this box. Once they select one, the next section appears.

Directly below the first section, the Model Preferences (2) section is displayed, this section contains a multi-selection box with the features of the model and a numerical input section for the dimensions of the model. The features displayed change according to the selected pilot, while there is no set of rules for which features to select, a few options are advised on the informative balloon. Furthermore, for exploratory reasons, there are no dimensionality constraints, however, an advised range of dimensions is also displayed to guide users in selecting appropriate dimensions for their exploration. More detail into this advice is given in Section 4.4. Users may then submit their preferences by pressing the "Submit Model preferences" button and unlock the next sections.

On the top right side of the page, the latent space preferences (3) section is displayed, which allows for the selection of the desired participants and the category to colour (distinguish participants by). The option to select all participants at once is also allowed with the "Select All" selection button. This option directly supports two of the analysis tasks, since it allows to differentiation of the data on different factors (**T2**) and allows for the exploration of contributing factors for the nudging performance (**T5**).

Once these preferences are selected, the latent space (6) section is displayed along with the user information (4) and the user sparklines (5). The Latent space representation figure displays the latent space of the selected participants coloured by the selected category. The information is displayed in the form of a trajectory per participant. By hovering over the data, the user is able to visualise a tooltip with all the information regarding that specific instance, this includes any consumption metric, social demographics, or any other information the model holds. This serves as a way to gather more insight and decipher patterns in the data. Users are also able to zoom, pan, autoscale, reset axes and remove categories from the represented latent space.

In order to provide more information to the user regarding the displayed participants and respective consumptions, the user information section (4) and user sparkline (5) are displayed. The first section reflects users' information in the form of a table. The second reflects the information in the form of a consumption sparkline. The displayed information changes according to the selected pilot. Overall, this page allows for the visualization of participant behaviour and changes over time (**T3**) and allows users to explore and differentiate contributing factors to these changes (**T2**, **T5**), as well as discern working from non-working nudges (**T4**).

5.0.2 Horizon Graph Page

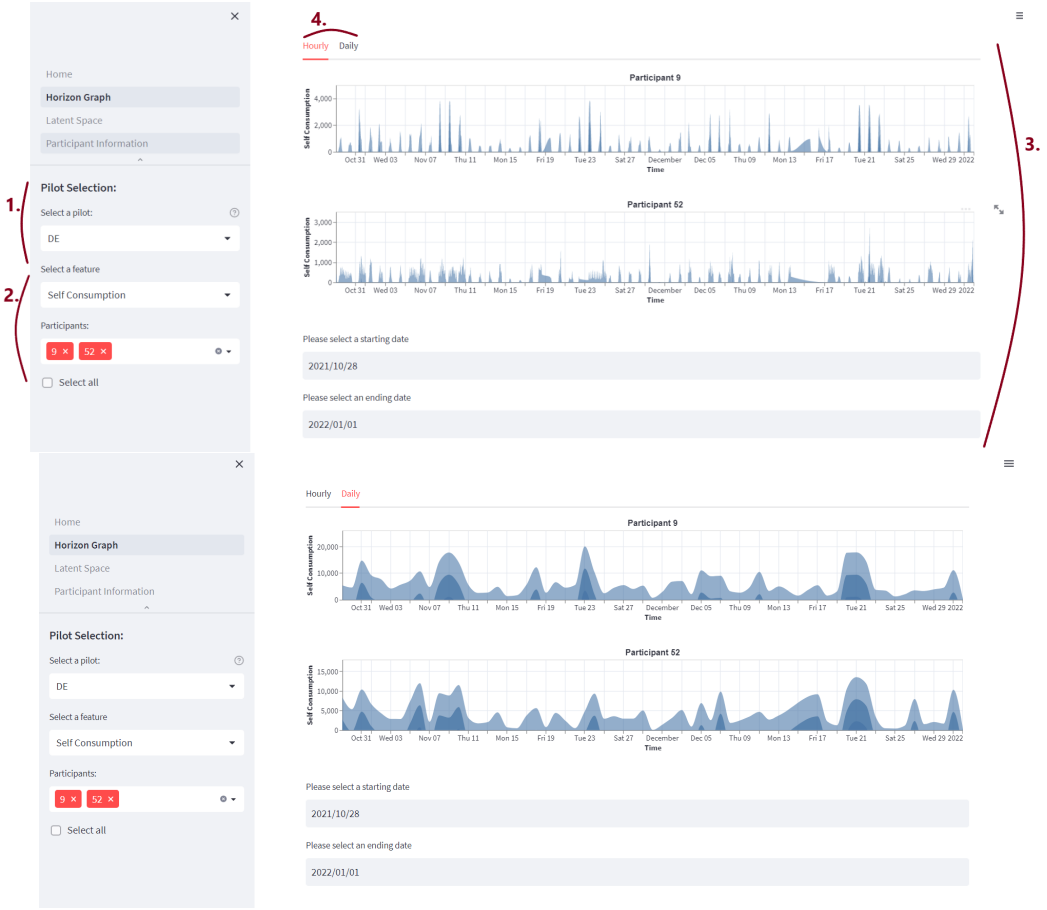


Figure 27: Horizon Graph Page

As mentioned in Chapter 2.4, general graph exploration techniques have long been popular for visualizing MTS data. Despite the many added advantages offered by domain-specific tools, we made the deliberate choice to incorporate an additional option, providing users with a broader range of exploration possibilities. After careful consideration of various options, such as line graphs, bar plots, and heatmaps, Horizon Graphs were selected for this page. Even though Horizon Graphs have some disadvantages, such as the decrease in precision when compared to classical line plots, its advantages outweigh its drawbacks. Particularly, its vertical compactness and ability to detect trends and changes while allowing for the comparison of multiple time series.

As shown in Figure 27, the design of the Horizon Graph page follows a similar layout to the Latent Space page, with several differences. The Pilot Selection section (1) is identical to the previous page and serves the same

purpose, of selecting the desired pilot and initiating the rest of the analysis. The feature and participant selection section (2) also follows the same rules. However, there is a small difference in this section: only one feature can be selected, and instead of being fed to a model, it will serve as the value to be explored in the horizon graph.

When all these selections have been made and processed, the user is guided to select a date range for which to visualise the horizon graph(s). Once the data range has been selected, the horizon graph is displayed (3).

To solve the issue of horizon graphs becoming difficult to visualize for big data ranges, two tabs have been added to the interface (4). The first tab, displayed in the top picture, presents the data hourly, which is ideal for limited data ranges, whereas the second tab, displayed in the bottom picture, was designed for larger data ranges. Users can still receive insight from the horizon graph without being overwhelmed by its intricacy by using the second tab, which presents the data daily. The interface enables users to customize the visualization to their particular needs and ensures they can correctly interpret the findings of their investigation by offering two tabs with various levels of granularity.

5.0.3 Participant Information Page

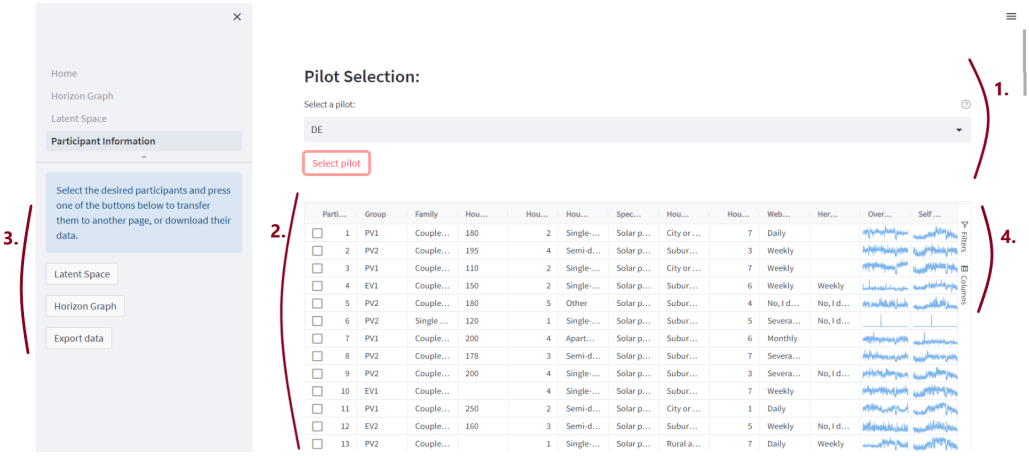


Figure 28: Participant Information Page

A screenshot of the Participant Information page is presented in Figure 28, which introduces a few new design elements from the "Latent Space" and "Horizon Graph" pages but also includes a similar section. The shared section is the Pilot Selection (1), just as in the previous pages, it allows the user to select a pilot and submit it. Once the choice has been made and the "Select Pilot" button has been pressed, a table (2) with all the information the model holds regarding every participant of the pilot is displayed, along

with the sparkline of its corresponding consumption metrics. This allows the user to get an overview of the data (**T1**). Additionally, on the upper right of the table (4) the user has to option to filter and group participants by category, and also has the option to select the columns it wishes to analyse.

Finally, each row has a checkbox which allows the user to select the desired participants once they have been grouped and filtered to meet the users' exploratory needs. Once the participants have been selected the user can press one of the two buttons on the left side of the page (3), which will transfer them to either the latent space analysis page or the horizon graph page. Once on the desired page, the pilot and participants will already be selected and analysis can proceed. Furthermore, a third button is displayed on the bottom for exporting the select rows into a CVS format.

6 Evaluation

This Chapter provides an overview of the evaluation approach and decisions made to build the model in this thesis, as well as a description of real-world application and the outcomes of using the visualization tool. It is noteworthy to mention that different evaluation metrics for deep learning models and projection algorithms exist. However, due to time constraints, a comprehensive evaluation of these techniques could not be conducted in this study. Moreover, the primary focus of the project was not to determine the optimal model but rather to explore whether such a model could effectively project behaviour.

6.1 Technical Evaluation

This Chapter provides an overview of the evaluation approach and decisions made to build the model in this thesis.

6.1.1 Normalization

As mentioned in Chapter 4.1 experiments were done with the Standard Scaler, Min-Max Scaler and Robust Scaler from the *sklearn.preprocessing* module of the *Scikit-learn* library, in order to find the best scaling algorithm to process the data.

Chapter 4.1 mentions the disparity of ranges between features of the model. These include temperature, which is measured in $°C$ and ranges from -5 to 30; radiation, measured in Jm^{-2} ranges from 0 to 30M; and consumption values between 0 and 40 KWh, including a few outliers.

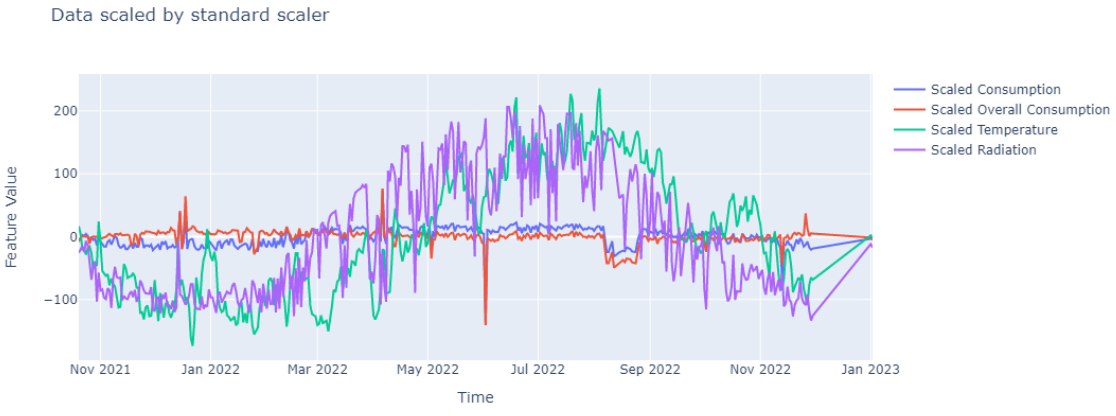


Figure 29: Visual Representation of the data scaled by the Standard Scaler

Figure 29 illustrates the limitations of the Standard Scaler, where we

can see that the technique is sensitive to outliers and allows temperature and radiation features to dominate. This is likely due to its assumption that features are normally distributed. Therefore, the Standard Scaler is not the optimal choice for normalization. It is noteworthy to mention that despite the expected scaled range in most cases being -1 to 1 due to standardization, in this case, the range is significantly higher, from -200 to 200. This happens due to the presence of outliers or extreme values in the dataset since features have significantly different scales. These values have a significant impact on the standardization process, leading to a broader range of standardized values.

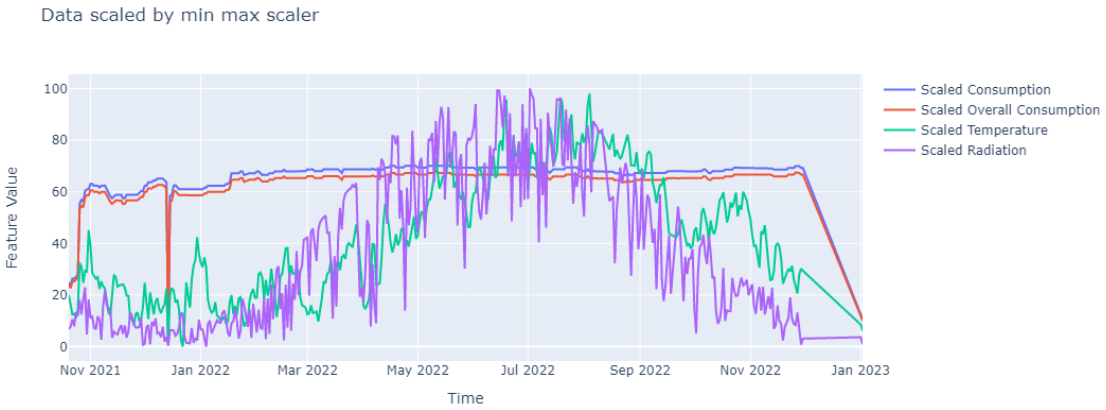


Figure 30: Visual Representation of the data scaled by the Min-Max Scaler

Figure 30 visually shows the data scaled by the Min-Max Scaler. Initially, the data was scaled between -1 and 1. However, due to the significant differences in the magnitudes of the feature values, some features ended up being reduced to 0. To address this issue, a new scaling range was manually chosen. The range selected for this scaler was between -200 and 200, similar to the previous technique used. Although this technique has the benefit of scaling information within the same range, it is far more sensitive to outliers compared to standard scalers. Additionally, it allows certain features to dominate. This property makes the min-max scaler a poor choice for this particular problem, as it does not address the issues of feature dominance and outliers.

Figure 31 presents the data scaled using the Robust Scaler. This technique successfully overcomes the limitations observed with the Standard Scaler and Min-Max Scaler. The Robust Scaler is not sensitive to outliers and effectively prevents features, such as temperature and radiation, from dominating the scaled data. Therefore, based on this visual evidence, the Robust Scaler surges as the most suitable option for normalization.

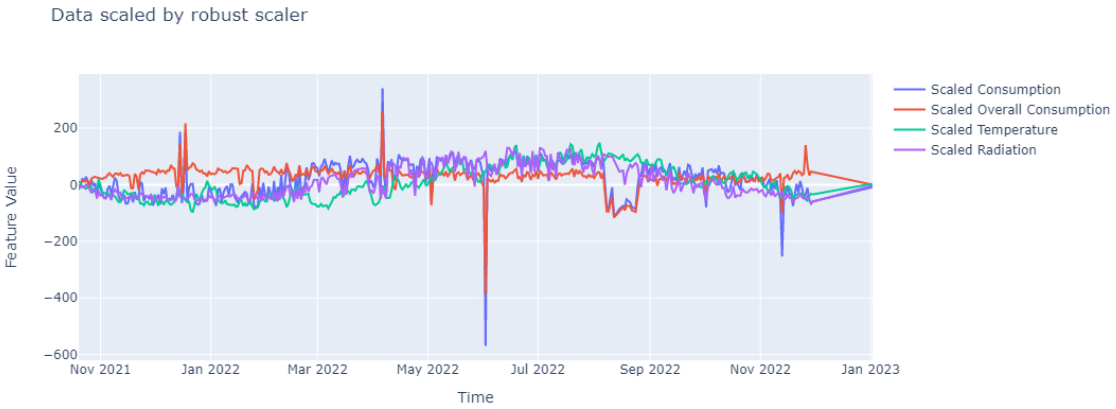


Figure 31: Visual Representation of the data scaled by the Robust Scaler

6.1.2 Variational Autoencoder Model (VAE)

In order to tailor the VAE model to our project, modifications were made to a base model obtained from the GitHub repository 'Autoencoder' by Mattia Campana [7], implemented in *Python* with the *Keras* library.

Initially, the model employed the Binary Crossentropy (BCE) loss function, which is commonly employed for binary multiclass classifications involving binary outputs. However, considering that our model addresses continuous output predictions, alternative loss functions suitable for regression tasks were evaluated. Notably, Mean Squared Error (MSE) and Mean Absolute Error (MAE) were considered potential replacements.

Given the robustness of MAE in handling outliers, it was selected as the preferred choice for our model. Consequently, the BCE loss function was replaced with MAE to better align with the nature of the regression task at hand.

To determine the optimal activation function, a series of experiments were conducted involving two prominent functions: Rectified Linear Unit (ReLU) and Hyperbolic Tangent (Tanh). These activation functions were specifically chosen due to their recognized effectiveness in regression models. The model was trained under identical conditions throughout the experiments, employing a consistent set of parameters, including 50 epochs and a batch size of 100. Each experiment was performed 5 times.

Training Loss	2.7355	3.2446	2.8530	3.1234	2.7672
Execution Time	40s	1m10s	57s	1m1s	56s

Table 3: ReLU Activation Experiments

Training Loss	2.3807	2.6008	2.5999	2.5994	2.6043
Execution Time	1m6s	1m1s	1m16s	1m2s	1m20s

Table 4: Tanh Activation Experiments

The outcomes of the experiments are presented in Tables 3 and 4. Notably, the ReLU activation function took less execution time, on average, than the Tanh function. However, it is noteworthy that the Tanh activation function yielded the lowest loss value, thereby establishing itself as the optimal choice among the considered options.

To determine the optimal set of parameters for model optimization, a grid search approach was utilized. The grid search involved exploring various combinations of optimizers, batch sizes, and learning rates. The optimizers considered were Adam, SGD, and RMSprop. The batch sizes considered were 16, 32, 64, 128, and 256, while the learning rates examined were 0.0001, 0.001, and 0.01.

The experimental setup ensured consistent architecture and maintained other parameters, including the MAE loss function, and Tanh activation. Given that this model will be integrated and used as part of a Visualization Tool, and therefore compiled in real-time, having a lower execution time (without compromising quality) is an important thing to consider. Furthermore, given that the model takes about 24 seconds on average per epoch to be trained, decreasing the number of epochs becomes a necessary step. Therefore, by combining all of these different hyperparameters we wish to discover the combination that ensures an optimal training loss for two epochs. The outcomes of these experiments are presented in Table 5.

By analysing Table 5 we can see that higher learning rates learn quicker early on and yield better results than others during the first epochs. On the contrary, lower learning rates start slow, yielding bad results in the first epochs, but most likely reach more optimal values later on in the training. Additionally, for all models lower batches yield better results for all learning rates. Therefore, we identify ADAM with 16 batches and a 0.001 learning rate as the ideal option, given that it achieves the lowest loss value of all the combinations (3.2663).

6.1.3 Projection Algorithm

In order to discover the most suitable projection algorithm to project the data encoded by the VAE model onto a bidimensional latent space, various experiments were conducted. These experiments were performed using PCA, t-SNE, and UMAP.

Table 6 displays the results of the execution time of the experiments conducted on the projection algorithms. According to 6, PCA proved to be

Optimizer	Batch Size	Learning Rate		
		0.01	0.001	0.0001
Adam	16	3.4890	3.2663	5.3579
	32	3.5747	3.3545	7.1736
	64	3.4898	3.9145	11.1036
	128	4.9871	3.3957	13.4817
SGD	16	3.5678	5.2199	11.3224
	32	3.5973	5.6992	21.3604
	64	4.3474	7.9954	26.1485
	128	5.1903	10.9517	29.3458
RMSProp	16	3.6891	3.3420	5.4177
	32	4.0599	3.4706	6.2852
	64	4.4055	4.1995	9.6177
	128	4.8658	4.8331	13.3968

Table 5: Optimizer Grid Search Experiments

Projection Algorithm	Learning Rate	Perplexity	Dimensions					
			2	4	8	16	32	64
t-SNE	50	5	444.72	368.08	370.26	334.42	336.51	339.74
		40	518.46	436.88	406.74	398.02	393.71	404.08
		100	578.68	583.12	525.93	510.66	512.74	515.38
	200	5	365.71	356.45	298.40	322.81	329.04	332.12
		40	420.87	442.33	336.71	388.92	391.21	408.96
		100	555.59	1109.51	498.78	564.35	505.55	608.75
	500	5	365.53	500.60	290.06	333.63	337.20	337.82
		40	428.29	557.68	356.84	392.80	387.68	338.36
		100	558.46	1665.24	462.38	504.50	511.56	507.97
PCA	N/A	N/A	2.65	16.77	11.29	16.80	22.39	21.51
UMAP	N/A	N/A	38.58	32.84	38.68	36.56	39.89	50.87

Table 6: Projection Algorithms Experiments

the most time-efficient algorithm taking at most 22 seconds to project 32 encoded dimensions. UMAP also proved to be time efficient taking at most 50 seconds to project 64 dimensions. On the other hand, t-SNE is much more computationally expensive, given that the lowest amount of time it took to train was 383 seconds.

In Chapter 4.3, the projection plots and their analysis were extensively explored. Among the many techniques evaluated, PCA stood up as the most successful in detecting clusters in the latent space. Furthermore, PCA displayed outstanding time economy, reinforcing its position as the best projection approach for the project.

6.2 Case Studies

Due to time restrictions, it is not possible for this project to provide an exhaustive description of all possible use cases that the tool could address. As a result, a carefully curated set of representative use cases has been chosen for presentation.

6.2.1 Group Case Study: As a user, I want to get an overview of how many different groups exist on the German pilot.

In this case study, the user successfully uses the app to get an overview of the German pilot. This case study directly corresponds to the first analysis task (**T1: Get an overview of the data**). The user follows the following steps in order to achieve their goal:

- Open the Exploratory App on your device.
- Navigate to the "Participant Information" page in the app menu.
- Select the German pilot from the "Pilot Selection" dropdown box.
- Click the "Select pilot preference" button to proceed to the next step.
- Hoover above the "Group" column on the table and press the "order by" button.
- Scroll through the table and visualize the number of groups.

By analysing the table displayed in Figure 32, the user is able to identify 4 groups for the German pilot. Users have the flexibility to sort participants by any category they choose, allowing for the customized organization of the data. Additionally, the table allows for filtering options, enabling users to extract specific information based on their needs. The inclusion of sparklines to visualize participants' consumption adds another useful feature, allowing users to quickly understand consumption trends. Furthermore, from this table users are allowed to export selected data, and transfer it to other pages in the app, such as the latent space and the Horizon Graphs.

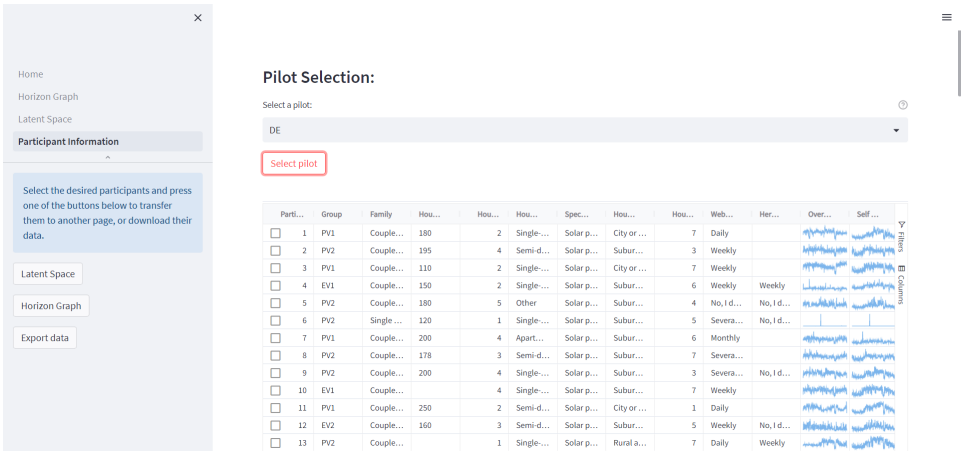


Figure 32: German "Participant Information" table

6.2.2 Intervention Case Study: As a user, I want to visualize the impact of different interventions on consumer 53 from the Croatian pilot.

In this case study, the user successfully uses the app to identify the most successful interventions for influencing consumers towards efficient energy consumption. This case study directly relates to the fourth analysis task: (**T4: Discern working from non-working nudges**). The user follows the following steps in order to achieve their goal:

- Open the Exploratory App on your device.
- Navigate to the "Latent Space" page in the app menu.
- Select the German pilot from the "Pilot Selection" dropdown box.
- Click the "Select pilot preference" button to proceed to the next step.
- Choose Consumption and Production as the features to include in the model from the "Feature Selection" multi-select box.
- Select 8 dimensions for the model on the "Model Dimensionality" box.
- Click the "Submit model preference" button to generate the model.
- Choose Participant 53 from the "Participants:" multi-select dropdown box.
- Select category "Phase" to colour the data, from the "Color data by:" dropdown menu.
- Visualize the latent space and observe the trajectory of the participant's energy consumption patterns.

- Compare the trajectories for each intervention phase to identify the most effective interventions for promoting efficient energy consumption.

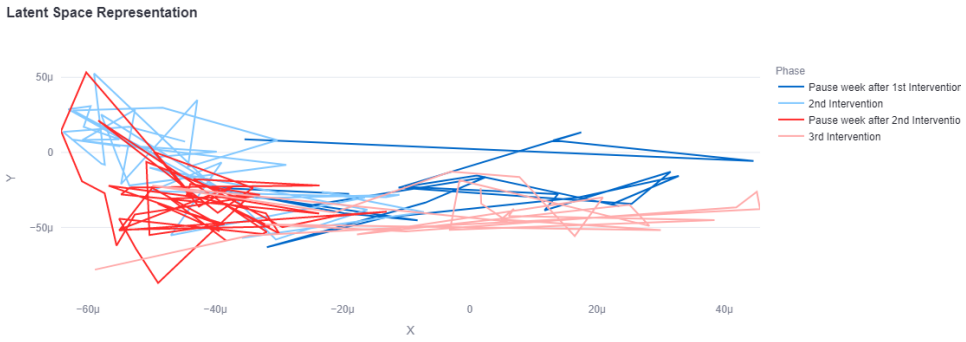


Figure 33: Latent Space representation of participant 53 from the German pilot

Upon analyzing the generated latent space, depicted in Figure 33, using the tooltip as a pattern lookup aid, a few conclusions can be drawn regarding the graph. Firstly, it becomes apparent that the x-axis represents the participant's energy production, while the y-axis represents their energy consumption. This *Response Pattern* has been explained in Chapter 4.4.6. With this in mind, a closer examination of the latent space reveals that the second intervention had a negative impact on the user, resulting in an increase in energy consumption and a decrease in energy production compared to the period before the intervention. In contrast, the third intervention was effective in reducing energy consumption and increasing energy production, making it the most beneficial intervention of the three. Therefore, based on the analysis of the latent space, it can be concluded that the third intervention was the most successful, and the second intervention was the least successful in achieving the desired outcomes.

The ability to analyze multiple features in the model simultaneously is a significant benefit of using the Exploratory App. This feature allows users to evaluate and compare different interventions on a broader scale and gain a more comprehensive understanding of their impact on energy consumption. Additionally, the app's visualization tools make it easier for users to spot trends and patterns in the data and quickly identify areas that require attention. By enabling users to assess multiple factors simultaneously, the tool simplifies the analysis process and makes it easier for them to draw meaningful conclusions from their data

A few improvements could be considered in future work to make users have an easier time analysing patterns and trends. Given that only the

tooltips are available as a pattern lookup aid, it can become tedious to "manually" analyse the latent space for insight. A solution to this problem would be to automatically identify key values in the latent space, such as the minimum consumption and the maximum consumption, or to visually identify by means of a "wind map" in which direction of the latent space the included features are represented.

6.2.3 Social demographic Case Study: As a user, I want to determine whether different family types of the 6th intervention group of the Croatian pilot have significantly different responses to interventions.

In this case study, the user wishes to determine whether the family type social demographical factor has any impact on the interventions of the 6th group of the Croatian pilot. This case study effectively addresses two key analysis tasks: (**T2: Allow to differentiate on different facets (pilot, demographics, etc)** and **T5: Explore contributing factors for nudge "performance"**). The user follows the described steps in order to achieve their goal:

- Open the Exploratory App on your device.
- Navigate to the "Participant Information" page in the app menu.
- Select the "Croatian" pilot from the "Pilot Selection" dropdown box.
- Click the "Select pilot preference" button to proceed to the next step.
- Select the "Filter" option from the participant information table.
- Select the "Group" option from the dropdown menu
- Type group number "6" to analyse it.
- Select all the participants filtered in the table.
- Press the "Latent Space" button to jump to the Latent space analysis.
- Choose the " Consumption" and "Production" features from the "Feature Selection" multi-select box to include them in the model.
- Select the number of dimensions "8" for the model.
- Click the "Submit model preference" button to generate the model.
- Select the social demographic category "Family Type" to colour the data from the "Color data by:" dropdown menu.

- Visualize the latent space and observe the clusters formed by the colouring of the selected factor.
- Compare the trajectories for each cluster and determine whether there are significant differences.

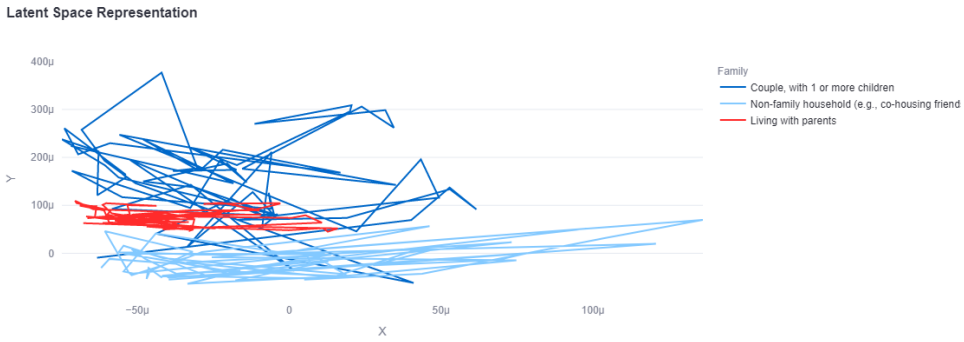


Figure 34: Latent space representation of Germany's Group 6, colour-coded by family types

Upon analyzing the latent space, depicted in Figure 34, it becomes evident that the coloured trajectories, corresponding to different family types, occupy clearly distinct regions of the latent space. By examining the axis of the plot with the aid of the tooltip, we can see that the x-axis represents the participant's energy production, while the y-axis represents their energy consumption. This *Response Pattern* is explained in Chapter 4.4.6. A closer inspection of the plot reveals that couples with children tend to have more varied energy consumption patterns, often reaching higher values than other family types. On the other hand, non-family households consume less energy and produce more energy than the other family types. Single individuals are located somewhere in between, with moderate levels of energy consumption and lower levels of production. With this, the user can conclude that, indeed, different family types respond differently to the interventions.

While the tooltip allows for the identification of which intervention each data point refers to, for the sake of interpretation, Figure 35 conveys the same data plotted by intervention. By looking at the plot we can follow the same axis rules as the previous figure given that it's the same latent representation coloured by a different category. In doing so we can reach a few conclusions. First, we can see that different intervention phases do not affect consumption for any of the family types. Despite being able to identify that the second intervention phase leads participants to produce twice as much energy as before, this increase in production seems to be proportional to each family type's previous production. With that said different interventions do not seem to have different impacts on distinct social demographic groups.

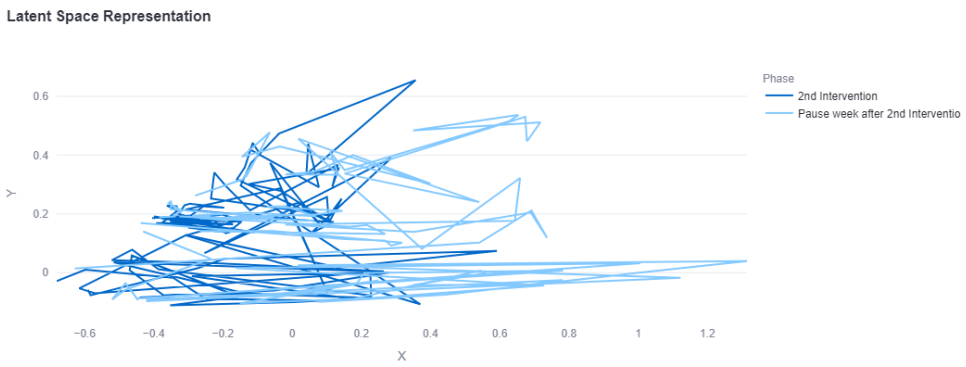


Figure 35: Latent space representation of Germany's Group 6, color-coded by intervention phase

It is worth mentioning that this lack of effect on different family types from different interventions is consistent among other social demographic groups, and that this is only an example.

The tool makes it easy to visualize and compare clusters to one another, which allows analysis to be easier and more fluid. Despite being able to find all the categorized information in the tooltips, this case study would benefit from an additional way to visualize two distinct categories in the latent space. To achieve this, one future improvement is to retain the colour coding feature for one category and introduce symbols to represent the other category.

6.2.4 Noise Case Study: As a user, I want to be able to detect noise present in the German pilot.

In this case study, the user intends to identify any present noise in the German data. The user follows the steps listed below:

- Open the Exploratory App on your device.
- Navigate to the "Latent Space" page in the app menu.
- Select the "German" pilot from the "Pilot Selection" dropdown box.
- Click the "Select pilot preference" button to proceed to the next step.
- Choose the "Overall Consumption" and "Self Consumption" features to include in the model from the "Feature Selection" multiselect box.
- Select the number of dimensions "4" for the model.
- Click the "Submit model preference" button to generate the model.

- Tick the "Select all" box to select all available participants in the chosen pilot.
- Select the category "Household Members" to color the data by from the "Color data by:" dropdown menu.
- Visualize the latent space and observe whether there are any outliers present in the graph.

Figure 36: Latent Space representation of outliers from the German pilot

It should be noted that for this particular task, the chosen features are merely an example, and any combination of Self Consumption, Overall Consumption, and Autarky can be utilized, along with the remaining features and any model dimension and colour category can be just as effective in detecting outliers. Nevertheless, upon analyzing the latent space, depicted in Figure 36, clear outliers can be easily identified. In the above figure, five participants with noisy data are well distinguished from the rest of the group. Examining their individual consumption graphs would lead to the same conclusion, as numerous anomalies can be found in their graphs. However, utilizing this tool allows for a five-fold increase in identification speed, as the analysis only needs to be performed once, and the results are just as clear. *Outliers in the latent space* are a common pattern found in latent space analysis and is described in Chapter 4.4.5. Despite the ability to remove any participant from the model can be found in the "Select Participant" selection box, one future improvement for this task would be to have the option to automatically eliminate all of these noisy participants at once.

6.2.5 Time-Difference Case Study: As a user, I want to identify differences in Participant 18's consumption, of the German pilot, over time.

In this case study, the user wishes to identify differences in the energy consumption of participant 18 of the German pilot, over time. This case study specifically focuses on the third analysis task: (**T3: Model and visualize user behaviour in order to track user behaviour change over time**). In order to do this, the user followed the following steps:

- Open the Exploratory App on your device.
- Navigate to the "Horizon Graph" page in the app menu.
- Select the "German" pilot from the "Pilot Selection" dropdown box.
- Choose the "Self Consumption" feature to visualize it over time in the "Select a feature" multi-select box.
- Select participant "87" from the "Participants" multi-select box.
- Select the default starting date, from the "Start date" date input.
- Select the ending date "30/04/2022", from the "End date" date input.
- Visualize the horizon graph and observe how the selected feature changes over time.

Figure 37: Horizon Plot of participant 87 from Germna pilot

In Figure 37, we can see a clear pattern in the energy use of user 87 over a certain time period. Notably, consumption generally tends to rise steadily over this time. The tooltip function allows us to identify the highest consumption value for that period, which occurred on April 19, 2022. With the help of this tool, users are able to maintain track of a number of important variables, such as self and general consumption, and see how these change over time, including when they are at their highest and lowest points. One possible enhancement would be to recognize the start and end of interventions automatically or to identify extreme numbers without relying on the tooltip.

7 Discussion

The pattern taxonomy, described in Chapter 4.4, helped in the detection and understanding of frequently recurring patterns in the latent space. These patterns, when applied to the case studies described in Chapter 6.2, highlight the proposed framework's practical relevance.

The *Intervention Case Study*, detailed in Chapter 6.2.2, provides an illustrative example of how to analyze the impact of interventions on participants, which contributes to addressing the first research question, **RQ1.**, of this project - "Which interventions are most effective in positively influencing a consumer towards efficient energy consumption?". This case study allows us to detect the behaviour of a specific participant towards each intervention it is exposed to, and compare them to each other. However, this analysis reveals that while straightforward answers can be provided for most individuals and participant groups, there isn't an intervention that performs exceptionally well across all participants. It has been observed that individuals respond differently to treatments, implying that there is no single intervention that works exceptionally well or poorly across all participants. Despite the complexity of the research question, we can confidently provide a positive answer. The effectiveness of interventions on consumers' energy consumption can be easily determined by analyzing the latent space representation of the consumers, as demonstrated in Chapter 6.2.2. This analysis allows us to identify whether interventions have a positive or negative impact on consumers' energy consumption.

The ability to compare interventions to each other can be extended to comparing participants to each other, in particular, those of different social demographic groups. This was thoroughly exemplified in the *Social Demographics Case Study*, described in 6.2.3, of which there is a clear distinguishment in consumption behaviour between participants of different family types. This case study allows us to address the second research question, **RQ2.**- "What effects do social-demographic and technical factors have on the interventions?". In the project, we thoroughly examined the various social demographic and technical categories within the latent space for the pilots at hand. However, due to time limitations, we couldn't provide a comprehensive analysis of all categories. You can find an exemplary illustration in Chapter 6.2.3. This particular Case Study aligns with the findings from other investigations into social demographics, leading us to the conclusion that the social-demographic factors observed in the data do not appear to influence the effectiveness of interventions. Thus, in direct response to the research question, it can be stated that social demographic and technical factors do not have an impact on the interventions. However, an additional discovery emerged, indicating that the demographic of family type does play a role in differentiating consumer behaviour. This was evident as participants with diverse social-demographic characteristics displayed distinct patterns of

consumption behaviour.

Several significant insights were derived from analyzing the latent space representation of the sustainability research data, which cannot be obtained solely from the input model representation, addressing the third research question, **RQ3.** - "What novel insights can be gained from analyzing the latent space representation of the sustainability research data that cannot be obtained from the input model representation?". Analysing latent space representations of the sustainability research data allowed us to get deeper insights into participants' responses to interventions while concurrently examining different features and determining their roles, which cannot be performed on the original data input. Furthermore, the ability to compare the behaviours and responses of multiple consumers to various interventions, while considering multiple aspects of their behaviour such as consumption and production, empowers us to draw stronger conclusions. To directly answer the research question, analysis of the latent space provides valuable insights which we do get from the input model representation, including the possibility to identify the most effective and least effective interventions for each participant and make comparisons among different participants as well. Furthermore, latent space analysis allowed us to compare interventions to each other and determine the role of social demographic groups in consumer behaviour, in particular, it was found that participants with Family Type "Couple with Children" tend to consume more and produce less energy than other Family Types, regardless of the Intervention Phase.

Another advantage of latent space analysis in comparison to input model representation is how easy it is to spot outliers in the latent space, which is directly correlated with research question four, **RQ4.** - "Can we distinguish a signal from noise in the extensive input space of the sustainability research project?". Outlier patterns were detected and explained in Chapter 4.4.5 as well as in the *Noise Case Study*, in Chapter 6.2.4. Therefore, in response to the research question, outliers and noise in the data can be effectively identified and differentiated from the rest of the data in the latent space. These outliers appear as data points that are significantly distant from other points in the latent space projection.

The response pattern observed in Chapter 4.4.6, together with the positive results of the case studies reported in Chapters 6.2.1 and 6.2.2, allows us to study the applicability of latent space analysis for detecting behavioural patterns. Our results demonstrate that latent space analysis not only makes it easier to identify behavioural patterns and trends in individuals but also makes it possible to compare behaviours across various groups, and it does so in a visually appealing manner. In light of this, we are confident in our ability to respond to the fifth research question, **RQ5.** - "Can latent space analysis reveal behaviour patterns?" - positively.

With that said, we can positively attest to the ability of the current framework to address the limitations of MTS data. Although there is room

for improvement and further enhancements, as discussed in Chapter 7.1, the existing framework serves as a solid foundation. It shows promise, particularly for potential future predictive models of energy consumption. Moreover, based on the outcomes discussed in Chapter 4.4.6, we can assert that latent space models are applicable tools for identifying behavioural patterns in event-related data. Additionally, these models are able to provide valuable insights, such as the one in Chapter 6.2.1, that can aid policymakers in making informed decisions.

7.1 Limitations & Future Work

While the project was overall successful in answering the research questions a few limitations were found. Firstly, one of the required tasks was not completed. This task regarded the comparison of pilots to find similarities and dissimilarities and was not made possible. Since pilots possess different metrics and interventions took place at different time periods it was not possible to build a model that learns from different pilots. A future improvement would be to make sure these requirements are met and complete this task. Other limitations include the shortage of visualization and manipulation options. While the developed latent space analysis had a tooltip as a pattern lookup aid, it becomes tedious to manually detect trends and feature directions. For this reason, latent space analysis could be improved with the addition of visualization tools such as windmaps, or maps that enhance the identification of important points in the latent space, and the direction in which each added feature is present. Given that the focus of this project was not to discover the most optimal model to effectively project behaviour, a simpler model was constructed, and a future direction would be to optimize it. Finally, a potential future direction for the project includes adapting the model to be capable of prediction. This could prove even more useful for experts looking to identify and optimize interventions for decreased electricity consumption.

8 Conclusion

This thesis suggested a framework for simplifying the complexity of MTS data and deepen its understanding. The employed framework consisted of a combination of machine learning techniques and visualization tools for the encoding, projecting and visualization of the data into a latent space representation. In particular, this thesis applied the framework to a general energy sustainability project, with the goal of understanding how interventions affect behavioural consumption. The framework displayed a multitude of benefits. Firstly, it facilitated the assessment of how interventions influenced users' behaviour. It also aided in detecting the influence of social-demographic factors on users responses to interventions and it allowed to distinguish signals from noises in the latent space. All of this allowed us to conclude that indeed latent space representation is able to reveal behavioural patterns, with added value in comparison to input model representations. Furthermore, we conclude that the employed framework demonstrates efficacy in handling MTS data. Given the capacities of the tool, we expect it to serve as a beneficial resource for researchers to analyse and derive meaningful insights from MTS data.

References

[1] What is principal component analysis (pca) and how it is used? https://www.sartorius.com/en/knowledge/science-snippets/what-is-principal-component-analysis-pca-and-how-it-is-used-507186/, 2020. Accessed: 2023-06-29.

[2] C. Narendra Babu and B. Eswara Reddy. A moving-average filter based hybrid arima–ann model for forecasting time series data. *Applied Soft Computing*, 23:27–38, 2014.

[3] Benjamin Bach, Conglei Shi, Nicolas Heulot, Tara Madhyastha, Tom Grabowski, and Pierre Dragicevic. Time curves: Folding time to visualize patterns of temporal evolution in data. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):559–568, 2016.

[4] Zoltán Bankó, Laszlo Dobos, and János Abonyi. Dynamic principal component analysis in multivariate time-series segmentation. *Conservation, Information, Evolution*, 1:11–24, 01 2011.

[5] Filippo Maria Bianchi, Lorenzo Livi, Karl Øyvind Mikalsen, Michael Kampffmeyer, and Robert Jenssen. Learning representations for multivariate time series with missing data using temporal kernelized autoencoders. *CoRR*, abs/1805.03473, 2018.

[6] Dmitry S Bulgarevich, Miezel Talara, Masahiko Tani, and Makoto Watanabe. Machine learning for pattern and waveform recognitions in terahertz image data. *Scientific Reports*, 11(1):1251, 2021.

[7] Mattia Campana. Autoencoders. https://github.com/mattiacampana/Autoencoders, 2018.

[8] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3), jul 2009.

[9] Song Chen. Beijing PM2.5 Data. UCI Machine Learning Repository, 2017. DOI: https://doi.org/10.24432/C5JS49.

[10] Zhiyuan Chen, Xiaomin Fang, Zixu Hua, Yueyang Huang, Fan Wang, and Hua Wu. Helixmo: Sample-efficient molecular optimization in scene-sensitive latent space. 2022.

[11] D. R. Cox, Gudmundur Gudmundsson, Georg Lindgren, Lennart Bondesson, Erik Harsaae, Petter Laake, Katarina Juselius, and Steffen L. Lauritzen. Statistical analysis of time series: Some recent developments [with discussion and reply]. *Scandinavian Journal of Statistics*, 8(2):93–115, 1981.

[12] Manuel Dahnert, Alexander Rind, Wolfgang Aigner, and Johannes Kehrer. Looking beyond the horizon: Evaluation of four compact visualization techniques for time series in a spatial context. *CoRR*, abs/1906.07377, 2019.

[13] Kemilly Dearo Garcia. *Unsupervised learning approaches for non-stationary data streams*. PhD thesis, University of Twente, Netherlands, April 2021.

[14] Ankur Debnath, Govind Waghmare, Hardik Wadhwa, Siddhartha Asthana, and Ankur Arora. Exploring generative data augmentation in multivariate time series forecasting: opportunities and challenges. *Solar-Energy*, 137:52–560, 2021.

[15] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[16] Zikun Deng, Di Weng, Xiao Xie, Jie Bao, Yu Zheng, Mingliang Xu, Wei Chen, and Yingcai Wu. Compass: Towards better causal analysis of urban time series. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):1051–1061, 2022.

[17] Julien Despois. Latent space visualization — deep learning bits 2. `https://hackernoon.com/latent-space-visualization-deep-learning-bits-2-bd09a46920df`, 2017. Accessed: 2023-06-29.

[18] Yi Ding, Neethu Robinson, Su Zhang, Qiuhao Zeng, and Cuntai Guan. Tsception: Capturing temporal dynamics and spatial asymmetry from eeg for emotion recognition. *IEEE Transactions on Affective Computing*, pages 1–1, 2022.

[19] P. Conradie M. Karaliopoulos S. Pelka F. Anagnostopoulos, S. Van Hove. Research methodology for assessing the effectiveness of interventions regarding change of energy-efficient behaviour (d2.2). 2022.

[20] Paulo Fernandes, João Correia, and Penousal Machado. Towards latent space exploration for classifier improvement. 08 2020.

[21] Candace Flatt and Ronald Jacobs. Principle assumptions of regression analysis: Testing, techniques, and statistical reporting of imperfect data sets. *Advances in Developing Human Resources*, 21:484–502, 11 2019.

[22] Ying-Huey Fua, M.O. Ward, and E.A. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. In *Proceedings Visualization '99 (Cat. No.99CB37067)*, pages 43–508, 1999.

[23] Mengpin Ge, Johannes Friedrich, and Leandro Vigna. 4 charts explain greenhouse gas emissions by countries and sectors, 2020.

[24] Samuel Gratzl, Alexander Lex, Nils Gehlenborg, Hanspeter Pfister, and Marc Streit. Lineup: Visual analysis of multi-attribute rankings. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '13)*, 19(12):2277–2286, 2013.

[25] Jiaming Han, Yuqiang Ren, Jian Ding, Xingjia Pan, Ke Yan, and Gui-Song Xia. Expanding low-density latent regions for open-set object detection. 2022.

[26] Siho Han and Simon S. Woo. Learning sparse latent graph representations for anomaly detection in multivariate time series. page 2977–2986, 2022.

[27] T Handhika, Murni, D P Lestari, and I Sari. Multivariate time series classification analysis: State-of-the-art and future challenges. *IOP Conference Series: Materials Science and Engineering*, 536(1):012003, jun 2019.

[28] M. Hao, M. Marwah, H. Janetzko, R. Sharma, D. A. Keim, U. Dayal, D. Patnaik, and N. Ramakrishnan. Visualizing frequent patterns in large multivariate time series. In Pak Chung Wong, Jinah Park, Ming C. Hao, Chaomei Chen, Katy Börner, David L. Kao, and Jonathan C. Roberts, editors, *Visualization and Data Analysis 2011*, volume 7868, page 78680J. International Society for Optics and Photonics, SPIE, 2011.

[29] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[30] Wei-Ning Hsu, Yu Zhang, and James Glass. Learning latent representations for speech generation and transformation. 2017.

[31] Dino Ienco and Roberto Interdonato. Deep multivariate time series embedding clustering via attentive-gated autoencoder. In Hady W. Lauw, Raymond Chi-Wing Wong, Alexandros Ntoulas, Ee-Peng Lim, See-Kiong Ng, and Sinno Jialin Pan, editors, *Advances in Knowledge Discovery and Data Mining*, pages 318–329, Cham, 2020. Springer International Publishing.

[32] Streamlit Inc. A faster way to build and share data apps, 2018.

[33] Jason Brownlee. Multivariate time series forecasting with lstms in keras. `https://machinelearningmastery.com/multivariate-time-series-forecasting-lstms-keras`, 2017. Accessed: 2023-06-29.

[34] Zhuochen Jin, Shunan Guo, Nan Chen, Daniel Weiskopf, David Gotz, and Nan Cao. Visual causality analysis of event sequence data. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1343–1352, 2021.

[35] Ian Jolliffe and Jorge Cadima. Principal component analysis: A review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374:20150202, 04 2016.

[36] Sasan Karamizadeh, Shahidan Abdullah, Azizah Manaf, Mazdak Zamani, and Alireza Hooman. An overview of principal component analysis. *Journal of Signal and Information Processing*, 08 2013.

[37] Fritz Lekschas, Brant Peterson, Daniel Haehn, Eric Ma, Nils Gehlenborg, and Hanspeter Pfister. Peax: Interactive visual pattern search in sequential data using unsupervised deep representation learning. *Computer Graphics Forum*, 39(3):167–179, 2020.

[38] Hui Li, Yunpeng Cui, Shuo Wang, Juan Liu, Jinyuan Qin, and Yilin Yang. Multivariate financial time-series prediction with certified robustness. *IEEE Access*, 8:109133–109143, 2020.

[39] Zhihan Li, Youjian Zhao, Jiaqi Han, Ya Su, Rui Jiao, Xidao Wen, and Dan Pei. Multivariate time series anomaly detection and interpretation using hierarchical inter-metric and temporal embedding. KDD '21, page 3220–3230, New York, NY, USA, 2021. Association for Computing Machinery.

[40] Troy Luhman and Eric Luhman. High fidelity image synthesis with deep vaes in latent space. 2023.

[41] Ivana Marin, Sven Gotovac, Mladen Russo, and Dunja Božić-Štulić. The effect of latent space dimension on the quality of synthesized human face images. *Journal of Communications Software and Systems*, 17(2):124–133, 5 2021.

[42] Dragan S Markovic, Irina Branovic, and Ranko Popovic. Smart grid and nanotechnologies: a solution for clean and sustainable energy. *Energy and emission control technologies*, 3:1–13, 2015.

[43] Hamid Moeeni, Hossein Bonakdari, and Seyed Ehsan Fatemi. Stochastic model stationarization by eliminating the periodic term and its effect on time series prediction. *Journal of Hydrology*, 547:348–364, 2017.

[44] Mohsen Mohammadzadeh. A spatio-temporal dynamic regression model for extreme wind speeds. *Extremes*, 17:221–245, 01 2014.

[45] Mohammad Amin Morid, Olivia R. Liu Sheng, Kensaku Kawamoto, and Samir Abdelrahman. Learning hidden patterns from patient multivariate time series data using convolutional neural networks: A case study of healthcare cost prediction. *Journal of Biomedical Informatics*, 111:103565, 2020.

[46] Nam Nguyen and Brian Quanz. Temporal latent auto-encoder: A method for probabilistic multivariate time series forecasting. 2021.

[47] Maya Papineau and Nicholas Rivers. Experimental evidence on heat loss visualization and personalized information to motivate energy savings. *Journal of Environmental Economics and Management*, 111:102558, 2022.

[48] Chi-Hieu Pham, Saïd Ladjal, and Alasdair Newson. Pcaae: Principal component analysis autoencoder for organising the latent space of generative networks. 2020.

[49] Tuan-Anh Pham, Jong-Hoon Lee, and Choong-Shik Park. Mst-vae: Multi-scale temporal variational autoencoder for anomaly detection in multivariate time series. *Applied Sciences*, 12(19), 2022.

[50] Zulfiqar Qutrio Baloch, Syed Raza, Rahul Pathak, Luke Marone, and Abbas Ali. Machine learning confirms nonlinear relationship between severity of peripheral arterial disease, functional limitation and symptom severity. *Diagnostics*, 10:515, 07 2020.

[51] Oxana Rodionova, Sergey Kucheryavskiy, and Alexey Pomerantsev. Efficient tools for principal component analysis of complex data— a tutorial. *Chemometrics and Intelligent Laboratory Systems*, 213:104304, 2021.

[52] Hayk Shoukourian, Torsten Wilde, Detlef Labrenz, and Arndt Bode. Using machine learning for data center cooling infrastructure efficiency prediction. In *2017 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pages 954–963, 2017.

[53] Knut J. Strømmen, Jim Tørresen, and Ulysse Côté-Allard. Latent space unsupervised semantic segmentation. 2022.

[54] Peiwang Tang and Xianchao Zhang. Mtsmae: Masked autoencoders for multivariate time-series forecasting. 10 2022.

[55] Xiaoji Wan, Hailin Li, Liping Zhang, and Yenchun Jim Wu. Dimensionality reduction for multivariate time-series data mining. *J. Supercomput.*, 78(7):9862–9878, may 2022.

[56] M. Wattenberg and J. Kriss. Designing for social data analysis. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):549–557, 2006.

[57] Ke Xu, Jun Yuan, Yifang Wang, Claudio Silva, and Enrico Bertini. Mtseer: Interactive visual exploration of models on multivariate time-series forecast. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA, 2021. Association for Computing Machinery.

[58] Ji Soo Yi, Niklas Elmqvist, and Seungyoon Lee. Timematrix: Analyzing temporal social networks using interactive matrix-based visualizations. *Int. J. Hum. Comput. Interaction*, 26:1031–1051, 11 2010.

[59] Jiaming Yin, Weixiong Rao, Mingxuan Yuan, Jia Zeng, Kai Zhao, Chenxi Zhang, Jiangfeng Li, and Qinpei Zhao. Experimental study of multivariate time series forecasting models. CIKM '19, page 2833–2839, New York, NY, USA, 2019. Association for Computing Machinery.

[60] Yuncong Yu, Dylan Kruyff, Jiao Jiao, Tim Becker, and Michael Behrisch. Pseudo: Interactive pattern search in multivariate time series with locality-sensitive hashing and relevance feedback. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):33–42, 2023.

A Qualtrics Survey

Response Summary:

Section 1. Research projects involving human participants

P1. Does your project involve human participants? This includes for example use of observation, (online) surveys, interviews, tests, focus groups, and workshops where human participants provide information or data to inform the research. If you are only using existing data sets or publicly available data (e.g. from Twitter, Reddit) without directly recruiting participants, please answer no.

- No

Section 2. Data protection, handling, and storage

The General Data Protection Regulation imposes several obligations for the use of **personal data** (defined as any information relating to an identified or identifiable living person) or including the use of personal data in research.

D1. Are you gathering or using personal data (defined as any information relating to an identified or identifiable living person)?

- No

Section 3. Research that may cause harm

Research may cause harm to participants, researchers, the university, or society. This includes when technology has dual-use, and you investigate an innocent use, but your results could be used by others in a harmful way. If you are unsure regarding possible harm to the university or society, please discuss your concerns with the Research Support Office.

H1. Does your project give rise to a realistic risk to the national security of any country?

- No

H2. Does your project give rise to a realistic risk of aiding human rights abuses in any country?

- No

H3. Does your project (and its data) give rise to a realistic risk of damaging the University's reputation? (E.g., bad press coverage, public protest.)

- No

H4. Does your project (and in particular its data) give rise to an increased risk of attack (cyber- or otherwise) against the University? (E.g., from pressure groups.)

- No

H5. Is the data likely to contain material that is indecent, offensive, defamatory, threatening, discriminatory, or extremist?

- No

H6. Does your project give rise to a realistic risk of harm to the researchers?

- No

H7. Is there a realistic risk of any participant experiencing physical or psychological harm or discomfort?

- No

H8. Is there a realistic risk of any participant experiencing a detriment to their interests as a result of participation?

- No